Matthew K. Hong* mkhong87@uw.edu University of Washington Seattle, Washington

Derek DeBellis dedebell@microsoft.com Microsoft Research Redmond, Washington

ABSTRACT

Prototyping AI user experiences is challenging due in part to probabilistic AI models making it difficult to anticipate, test, and mitigate AI failures before deployment. In this work, we set out to support practitioners with early AI prototyping, with a focus on natural language (NL)-based technologies. Our interviews with 12 NL practitioners from a large technology company revealed that, in addition to challenges prototyping AI, prototyping was often not happening at all or focused only on idealized scenarios due to a lack of tools and tight timelines. These findings informed our design of the AI Playbook, an interactive and low-cost tool we developed to encourage proactive and systematic consideration of AI errors before deployment. Our evaluation of the AI Playbook demonstrates its potential to 1) encourage product teams to prioritize both ideal and failure scenarios, 2) standardize the articulation of AI failures from a user experience perspective, and 3) act as a boundary object between user experience designers, data scientists, and engineers.

CCS CONCEPTS

• Human-centered computing \rightarrow Interface design prototyping.

KEYWORDS

Human-AI interaction, prototyping, AI failures, natural language technologies

ACM Reference Format:

Matthew K. Hong, Adam Fourney, Derek DeBellis, and Saleema Amershi. 2021. Planning for Natural Language Failures with the AI Playbook. In CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 11 pages. https: //doi.org/10.1145/3411764.3445735

CHI '21, May 8–13, 2021, Yokohama, Japan

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-8096-6/21/05...\$15.00 https://doi.org/10.1145/3411764.3445735 Adam Fourney adamfo@microsoft.com Microsoft Research Redmond, Washington

Saleema Amershi samershi@microsoft.com Microsoft Research Redmond, Washington

1 INTRODUCTION

Early prototyping is a common design practice for exploring alternative user experiences before making significant investments in building out fully functional systems. Prototyping AI user experiences, however, is challenging due in part to the probabilistic nature of AI models making it difficult to anticipate the success and failure behaviors of AI-driven systems. For example, testing interaction design ideas with established low- or medium-fidelity prototyping techniques, like paper prototyping or Wizard of Oz experiments, may appear intractable when AI models can generalize in unpredictable ways and evolve over time [30]. Failure to consider, test, and mitigate potential AI failures prior to deployment can, however, be detrimental to users and costly to fix [20].

In this work, we set out to support practitioners with early prototyping of AI-driven systems. Our formative interviews with practitioners from a large technology company revealed that, in addition to the challenges of anticipating AI errors [7, 30], prototyping was often overlooked altogether or focused only on idealized scenarios for reasons including a lack of tools and limited time allocated to this stage of development.

To encourage proactive consideration of AI errors while balancing the need to keep pace with rapid deployment cycles, we created the AI Playbook, a low-cost tool for systematically exploring common error scenarios of envisioned AI products and providing contextually relevant and actionable guidance for simulating and testing those scenarios prior to deployment. We scoped our explorations with the AI Playbook in this work within the context of human interaction with natural language (NL) based systems (e.g., conversational agents, writing assistance). Our qualitative evaluation of the Playbook with practitioners working on NL products demonstrates its potential to 1) encourage teams to prioritize both ideal and failure scenarios, 2) standardize the articulation of AI user experience failures, and 3) act as a boundary object between interdisciplinary teams.

In summary, this paper contributes the following:

- Semi-structured interviews with 12 AI practitioners highlighting factors preventing or impeding early prototyping as well as the costs of discovering errors post-deployment;
- A taxonomy of natural language failures based on a systematic characterization of common user experience errors with natural language based products;
- The AI Playbook, a tool that operationalizes our taxonomy by asking practitioners questions about their envisioned AI

^{*}Work done as an intern researcher at Microsoft Research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Hong and Adam et al.

products and recommending a set of failure scenarios to consider and test at design time;

 An evaluation of the AI Playbook with 9 AI practitioners within the context of natural language-based technologies demonstrating its potential to standardize early stage prototyping of AI user experiences.

2 RELATED WORK

We situate our research within the context of work supporting AI user experience design and, in particular, helping practitioners anticipate and plan for AI an NL failures.

2.1 Challenges Designing AI User Experiences

Numerous studies have highlighted challenges of designing AI user experiences [7, 29, 30]. From a synthesis and reflection on work involving designing with AI, Yang et al. [30] catalogued challenges spanning the classic end-to-end user centered design process including difficulties envisioning new AI products and features due to limited understanding of AI capabilities, difficulties with rapid prototyping and testing given unpredictable AI behaviors, and difficulties communicating and collaborating across design and engineering. Dove et al. [7] highlighted similar challenges based on a survey of 51 UX practitioners working regularly with AI, citing a lack of adequate education around AI and machine learning as design materials. Yang et al's interviews with 13 design practitioners [29] raised similar issues and recommended creating AI user experience abstractions and examples to support designers in practice. Our interviews with AI practitioners add to these findings that the rapid pace of industry development along with agile philosophies of shipping early and failing fast further exacerbate AI design challenges, resulting in ad hoc consideration of potential AI errors or in some cases none at all.

These previous studies of AI design challenges have all called for new tools to help design practitioners anticipate, test, and mitigate potential AI errors before deployment [7, 29, 30]. This along with our findings around concerns about potentially slowing down development to conduct comprehensive design explorations suggests the need for low-cost tools for proactive consideration of AI errors. Our AI Playbook aims to fill this gap and standardize how errors are approached during early design with AI.

