Evaluation and Continual Improvement for an Enterprise AI Assistant

Akash V. Maharaj, Kun Qian, Uttaran Bhattacharya, Sally Fang, Horia Galatanu,
Manas Garg, Rachel Hanessian, Nishant Kapoor, Ken Russell,
Shivakumar Vaithyanathan, and Yunyao Li
Adobe Inc.
{maharaj, kunq, ubhattac, xinf, horiag, mangarg, hanessia}@adobe.com
{niskapoor, kenrusse, vaithyan, yunyaol}@adobe.com

Abstract

The development of conversational AI assistants is an iterative process with multiple components. As such, the evaluation and continual improvement of these assistants is a complex and multifaceted problem. This paper introduces the challenges in evaluating and improving a generative AI assistant for enterprises, which is under active development, and how we address these challenges. We also share preliminary results and discuss lessons learned.

1 Introduction

Generative AI assistants for enterprises hold the great promise of significantly improved productivity, lowered barrier-to-entry, drastically increased product adoption, transformative amplification of creativity, and delivery of better customer and employee experiences (Kumar et al., 2023). Developing such an AI assistant is typically an iterative process, with its evaluation and continual improvement at the center.

Fig. 1 depicts the high-level architecture of Adobe Experience Platform AI Assistant[1] (Bhambhri, 2024), a generative AI assistant built for an enterprise data platform. As can be seen, it is a complex pipeline with multiple underlying components consisting of one or more machine learning models based on large language models (LLMs) or small language models (SLMs). Users interact with the system via a conversational interface to obtain answers based on heterogeneous data sources. The evaluation and continual improvement of such a system is a complex and multifaceted problem with the following key challenges.

Metrics. The success of Assistant is ultimately measured by metrics such as user engagement, user satisfaction, and user retention. However, such metrics are lag measures obtainable only after building and deploying Assistant in production. To guide continual improvement of Assistant, we also need to define metrics that are *lead* measures for various aspects of Assistant that are likely to impact the lag measures.

Data. To produce reliable evaluation metrics for Assistant, we need data that are both representative and high-quality. We need a systematic approach to obtain such high-quality data at scale.

Dynamics. As shown in Fig. 1, a real-world AI assistant usually consists of a complex pipeline of components. Each component evolves over time as both the underlying models and the assistant's functionalities change. Further, in enterprise settings, the distribution of questions asked is ever-changing as the customer base shifts and grows and existing customers mature in their adoption of the assistant. We need to consider such customer dynamics.

Human-Centered Design. The success of Assistant depends on both the capabilities of its underlying components and the user interface (UI) that surfaces those capabilities to support the overall user experience. As such, the evaluation and continual improvement for Assistant need to take all underlying components as well as UI into consideration for such a human-centered system (Liao and Vaughan, 2023).

Privacy and Security. Enterprise AI assistants like Assistant often deal with sensitive user data. We need to evaluate its performance while securely handling customer data and prevent unauthorized access or misuse (Wu et al., 2023; Yao et al., 2024).

The rest of this paper presents our proposed solution for addressing these changes. We also share our preliminary results and discuss lessons learned so far. Our main contributions include:

- A comprehensive continual improvement framework to support the evaluation and continual improvement for Assistant.
- A taxonomy of error types for error analysis and continual improvement.

[1] Hereafter referred to as Assistant

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 17–24
June 20, 2024 ©2024 Association for Computational Linguistics

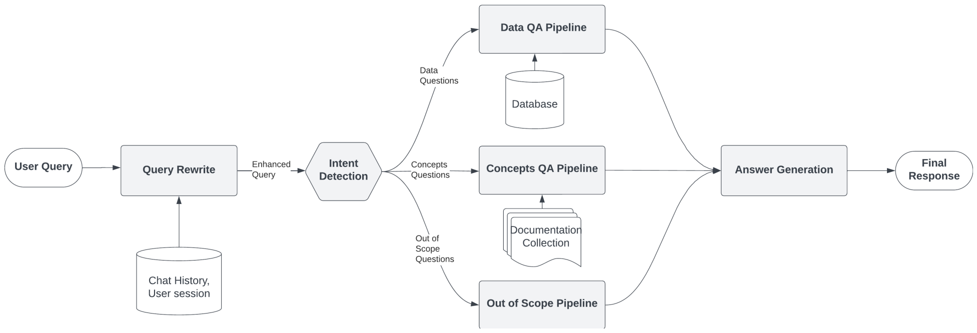


Figure 1: `Assistant` Overall Architecture

- Identifying the limitations of existing approaches on the evaluation of AI assistants.
- Highlighting the influential role of human-centered UI design in the evaluation and continual improvement of `Assistant`.
- Productionizing such a framework, sharing initial results and lessons learned.