2.2 Design Guidance for AI User Experiences

The human-computer interaction and design communities have proposed numerous guidelines and recommendations for designing AI user experiences, including best practices for mitigating AI errors [1, 11, 15, 16]. Horvitz's principles of mixed initiative systems [15], for instance, advocates for inferring ideal actions in the face of AI uncertainty and costs of errors and recommends techniques for helping users recover from or refine erroneous behaviors such as employing dialog or allowing users to intervene via direct manipulation. In another example, the Guidelines for Human-AI Interaction [1] synthesizes and validates AI user experience guidance to prescribe how general AI systems should behave during different phases of user interaction including when the AI might be wrong (e.g., supporting efficient dismissal and correction, and scoping services when uncertain). In the domain of natural language interaction, IBM's guidelines on conversation planning [16] recommends designing for ideal paths as well as potential failures, ensuring AI-driven bots can 'elegantly fail' and repair the conversation.

While these guidelines and recommendations describe best practices, they are not intended as requirements for all AI user experiences. This leaves designers with the tasks of identifying which recommendations are applicable to an envisioned application and deciding how to implement the guidance, both of which can be challenging for designers new to or untrained in AI [7, 30]. In addition, these recommendations often provide guidance at a high-level and lack actionable specifics about how to realize or simulate likely interaction scenarios and failures. Our AI Playbook complements these efforts by enumerating error scenarios likely to be encountered in specific applications and offering contextually relevant guidance about simulating and testing those errors early in development.

2.3 Prototyping AI User Experiences

Classic low-fidelity prototyping techniques, such as sketching and paper prototyping, allow designers to rapidly explore multiple alternative design choices to *get the design right* prior to investing in implementing fully functional systems [4]. As previously described, however, prior work has highlighted numerous challenges in prototyping AI interactions due in part to difficulties knowing what behaviors and error conditions might manifest and should be sketched out [28]. Our AI Playbook addresses these challenges by recommending error scenarios to test given characteristics of an envisioned AI product.

Wizard of Oz (WoZ) techniques support early testing of user interaction with computing systems by employing human experimenters (the 'Wizards') to control system behaviors in reaction to user input [10, 17]. WoZ techniques were originally developed in the context of envisioning future intelligent language experiences [12] and have evolved over the course of three decades to aid wizards in tasks such as simulating realistic system responses and reducing response latency [25, 26]. For example, panels of wizards have been used to reduce response latency by distributing output generation across multiple people [18].

Still, convincingly simulating probabilistic AI behaviors and errors remains challenging [26, 30]. Recently, Yang et al. [28] proposed an approach to simulating natural language errors via a hybrid of WoZ and off-the-shelf ML toolkits, such as by passing text through multiple rounds of machine translation to introduce 'noise' in intent detection. Our AI Playbook leverages these and other advances when recommending techniques for simulating and testing specific AI errors.

Medium- and high-fidelity prototyping methods involve building out semi-functional systems to better assess interactivity. Within the context of AI, this may involve temporarily leveraging off-theshelf AI models or services or building out simple models with limited behaviors to approximate a finite range of functionalities before a more robust model is available. For example, systems such as Crayons [9] and the \$1 gesture recognizer [27] enable practitioners to quickly approximate an envisioned AI model's behavior with a few examples. Similarly, crowdsourcing platforms have been used in place of or in combination with AI models to support testing before deployment or to drive intelligent behaviors of shipped products before robust models are available (e.g., [6, 18]). Our investigations revealed that practitioners struggle to conduct any type of prototyping given tight shipping schedules. We therefore focus our current work on supporting rapid prototyping typically conducted in the earliest stages of design ideation.

2.4 Natural Language User Experience Errors

Errors are an unavoidable byproduct of AI systems that often rely on imperfect data and probabilistic models. Previous work provides a broad catalogue of such errors in the context of natural language understanding and processing [22, 23]. For example, Moldovan et al. analyzed modules of a linear NL model used in an open-domain question answering system to characterize information retrieval errors [22]. The authors identified 10 modules, or sources of errors, ranging from keyword preprocessing (e.g., spellcheck) to response formulation (e.g., answer string shifted slightly to left or right).

Others have mapped NL errors to theoretical frameworks of communication and interaction [2, 8, 13, 24]. For example, according to Herbert Clark's grounding model of human-human communication, participants in conversation coordinate on four different levels to achieve mutual understanding including channel, signal, intention, and conversation levels [5]. Paek and Horvitz applied the same coordination scheme to ground human-machine interactions and errors [24]. Here, errors are classified into four categories depending on the level in which they occurred: channel level errors occur when an AI fails to attend to a user's attempts to open a channel of communication (e.g., uttering a wake command); signal level errors occur when an AI fails to process a user's spoken utterance due to transcription issues; Intention level errors happen when an AI fails to comprehend the semantic content of the user's spoken utterance (e.g., calendar application fails to recognize 'erase event' instead of 'delete event'); and conversation level errors occur when a user formulates a query that falls outside of the AI's trained capabilities (e.g., asking a calendar application about the weather).

Our taxonomy of NL failures builds on Clark's grounding model to characterize failures that affect the end user experience of NL products. We further operationalize this taxonomy within the AI Playbook, an interactive tool to support practitioners with anticipating, prototyping, and testing common error scenarios early in product development.

3 EXISTING NL PROTOTYPING PRACTICES

We scoped our efforts to support AI prototyping to prototyping interaction with technologies driven by natural language understanding and processing. Our formative investigations therefore involved semi-structured interviews with NL practitioners to understand their existing practices and challenges.

3.1 Interview Study

3.1.1 Recruitment. We recruited 12 practitioners from a large technology company through snowball sampling. Our inclusion criteria included product teams working with NL technologies who were willing to participate in a follow up user study (described in Section 5 of this paper). Our participants had one to 15 years of practical

experience and included a mix of user researchers, designers, and program managers—representing NL product teams in the areas of search, writing assistance, conversational AI, and recommendation (See Table 1). IRB approval was obtained prior to the consent and data collection period of April 9 to June 2, 2020.

3.1.2 Procedure. We used a video conferencing tool to conduct our interviews. Interview topics included practical challenges during design and development and, in particular, in designing for NL errors. Questions included: What does your prototyping process look like? How do you test early designs? What happens when failures occur? Have you experienced any surprises post-deployment? If so, why and what was the impact or costs? All interviews lasted approximately 30 minutes, were audio recorded, and transcribed for analysis.

3.1.3 Analysis. We collected 352 minutes of audio data, and the matically analyzed the transcripts through multiple iterative phases. The first author coded the transcripts through an open coding process, identifying and noting text segments related to challenges in prototyping and testing AI experiences. In a second pass, the author organized the annotated quotes into multiple low-level subthemes. All the authors then reviewed the annotated quotes and assessed thematic relevance between subthemes, arriving at major themes providing coverage for identified clusters [3]. Through successive discussions, the authors iteratively refined the themes until no new themes emerged.

3.2 Findings

We initially set out to understand practitioner challenges with respect to NL prototyping. To our surprise, however, we found that early prototyping was rarely considered. When prototypes were developed, numerous practices, challenges, and compromises served to limit the validity of insights derived from the prototypes or their evaluation. Below we unpack key forces that give rise to these issues.

3.2.1 Prototyping slows development. In several cases, participants explained that, in considering the often tight product timelines, traditional prototyping and testing methods were perceived as too costly to perform—slowing down the otherwise fast pace of development. For example, P4 said "*a lot of prototyping that we do is, it takes a lot longer than we typically have.*" In another example, P6 said "*I think it's a little bit different within [search product], because things happen so fast that we tend to skip over prototypes. We do prototyping for really, really big things, but a lot of we do, it just goes straight to code that gets flighted.¹"*

3.2.2 Lack of tool support. The slow speed of early-stage prototyping was often attributed to a lack of established methods and tools. As noted above, participants often relied on 'flighting' to evaluate NL experiences. Flighting, however, requires a shipped product with an established audience and a robust instrumentation pipeline. In cases where flighting was not possible, practitioners felt they lacked comparable tooling for evaluating prototypes. For example, P3 remarked, "I don't know how you prototype that. Right? The equivalent for prototyping was to build the options [intelligent

¹Limited audience A/B testing of a feature in an already shipped product.

ID	Practitioner Role	Product Team	Exp.	Envisioned Feature in User Study (Task II)
P1*	User Researcher	Writing Assistance	1	-
P2	Program Manager	Writing Assistance	10	Text prediction in writing assistance
P3	Designer	Writing Assistance	5	Tone detection in writing assistance
P4	Design Manager	Conversational AI	10	Content consumption in conversational AI
P5*	Program Manager	Search	9	-
P6	Design Manager	Search	4	Recommended content in in-app search
P7	Program Manager	Search	15	Query completion in in-app search
P8*	Program Manager	Speech & Search	3	-
P9	Program Manager	Recommendation	15	Recommended designs in presentation app
P10	User Researcher	Recommendation	5	Query completion in in-app search
P11	User Researcher	Conversational AI	1	Recommended content in conversational AI
P12	User Researcher	Writing Assistance	2	Text detection in email communication

Table 1: Demography of study participants. *=participation in first study only; Exp=Experience in years of practice

features] and then flight them." Likewise—and again noting the fast pace of development—P4 explained, "this is a super ambiguous space. The most ambiguous space that I've ever worked in, in my years of working in design. We are literally making this up every day as we go along because things are moving so fast... there aren't any real rules. And we don't have a lot of the tools."

3.2.3 Hero scenarios and golden paths. In cases where prototypes were developed, participants noted several threats to the validity of insights derived from their evaluations. In particular, numerous participants discussed their team's focus on high-value 'hero' scenarios, which were often conceptualized, specified, and prototyped under ideal circumstances. This made it easy to overlook possible AI failures and error scenarios that deviate from this 'golden path'. For example, a PM in the writing assistance team remarked "we focus more on hero kind of scenarios." Similarly, P7 said "I think design still oftentimes is in the space of designing for kind of ideal solutions."

P8 described their experience releasing a hero experience and then only later addressing a known user experience issue: "for the MVP^2 that we shipped two years ago, we ignored the problem of the speech engine returning low confidence results. For the design which is just shipping now, we actually came up with a system that said, 'Well, when we get low confidence results back, we're going to provide a visual treatment to [the user]".

Failure to identify and address errors early incurs technical debt it can be costly to discover and address such errors later in development. Indeed, several participants described experiencing such costs. P3 noted, for example, that it became more difficult to collect rich qualitative feedback post deployment: "*[users] only react when there is something super positive or super negative that's happening to them*" while P1 emphasized the significant burden of reviewing and gaining meaningful insights from that feedback: "*how many of these results are we going to get a day? How are we going to parse through them and understand? It's a lot of burden on both ends.*" When issues are identified from post-deployment feedback, the prohibitive costs of addressing those issues can lead teams to reactively address only the most severe cases "unless it's like a super severe escalation… *it's going to be hard for us to flag it*" (P1). Moreover, the difficulty of updating an already deployed system often results in a slow turnaround time in improving the user experience "*if I do design today, I won't actually be able to see [user feedback] in an internal* [*deployment*] *in a product for almost three months. So, my ability to effect something is really delayed*" (P4).