2 Limitations of Existing Approaches

Common approaches for evaluating AI assistants include evaluation using explicit feedback, evaluation using implicit feedback, benchmarking (Liang et al., 2022), and human evaluation (Fernandes et al., 2023). Explicit feedback is collected from users through feedback buttons, direct prompts, or questions on their preferences. In contrast, implicit feedback is gathered from user actions within a system, such as clicks, views, or navigation patterns, providing insights into user behavior and preferences without requiring direct input. Evaluating with benchmark datasets is also a common way to evaluate AI assistants. These approaches, while important and effective to a certain degree, suffer from various limitations when it comes to evaluating an enterprise AI assistant such as `Assistant`, which is under active development and improvement.

2.1 Limitations of Explicit Feedback

Collecting explicit feedback from the users seems to be the most straightforward way to gauge user satisfaction and gather input to measure and improve the performance of an AI assistant. Table 1 illustrates the initial set of explicit feedback for `Assistant` from our early customers. We can observe several limitations of this approach.

Sparsity. Explicit user feedback is sparse. From Table 1, we can see that 76% of all customer interactions receive no explicit feedback at all. This sparsity issue makes it challenging to understand user experience and satisfaction comprehensively and hampers efforts to improve `Assistant`.

Representativeness. Since sharing explicit feedback is not mandatory, not every user does so. As shown in Table 1, users from two organizations shared no feedback at all. Further examination showed that most feedback came from a small number of users. In fact, about 30% of all the feedback originated from one user. Such a highly skewed feedback distribution may misrepresent the overall sentiment towards `Assistant`, and fail to reflect the diversity of users' experiences and opinions.

Lack of detailed feedback. partly due to minimizing user effort and partly because users only see the final response, explicit feedback is usually gathered via a simple UI form (*e.g.*, 👍, 👎 buttons). Unfortunately, feedback gathered this way often fails to capture the nuances of user experiences and preferences. For instance, a negative feedback indicating an incorrect final response is insufficient to pinpoint specific components for improvement. New approaches like showing step-by-step explanations and getting user feedback for the explanation are alternative ways to get detailed feedback from users and map them to specific components.

2.2 Limitations of Implicit Feedback

Implicit feedback has been extensively used in evaluating and improving intelligent systems (*e.g.*, Jawaheer et al. (2014); Koren et al. (2021)), and performance measurements of concrete tasks have been recommended as the best metric for evaluating natural language generation systems (Saphra et al., 2023). This approach has several limitations when evaluating AI assistants. First, since implicit feedback is obtained indirectly and passively from user actions, it may not always reflect users' true preferences. Prior work uses denoising techniques to prune the noisy interactions to avoid serious negative impact (Wang et al., 2021). In addition,

Table 1: Feedback type distribution and engagement ratio from different customers

Customer	Positive feedback	Negative feedback	No feedback
Org1	22.8%	16.2%	61%
Org2	12.6%	11.2%	76.2%
Org3	3.2%	24.9%	71.9%
Org4	2.7%	5.0%	92.3%
Org5	11.3%	5.2%	83.5%
Org6	5.6%	9.7%	84.7%
Org7	8.6%	21.4%	70%
Org8	15.4%	7.7%	76.9%
Org9	0%	0%	100%
Org10	0%	0%	100%
Total	**10.72%**	**13.12%**	**76.16%**

deriving implicit feedback from user interactions could be a challenge on its own. For instance, while meaningful implicit feedback is readily available for recommender systems in contexts such as online shopping (clicks, page views, add-to-cart, etc.), implicit signals available in AI assistants are less clearly related to concrete user goals. Specifically, users have a wide variety of goals, and the concrete tasks to achieve those goals are often very delayed.