3.2.4 It is difficult to anticipate errors. Unfortunately, when teams did attempt to identify potential errors and failure scenarios during early stage design and development, many participants described difficulties anticipating such failures, or approaching failures in a systematic way. For example, P1 said "it's so hard to know until the bad thing happens." Similarly, P9 described the difficulty accounting for the many ways an AI system might fail: "sometimes we may be able to identify a result that we're highly confident in. And in those cases we try to go for design... But we also need to account for the many, many times where we have very little context to go on where we may have very low confidence." Another participant, P5, described a trial-and-error approach to understanding errors early in design, saying "we started mucking around a little bit, rerunning things and then switched out some parameters. So, a little bit of trial and error."

3.2.5 It is difficult to prototype personalized experiences. A second threat to the validity of the insights derived from prototypes arises from the challenges of simulating personalized or highly-contextual experiences. For example, P9 said "that's not something that we normally do...[It] takes a lot of time and it's very personal. Like it's one thing to say, 'hey, come down to [company] and we'll pay you 50 bucks for your time'. And just, you know, use a computer for us. It's another thing to say, 'and by the way, bring your data along'. So that changes the conversation quite significantly. So that's why we don't do that." Similarly, P1 said "well, we can't learn about how people are going to react to it [AI writing assistant] until we ship it."

In the few cases where participants did describe efforts towards early prototyping of personalized experiences, they emphasized the limitations of current techniques for doing so. For example, some participants described using synthetic data (e.g., pre-canned text, email accounts, etc.) or simple rule-based models to simulate planned models for early testing. However, these participants also lamented that neither approach delivered AI experiences that came close to what users would experience in the context of their personal

²Minimum viable product

data. For example, P1 said "we simulated [the intelligent feature] very early on but like with [...] fake dummy writing, but the results just didn't have the depths we were really looking for. So that's why in the end it just worked best to not simulate it at all and just test out, like even if it's buggy. At least we'll have people's real writing so people can truly assess the quality of the results and the experience of how we surface these results."

3.2.6 Breakdowns in interdisciplinary communications. Finally, we note the development of AI-driven interactions is inherently crossdisciplinary. However, along with the lack of tools and techniques to support efficient and effective early prototyping, several of our interviewees described challenges collaborating and communicating across interdisciplinary teams.

In several cases, participants described impediments to iterative prototyping due to late and limited communication about identified issues. For example, P6 said "design was rarely involved in the conversations with [engineers], which led to kind of like a game of telephone, where like stuff would be happening that would impact the design over here, but nobody would tell the designers until a bunch of stuff had been done, and then [the engineers] would have to go redo it because it wasn't matching." In another example, P1 described feeling hesitant to raise issues late after an AI model is developed knowing the cost that might be incurred to address those issues "oftentimes it's just, these projects come from PM of like, this thing is happening, our data scientists already have a model, let's test what this looks like in real life. Like, understanding the usability and comprehension of it. And in a way, like, it just feels bad because... I feel like some of my research like, is crapping on a lot of really great work that our data scientists have done to perfect the models."

Participants also described challenges engaging across disciplinary boundaries due to varying levels of familiarity with AI and ML concepts, including understanding what can go wrong and what can be easily updated. For example, P2 said "*it's sometimes challenging so*, *if you talk to somebody who haven't been working in the AI domain... sometimes that's a challenge to communicate that as an idea... like communicating to designers and researchers to show, like what is the problem we may face?*" Similarly, P3 remarked that "*a better understanding of what are the things that you can push back, or you can change*" would help empower team members to contribute to the design and development of AI-based products.

In summary, our participants indicated that they did not often prototype new AI features or experiences, in part because they lacked tooling and believed it slowed down the pace of development. In cases where prototyping was done, a focus on golden paths and hero scenarios, together with the use of synthetic data, threatened the utility of the prototype, and the validity of user feedback. Each of these challenges was further exacerbated by breakdowns in cross-disciplinary communication. These findings suggest the need for low-cost tools to encourage proactive and systematic consideration of potential AI failures. In the next section we taxonomize AI failures within the context of user interaction with NL-based systems—a necessary first step in the development of the AI Playbook.

4 TAXONOMY OF NATURAL LANGUAGE FAILURES

Our interviews revealed that many practitioners were not aware of the range of errors possible in their AI application contexts. This rudimentary challenge prompted us to systematically taxonomize types of errors as a first step towards supporting AI prototyping. Given our participants' resistance to anything perceived to delay development, we elected to focus on errors that are either very common, or rare but very costly. Moreover, we focused on errors that directly affect the end-user as opposed to other users of the system, or other stakeholders. Finally, we narrowed our scope to errors apparent within a single session. We expected this would have the most impact in early prototyping because realistically simulating long-term failures is particularly difficult without highfidelity prototypes. While not exhaustive, this scoping allowed us to explore how consideration of common errors might be standardized in practice via a low-cost tool. In Section 7, we discuss how the AI Playbook may be extended to support other AI error scenarios.

To develop the taxonomy, we enlisted the help of eight computer science researchers with relevant expertise in human-AI interaction and natural language technologies. First, we collected an initial set of failures outlined in prior work or from our own AI development experiences to provide broad coverage of various NL scenarios (such as in the domains of information retrieval [22], discourse and dialogue management [2, 8, 13]). Through multiple meetings and small discussions, we then organized the failures according to Paek and Horvitz's adaptation of Clark's model [24], which we found helped establish grounding to reason about human-computer communication failures. We adapted their four constructs, including channel level (attempt initiate communication), signal level (understanding what behavior is intended as signal), intention level (understanding of semantics), and conversation level (in which a response is generated), to form the highest hierarchical layer of our taxonomy, namely: attention, input perception, understanding, and response generation errors (Table 2, leftmost column).

We then did a final round of iteration by walking through concrete NL tasks (e.g., extracting a meeting request from an email, booking a flight with a voice assistant, etc.) and expanding or modifying the taxonomy to account for potential points of failure. Finally, we generated guidance for simulating failures, building on techniques suggested in previous work (e.g., [2, 8, 13, 22]).

It is important to note that the taxonomy is designed to capture a broad range of scenarios. Not all failure types and error sources will apply to every product or feature. For example, attention/triggering scenarios are unlikely to apply in cases where the AI experience is explicitly and unambiguously invoked by the user. Likewise, generative response errors may not be relevant to systems that produce a ranked list over existing items. To this end, we evolved the taxonomy into the interactive AI Playbook tool, which leverages a survey instrument to recommend error scenarios that are relevant to a given application context. We describe this tool next.

5 AI PLAYBOOK

In light of practitioner reported challenges in prototyping AI, we designed the AI Playbook to support the following goals:

lable 2: laxonomy of natural language failures. laxonomy consists of failure types, sources and

Failure Type	Failure Source	Failure Scenario	Example Failure Scenario with NL Products
Attention (channel level)	Missed trigger	System fails to detect a valid triggering event.	[Scheduling assistant] fails to detect a meeting request in an email. [Voice assistant] fails to detect a wake word.
	Spurious trigger	System triggers in the absence of a valid trig- gering event (it triggers when not intended).	[Scheduling assistant] mistakes meeting minutes as a new meeting request. [Voice assistant] mistakes background speech as a wake word.
	Delayed trigger	System detects a valid triggering event, but responds too late to be useful.	[Scheduling assistant] detects a meeting request, but offers support only after the user has already manually scheduled the meeting.
	Truncation	System begins capturing input too late, or stops capturing input too early, and thus acts only on partial input.	[Voice assistant] interprets a pause as the end of an utterance, and answers before the user has finished speaking.
Perception (signal level)	Overcapture	System begins capturing input too early, or stops capturing input too late, and thus acts on spurious data.	[Voice assistant] captures user's intended query along with a few more words spoken after the query.
-	Noisy channel	User input is corrupted by spelling errors in written text, or background noise in audio input.	[Search engine] user misspells their search query. [Voice Assistant] background noise, such as music, makes speech less discernible.
	Transcription	System generates common transcription errors such as homonyms, homophones, plural word forms, etc.	[Automatic captioning service] fails to transcribe certain speech utterances by adding or removing inflected endings (e.g., -ing, -s, -ed).
	No understanding	System fails to map the user's input to any known action or response category.	[Virtual assistant] responds "I'm sorry, I don't know how to answer that question."
	Misunderstanding	System maps the user's input to the wrong category of action or response.	[Shopping assistant] mistakes an item refund request for an item exchange request.
Understanding (intention level)	Partial understanding	System correctly infers some aspects of the user's intent, (e.g., correct action or response category) but get some details wrong.	[Travel assistant] mistakes the origin for the destination. [Scheduling assistant] fails to consider a location's timezone in a meeting request.
	Ambiguity	There may be several reasonable interpreta- tions of the user's intent, leading to ambiguity. The system fails to correctly resolve ambiguity.	[Search engine] responds to the query "US Open" by returning results for tennis, when the user intended golf. [Scheduling assistant] sends a meeting request to the wrong "John".
	Action execution	System fails in executing the desired action.	[Travel assistant] attempts to re-book a trip by canceling and reissuing a ticket, but fails partway, leaving the reservation in an unknown state.
Pernonse	Generative response	System generates an incoherent, inappropriate, factually incorrect or partially correct response.	[Social chatbot] produces an offensive response to a user's input.
generation (conversation	Ranked list	System produces a ranked list with low precision, recall, or result diversity.	[Presentation design assistant] offers five recommended slide designs, but two designs are duplicates, or are otherwise indiscernible.
level)	Binary classification	System generates a false negative or false positive response.	[Spam filter] misclassifies an email as unsolicited spam. [Content moderation system] misclassifies a comment as obscene.
	Multi-class classifica- tion	System fails to produce correct classifications among close or distant categories.	[Email client] classifies a receipt as a promotional email.

- Provide a means to discover non-ideal, error conditions that are outside of the golden path scenario.
- Provide actionable and contextually-relevant guidance on how to simulate these scenarios.
- Provide a means to explore a range of options and consequences of in-the-moment decisions.
- Provide the above-mentioned features as efficiently, and low-cost as possible.

Our taxonomy of NL failures provided the necessary content and structural support to realize these goals in the form of an interactive web-based tool. Practitioners interact with the tool by answering a series of questions to describe their envisioned AI scenario. While answering, the *Help Center* displays definitions and examples to support varying levels of AI expertise and aid in interdisciplinary communication. In addition, the *Scenario* section interactively builds up relevant scenarios to test, allowing users to efficiently explore the consequences of different design choices. Finally, the Playbook outputs a report of recommended test scenarios along with contextualized explanations and examples to further aid in developing and testing prototypes.