2.3 Limitations of Off-the-Shelf Benchmarks

Although public benchmark datasets for general tasks are abundant (*e.g.*, Chang et al. (2023) lists 46 public benchmark datasets), they are often not applicable for domain-specific AI assistants. Creating domain-specific benchmark datasets is labor-intensive, time-consuming, and requires domain expertise. Moreover, assistants' workload and tasks may also evolve. Thus, there is no one static benchmark data that suits all (Mizrahi et al., 2024). Therefore, benchmark data creation itself is a continual process.

3 Our Approach

In this section, we introduce our framework to overcome the aforementioned challenges (Section 1) and limitations of existing approaches (Section 2) for evaluating an enterprise-grade AI assistant under active development.

3.1 Design Decisions

We first present a few key design decisions to balance the trade-offs to be made, both in terms of breadth and depth of any given type of evaluation.

Prioritize metrics directly impacted by production changes. The ultimate goal of `Assistant` is to improve the productivity and creativity of our users and lower barriers to entry. Since it takes time to materialize such lag measures, we focus on directly responsive “correctness” metrics, assuming a more correct `Assistant` will ultimately lead to positive downstream outcomes.

Align metrics with user experience. Not all errors are equal. The impact on the user experience of one incorrect citation in an otherwise correct answer is very different from that of a completely hallucinated answer. We aim to capture this nuance in the design of our error metrics.

Human Evaluation over automated evaluation. We believe that, despite challenges (Clark et al., 2021), human judgments are still best aligned with eventual user outcomes. As such, we prioritize human evaluation over automated evaluation. Once high-quality human judgments are collected, they can be used to validate which automatic evaluations are meaningful for specific tasks and components.

Efficient allocation of human evaluators. To conduct human evaluation at scale, we focus on the efficient allocation of human annotators. Specifically, simple annotation tasks are done by non-experts, while complex error analysis and the determination of how to make improvements are left up to engineers with domain expertise.

Collect both end-to-end metrics and component-wise metrics. We collect both individual and collective metrics to understand the overall quality of the system as well as which parts need to improve.

System-wide improvements. All components in `Assistant`, from ML/rule-based models, UI/UX components, to underlying data, may impact system performance. Therefore, instead of focusing solely on ML model improvements, we consider the entire “vertical" system holistically and leave no improvement off the table.

Prioritize human evaluation. Automated evaluation, which utilizes standard metrics and evaluation tools, is popular for its efficiency and objectivity (Chang et al., 2023). However, although more labor-intensive and time-consuming, manual evaluation by domain experts is more reliable in reflecting the final user impact. As such, we prioritize human evaluation over automation.

3.2 Severity-based Error Taxonomy

Designing metrics that align with our end users' judgments of the correctness and usefulness of `Assistant` is a complex task. We observed relatively high error rates from an early version of `Assistant` (over 50%), yet our users did not seem

Table 2: Error Severity Framework in `Assistant`

Category	Definition	Consequence	Examples
Severity 0	**Answer looks right, but is wrong**	***Erodes trust with the users***	- Convincing Concepts QA answers that are pure hallucinations - Incorrect Data QA answers that cannot easily be verified independently
Severity 1	**Answer looks wrong, user can't recover**	***Frustrates users***	- Failure to answer with generic error message - Answers with obvious logical inconsistencies, *e.g.*, mixing UI docs and API docs
Severity 2	**Answer looks wrong, user can recover**	***Annoys users***	- Misunderstood questions that user is able to rephrase and get correct answer - Incorrect out-of-scope question rejection that user is able to override

to perceive error rates to be this high in their self-reported surveys and regular feedback sessions. This discrepancy, consistent with the earlier observation that not all errors are the same (Freitag et al., 2021), led us to develop a *taxonomy* of errors.

To illustrate this point, consider the past two decades, where internet search has become a dominant (semi) natural language interface. In this domain, humans have become accustomed to certain classes of errors. When we do not get the desired results from a search engine, we rephrase and iterate till we find the answer. The initial failure of the search engine is annoying but generally tolerable unless we cannot find our answer even after many re-phrasings. At this point, we are left frustrated. Inspired by DevOps terminology (Kim et al., 2021), we can define two separate classes of errors: **Severity-2** ("Sev-2" for short) errors are annoying but repairable via rephrasing, while **Severity-1** ("Sev-1" for short) errors are not repairable.