We depict the AI Playbook in Figure 1, and describe its key features in the sections below. The AI Playbook is also available online at https://aka.ms/haxplaybook.

5.1 Interactive Survey

Using the taxonomy as a foundation, we designed an *Interactive Survey* (Figure 1-A) that asks practitioners to walk-through their envisioned product or feature by answering questions about the type of system they are designing, input modality, trigger, delimiter, and the form of the expected response. Branching logic ensured that the survey questions, examples, actionable advice, and help, were relevant to the current context. For example, if users indicated their feature used speech as an input, subsequent questions and responses were phrased accordingly.

AITIMYDOOK		
Page 2 of 6	Help Center	D Input Errors ()
What is the primary input modality for your	() Speech	Your system will take speech as its primary input modality. Here are some scenarios that you can consider including in your user testing protocol.
Text Speech	Speech input is widely used in conversational assistants such as home automation systems, dictation and language translation services.	Speech recognition error results in wrong input Speech recognition is often a common source of error in technologies that rely on language intelligence. Here are some
	V Example	Ways to simulate space in acception errors: Use an automated speech-to-test transcriber to convert user's utterance to test. Use any of the four techniques to manipulate the user's utterance.
Previous Next	A voice assistant is a user interface that allows a user to complete an action or retrieve information simply by speaking a command.	 Innection: removing inneces enlargs (e.g., -ing, -s, -ed) or arcsis (tre) Substitution: regioning utterates with a homo-phone/-nym, or words that share the same pronunciation, regardless of spelling. Insertion: adding inflected endings (e.g., -ing, '-s', -ed') Extension Splitting inflected endings (c-s') from words (cats') to add a new word (e.g., catg -> cat git)
. Л		System ignores speech input
		Speech recognizers can ignore a user's utterance due to a disruptive noise signal. Simulate this error by turning off audio-visual feedback that indicates proper input registration.
5 Scenarios		System generates a non-response
Correct Operation 🕕	^	Speech recognizers can also fail to make sense of a user's ulterance due to a disruptive noise signal. Simulate this error by providing feedback to the user which indicates that the system has failed to register the user's input. For exemple, a system can respond with "Sory, I did not here what you said".
System demonstrates normal operation		
Input Errors 3	^	Trigger Errors 3 Any uncertainty in knowing when to invoke the intelligent feature can sometimes result in triggering wrongfully, or triggering the system to no error or othen.
Your system will take speech as its primary input modality. Here are some sc user testing protocol.	enarios that you can consider including in your	Here are some scenarios that you can consider including in your user testing protocol.
Speech recognition error results in wrong input		System triggers too early
		Your system can trigger too early when it has a problem perceiving an input signal.
System ignores speech input		Simulate the end by diministry menung a series are span between the user's input and system reedback. Choose a short time span (e.g., to 2 scood) between the beginning of the user's input and system feedback that indicates receipt of input. Use this time span as a condition to trigger system feedback.
I		

Figure 1: Screenshot of the AI Playbook user interface. A: Interactive Survey with buttons that allow users to navigate between different question and answer pairs; B: Help Center displays supportive information (e.g., description and examples) tailored to the user's selected answer choice; C: Scenario Builder allows users to view potential failure scenarios based on their selected answer choice; D: Playbook Report allows users to view the full list of scenarios along with detailed guidance on how to simulate them.

5.2 Help Center

To support practitioners' understanding of the concepts indicated in the questions, the *Help Center* displays general descriptions of NL concepts along with specific examples of how the concepts are realized in real world products (1-B). As noted above, this content is contextually-relevant and based on the user's responses to survey questions.

5.3 Scenario Builder

The *Scenario Builder* (1-C) enumerates relevant scenarios for practitioners to consider, prototype, and test in response to their survey responses. Users can use the *Scenario Builder* to interactively explore how different design choices impact likely failure scenarios that should be tested.

5.4 Playbook Report

The *Scenario Builder* can be expanded into a detailed report (1-D). The report provides actionable guidance on how to simulate each scenario. A partial report can be accessed at any time by expanding the items in the *Scenario Builder*, and the full report is generated automatically at the completion of the survey.

6 EVALUATION

After developing the AI Playbook, we presented it to participants from our formative interview study to assess if and how it might meet their stated needs and for further discussion. We describe the design and results from this qualitative evaluation below.

6.1 Study Setting

6.1.1 Recruitment. We invited participants from our previous interview study to explore and provide feedback on the AI Playbook, allowing us to evaluate if our design was meeting their needs as intended. 10 out of 12 previous interview study participants responded to an email study invitation, and nine participants consented and completed the follow-up user study.

6.1.2 *Procedure.* The AI Playbook was developed to run on a web browser, and the application was hosted via a local server on the facilitator's (first author) computer. We used a video conferencing tool to share the facilitator's screen and gave study participants access to control the mouse cursor and keyboard to remotely interact with the AI Playbook system.

We asked participants to complete two tasks using the AI Playbook. In Task I, we asked participants to use the tool to facilitate prototyping a fictional chatbot system designed to help website visitors search for an apartment. In Task II, we asked participants to apply the tool to a feature or system that their product team was currently in the process of conceptualizing or prototyping. If participants were not currently in that phase of design, we asked them to consider the most recent real-world experience that applied. Table 1 lists the features participants considered in Task II.

Participants were encouraged to think out loud as they completed each task. A semi-structured interview was conducted at the end of Task II. Through the interviews, we elicited qualitative feedback spanning topics including: concrete aspects of the tool, suitability and applicability of the tool to their current work practices, perceived or anticipated value of the tool, and opportunities for further development. The entire procedure took 60 minutes.

Sessions were screen-recorded, and later transcribed. Immediately after their completion of each task, we downloaded a PDF copy of the generated reports. We also immediately sent copies of the reports to the participants for additional scrutiny, and to ground the subsequent semi-structured interviews. Data collection took place between June 22 and July 10, 2020.

6.1.3 Analysis. We obtained 479 minutes of screen and audio recorded data, which were transcribed verbatim. All qualitative analysis of transcripts was conducted in similar fashion to the prior interview study. Any specific references made with regards to the design or contents of the AI Playbook in the transcripts have been cross-referenced with relevant video footage of participants' usage. We then took notes from our observations of screen recordings and included them in the thematic analysis.

6.2 Findings

We now turn to share key insights from our evaluation of the AI Playbook. Our findings are organized such that they mirror the above-mentioned design desiderata, then delve into other aspects uncovered in the interviews.

6.2.1 Consider scenarios outside of the golden path. After their use of the AI Playbook, many participants were convinced that a tool such as AI Playbook could help them consider atypical situations or failures that are often outside the radar of the golden path scenario. For example, P6 commented: "I think so often we only test the perfect scenarios. Like we only design and test the end-to-end experience exactly as it is supposed to work, and I think this [AI Playbook] would actually help us to test where it might break [...]. So if something didn't trigger when it was supposed to, how much does that impact a user? Would it impact how much they want to use it in the future? Or something like that. And we could actually design and test for that instead of designing these closed loop perfect end-to-end things. This gives us some flexibility and I think it would change the process."

Participants also suggested that the tool can help product teams navigate the error space systematically, instead of implementing ad hoc approaches. For example, P3, a designer, reflected on her team's struggled to generate mistakes in an arbitrary, ad hoc manner: "this is something that we did manually at first [...] we made mistakes on purpose and wanted to figure out if people will catch on to that and how will they react." Likewise, P7 reflected: "I've not had the experience of like going through an audit of the potential errors and saying 'Okay, for each one of these things, what are you going to do from a design point of view?' It oftentimes is more organic. It kind of happens ad hoc you know? Design puts red lines down. [...] Engineers realize, 'hey, we're going to run into this particular error.' We go back to design. [...] There's lots of efficiency that is lost there. So something like this can help with that efficiency."

In addition to helping teams systematically navigate the error cases, participants saw that the tool could help standardize the types of errors that practitioners can include in prototyping and testing. P2 shared his excitement about the potential opportunities: "What I see would be beneficial from this is, you will standardize the error case design. You could potentially help to improve the the PM spec a lot." Likewise, P9 explained: "I see it as a checklist tool to make sure that like we are able to force a consistent bar across different teams in different organizations."

6.2.2 Provide actionable and contextually-relevant guidance. While the AI Playbook was deemed useful in stepping off the golden path, it was also found to provide actionable guidance that participants felt they could immediately apply. For instance, P4, a design manager explained: 'we deal with all this stuff everyday. But yeah, there's some good notes here about how to test and validate for these things. So yeah, very real world for us." Likewise, P2, a program manager, remarked: "I think this level of details are enough so we can do a proper documentation and then guide engineers to do their work." Similarly, P6, also a design manager, said the details were enough to impact how she would approach writing her user study protocols: "even if we are thinking about [error scenarios] it's not to the depth that this Playbook surfaces. [...] I would rewrite so many studies if I had in this list of scenarios."

Even when guidance was deemed not to be perfectly applicable, some participants felt they could easily adapt and tailor the guidance to their situation. P6 expressed this sentiment best as follows: "to me, this serves as a really good example of what to do. It may not be the exact solution for every feature. That's fine. It gives me the right direction to come up with my own solution."

6.2.3 Provide a means to explore the full range of options and the consequences of decisions. Some participants also appreciated the comprehensive nature of the tool, and the ability to immediately observe the consequences of their actions. For example, P11 explained: "I think it's a very helpful tool. Like when you think about it, it gives a very concrete way to articulate like all the ways in which the system could fail." Likewise, P4 noted "the fact that it's so short is nice, because if you got to the end and the recommendations were kind of missing a piece of what you were doing you would sort of realize, 'oh, maybe I should try a different category and see if I get something that feels a little closer."

Nevertheless, our design may have erred too far in favor of exploration. Some participants left wondering about the scenarios omitted from the report, and longed for comprehensive list of errors—including those deemed not to apply. For example, P10 noted an analogical reference to her own experience testing a recommendation product, "a lot of time [users] would be like, 'this is great. It's relevant, but like I want to see everything' [...] and [the short list] is not giving them the confidence that they're finding like the best. So they needed to know like 'how relatively good is this'?" Likewise, P3 offered careful advice that too much contextualization would not be ideal for new and inexperienced practitioners: "I imagine that

if someone is starting to work in that area, [...] they wouldn't even know where to look for the problems. They will have to go through all the process in order to say 'OK, we didn't check for that. Oh, I wish I knew!"