Meanwhile, the rise of generative AI has introduced an entirely new class of error: answers that are convincing and look correct but are, in fact, wrong. Depending on the use case, these may be tolerable (or even desirable), but in the realm of enterprise assistants, these errors are troubling. They erode user trust and may lead to complete abandonment of the assistant. We term these **Severity-0** errors, "Sev-0" for short. Table 2 summarizes this severity-based error taxonomy, which has become an organizing principle for the evaluation and improvement of `Assistant`, as we discuss next.

3.3 Framework for Evaluation and Continual Improvement

Fig. 2 depicts our proposed evaluation and improvement framework. It includes three main components: `Assistant`, itself, a dedicated Annotation Tool, and a separate environment for Error Analysis. Human evaluation drives the evaluation and improvement of `Assistant`.

To ensure the efficient allocation of human resources, non-experts provide large-scale annotation of masked production data, while domain experts provide detailed error analysis on a sample of production data. For each annotation task, to ensure annotation quality, we design the UI and annotation guidelines iteratively with pilot study and improvements. We include training modules and exercises to ensure annotators meet a minimum bar of sufficient domain understanding. We assign multiple annotators for each annotation task to further ensure the annotation quality and conform to best practices (van der Lee et al., 2021).

We design different annotation tasks to assess the quality of different `Assistant` components and improvements needed. By collecting annotations based on prior interactions in the production system, we can generate both error metrics by severity (by comparing human labels to the choices the system made in production), *and* new golden-labeled data for model improvements.

Error analysis is a crucial step in gating improvement. At this step, domain experts — those with deep knowledge of how `Assistant` is designed — review samples of errors, identify error patterns, and determine specific improvements. These improvements take many potential forms, from prompt engineering to training and improving in-house models, to creating new templates and patterns for synthetic data, to more holistic changes such as improving the user experience or optimizing the specialized data indexes that are queried by `Assistant`, (for example, fine-tuning embeddings,

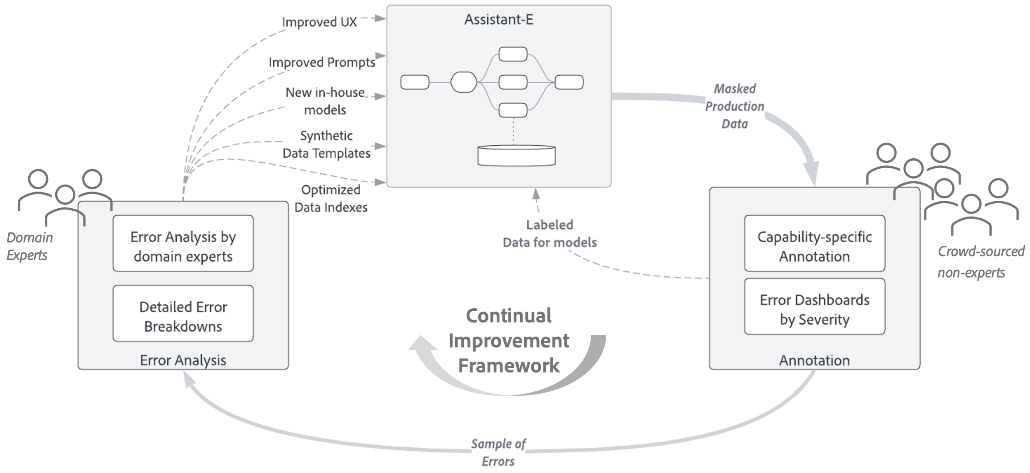
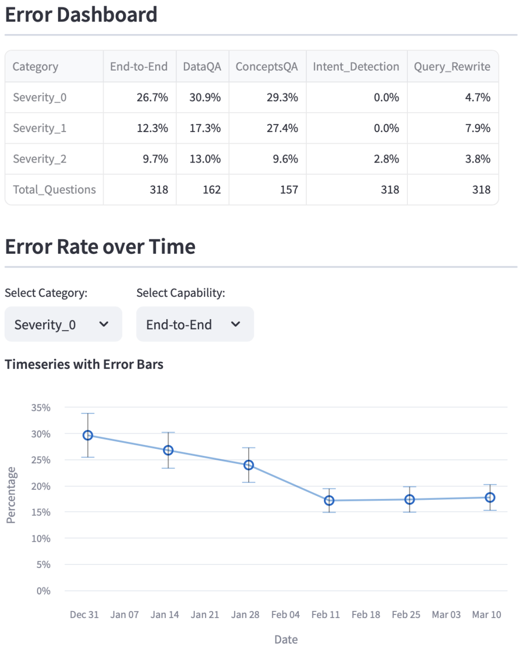


Figure 2: Evaluation and continual improvement framework of `Assistant`

or updating database schema). This last category of improvements is only possible when the application is viewed holistically, and all stakeholders are involved in error analysis.

4 Preliminary Results: Examples

While `Assistant` remains in active development, our evaluation and continual improvement framework already show promising impacts on both the prioritization and the design of improvements. In this section, we share the preliminary results obtained so far by examples.



Error Dashboard

Category	End-to-End	DataQA	ConceptsQA	Intent_Detection	Query_Rewrite
Severity_0	26.7%	30.9%	29.3%	0.0%	4.7%
Severity_1	12.3%	17.3%	27.4%	0.0%	7.9%
Severity_2	9.7%	13.0%	9.6%	2.8%	3.8%
Total_Questions	318	162	157	318	318

Error Rate over Time

Select Category: Severity_0

Select Capability: End-to-End

Figure 3: Dashboard showing snapshot of Error Severities and time-evolution for a single component. Illustrative data of similar magnitude to production numbers.

Fig. 3 illustrates an example error dashboard produced by the annotation tool, showing component-wise and end-to-end errors with further breakdown by severity levels as well as how they change over time. This dashboard is monitored by all stakeholders and is used to track the impact of feature releases and improvements. While ideas for improving `Assistant` may be endless, detailed error analysis allows the team to follow a powerful organizing principle: focusing on reducing error rates based on their actual impact on the users.

For instance, the example report in Table 3 shows that Out-of-Scope errors were our largest contributor to Sev-0 errors in Sprint 1. To address this, we introduced an Out-of-Scope text classifier using an in-house model, which achieved 90% precision and successfully reduced most such errors.

However, the new classifier also led to a new, particularly frustrating source of errors: in-scope questions misclassified as out-of-scope would no longer be answered. Without being able to quickly improve the classifier's precision, we used our other available lever of improvement and designed an override mechanism in the UI to allow users to receive an answer. As the Sprint 2 report shows, this UI change converted a potential Sev-1 error (refusal to answer) into a Sev-2 error (the user could now recover), showcasing how human-centered UI design allows holistic improvement of `Assistant`.

Explainability is important for improving user trust and comprehension. By helping users discover wrong answers with better explainability, we can reduce Sev-0 errors and move them to Sev-1/Sev-2 error buckets. We took a data-driven approach to choose from many applicable explainability tech-

Table 3: An example output of error analysis for Concept QA (illustrative data, real labels) from one sprint to the next after Out-Of-Scope detection was deployed.

Error Severity	% Sprint1	% Sprint2
Sev-0	53.4%	36.6%
OutOfScope	**21.6%**	**6.2%**
Hallucination	17.0%	16.4%
Doc-Retrieval	13.6%	14.0%
LLM-Error	1.1%	0.0%
Sev-1	46.6%	44.4%
Hallucination	36.4%	33.0%
Citation	5.7%	5.1%
LLM-Error	4.5%	6.3%
Sev-2	-	6.9%
OutOfScope *(incorrect rejection)*	-	6.9%

Table 4: Decision matrix for explainability techniques

Explainability techniques	Potential impact	Engineering difficulty	Congitive load
technique1	0.0%	high	low
technique2	8.6%	high	high
technique3	48.6%	low	low
technique4	88.6%	medium	medium
technique5	20.0%	high	low
technique6	100%	medium	low
technique7	74.3%	high	low

niques (Danilevsky et al., 2020). We first went through the Sev-0 queries obtained during a certain window and examined which technique(s) can be used to alleviate the severity of each error based on the potential overall impact of each explainability technique, its implementation difficulty, and human cognitive load. We created a decision matrix (Table 4) based on the analysis, and we focused on only 2 of the 7 options from (Li et al., 2024). As we move forward, we expect many more such informed improvements based on our framework.

5 Discussion

This framework has organically evolved during the development of `Assistant`. While many of the design choices laid out may seem obvious in hindsight, they were not as clear at the beginning of this project, and so it is worth discussing the lessons we have learned along the way.

First, we have found that metric design is of paramount importance. The severity framework came after many iterations in trying to connect enthusiastic early customer feedback with a seemingly large overall error rate. The insight that customers have varying tolerance depending on the class of errors has become a powerful organizing principle for our prioritization and resource allocation to improve `Assistant`.

Next, we have seen firsthand the benefits of building a decomposed system as opposed to depending on a single, monolithic model. The choice to decompose into multiple, orchestrating models was led by constraints such as task specialization and the need to query real-time data. We have also reaped the secondary benefit of having many available "levers of improvement" (prompts, in-house models, specialized indexes, UX improvements, etc.), many more than what is possible in a single language model paradigm.

Finally, iterative and agile development are more important than designing everything upfront and building specialized tools. For instance, while it is tempting to build in-house tools, using spreadsheets as a simple alternative initially allows us to learn important lessons on designing the annotation tasks, from annotation guidelines to the actual UI.

6 Future Work

As we continue to develop `Assistant` and onboard more customers, we plan to extend our evaluation and continual improvement framework with more human-in-the-loop/LLM-in-the-loop automation to scale our evaluation and error analysis processes (Zheng et al., 2023). In addition, the current framework heavily focuses on retrospective analysis based on *past* customer interactions. We plan to extend it with more proactive user studies and evaluation of in-development functionalities. Moreover, personalization is also important for enterprise AI assistants since we have customers with different technical levels. To provide the best experience to various personas in potentially different languages, additional evaluation metrics and datasets proposed in (Jadeja and Varia, 2017; Ahuja et al., 2023) may also be considered. As we have emphasized, human-centered design is essential for the success of `Assistant`. We plan to further explore how the deeper interplay between ML and UX components in this new paradigm of HCI can lead to more explainable and accurate assistants. Finally, the impact of generative AI applications in the workplace is an important new area of study (Brynjolfsson et al., 2023). As we enroll new customers, we intend to run A/B tests (Hussey and Hughes, 2007) that assess the causal impact of `Assistant` on the engagement and productivity of customers.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, et al. 2023. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Anjul Bhambhri. 2024. New ai integrations in adobe experience platform. `https://business.adobe.com/blog/the-latest/new-ai-assistant-in-adobe-experience-platform`. Accessed: 2024-04-23.

Erik Brynjolfsson, Danielle Li, and Lindsey R Raymond. 2023. Generative ai at work. Technical report, National Bureau of Economic Research.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. *CoRR*, abs/2010.00711.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, and André F. T. Martins. 2023. Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation. *Transactions of the Association for Computational Linguistics*, 11:1643–1668.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Michael A Hussey and James P Hughes. 2007. Design and analysis of stepped wedge cluster randomized trials. *Contemporary clinical trials*, 28(2):182–191.

Mahipal Jadeja and Neelanshi Varia. 2017. Perspectives for evaluating conversational AI. *CoRR*, abs/1709.04734.

Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, 4(2).

Gene Kim, Jez Humble, Patrick Debois, John Willis, and Nicole Forsgren. 2021. *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. IT Revolution.

Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142.

Akit Kumar, M.S. Lakshmi Devi, and Jeffrey S. Saltz. 2023. Bridging the gap in ai-driven workflows: The case for domain-specific generative bots. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2421–2430.

Yunyao Li, Dragomir R. Radev, and Davood Rafiei. 2024. *Natural Language Interfaces to Databases*. Springer Nature.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Q Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation.

Naomi Saphra, Eve Fleisig, Kyunghyun Cho, and Adam Lopez. 2023. First tragedy, then parse: History repeats itself in the new era of large language models. *arXiv preprint arXiv:2311.05020*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 373–381.

Xiaodong Wu, Ran Duan, and Jianbing Ni. 2023. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.