6.2.4 Deliver features as efficiently, and low-cost as possible. After using the AI Playbook, all participants appreciated the way in which the AI Playbook's format allowed them to efficiently cover a lot of ground in three to five minutes of their time. P4 expressed this sentiment best: "You know, that's pretty low investment, right? For a high [value] trade off. Yeah, I really enjoyed this." Some participants attributed the high efficiency to the tool's ability to support their easy navigation and exploration. For example, P6 expressed her enthusiasm for the 'help center' feature as follows: "This, I mean, that help center was my favorite part of this whole process. [...] I'm sure you saw me change my answers a bunch of times as I would read the different things because it helped me [...] I really liked the examples in there too, because that's what actually helped me the most."

6.2.5 Bridge considerations across teams and disciplines. While we did not explicitly include support for team communication and collaboration in our desiderata, many saw the potential value that the AI Playbook can serve in bridging the culture and language of practice. For example, P6 remarked: "we all feel like we're sometimes speaking different languages, and this [AI Playbook] just instantly like levels it out and says, here's your report [...] I feel like it puts everybody on the same level in terms of being able to use the same terminology to have very clear scenarios laid out." More simply, P10 noted "I would use [the Playbook] as a tool to connect the disciplines around these issues", and P3 added the the AI Playbook served as a "kind of a boundary object" between interdisciplinary teams.

Going further, some participants emphasized the value of the AI Playbook in encouraging shared accountability and consensus building. P6 in particular emphasized this point, saying "I actually feel it [...] could be a great exercise for everyone on the team to do this individually and then we could compare how we all answered it. I think that also then identifies where we may not be on the same page." Likewise, P9 suggested a potential for the tools to drive accountability, perhaps automatically scheduling review meetings. He explained, "I think the one thing to really be mindful is to not let this be a tool where people just go through the motions and have a result and then just kind of move on with their lives [...] So at the very least, there's some sort of accountability in making sure that the users of the system actually go through it and actually think about it." P4 also agreed with this sentiment and suggested making use of existing group meetings such as brown bag events or workshops to reinforce product teams' group knowledge of the error scenarios. "not just dump it in email and say 'check it out', but more use it as one of our brown bags or one of our team hands-on workshops or something to get people more engaged in the conversation."

Finally, an important value that participants saw in their use of the AI Playbook is its potential to empower UX practitioners to communicate their opinions more effectively with interdisciplinary teams. P6 shared her excitement, "*it felt like*, 'oh, I could actually go and have a conversation with a ML person about these issues' and feel confident going into that—which I think would just build better team dynamics." Likewise P11, expressed a similar sentiment, and considered the tool as an added resource when communicating with designers. She said, "I could see this as if I'm having a conversation with the designers who are actually building this prototype and saying. Hey um. You know, like 'what happens if we're making bad recommendations' or 'I think we should include some scenarios that maybe don't [succeed]."

7 DISCUSSION AND FUTURE WORK

In the following sections, we reflect on limitations of this research and point to future opportunities for further discussion and investigation.

7.1 Generalizing Our Findings

While we carried out this research project in a single large software company (comprising numerous disparate teams and organizations), we recognize that our findings cannot directly speak to the practices of other companies. However, people in our industry are highly mobile, often transitioning between teams and companies, and bringing their practices with them.

While prior work has articulated several challenges of prototyping AI experiences [7, 30, 31], our research sheds new light to this discussion, suggesting that practitioners often skip early-stage interaction prototyping. We suspect the lack of early prototyping and consideration of errors is related to a focus on one or few hero scenarios, perhaps reached through idealized paths through the interface. This often occurs in the context of agile and lean project management to deliver new features to customers of existing products. While this approach advocates for shipping fast and learning from customer feedback, our study shows that teams find identifying and addressing user experience issues post-deployment costly and ineffective. We conjecture that this could be to prevent disrupting existing customers and difficulty teasing apart feature-level issues from the rest of the product.

To this end, we hypothesize that similar practices are observed elsewhere in the industry. To confirm or reject this hypothesis, and to better understand cross-industry practices, we welcome future work in this space.

7.2 Extending to Other AI Scenarios

The AI Playbook covered a limited set of AI failure scenarios, focusing on natural language-based products, yet we imagine the same approach could be used for a wide variety of AI-based systems (e.g., scenarios that utilize computer vision).

Likewise, the AI Playbook does not consider AI failure types that are contingent on the end-user's interpersonal, sociocultural, and societal context; nor does it consider systematic errors that result in unethical and unfair outcomes such as favoring one user group over another. Indeed, our participants talked about extending the AI Playbook to cover such scenarios, and even provided examples of real-world situations that resulted in racially or socially insensitive outcomes (e.g., suggesting the response "Great!" to a sad message about a cancer diagnosis). Despite the importance of such errors, a full understanding of this space is an active area of research [14, 19, 21], and we welcome expanding the AI Playbook to better predict in which circumstances such errors may arise, and to provide better guidance on how they can be simulated and ultimately mitigated. The AI Playbook also does not consider errors that manifest longitudinally or via personalization. Such scenarios pose the challenge of prototyping interactions that depend on a history of past activity or data that may not yet exist. While synthetic data, and use of personas offer possible solutions, it may be difficult for people to detect errors, or assess their severity, when the data is not their own. Future research should explore approaches to extend the Playbook to cover such usage sessions.

Additionally, as noted in Section 3, personalized experiences inevitably require access to users' personal data, yet such access invites other technical and ethical considerations around privacy. Privacy considerations in existing products limit what logs can be collected, and what data can be directly inspected for this type of qualitative exploratory analysis.

Finally, the AI Playbook considers only scenarios in which the end-user is directly impacted by an error. In other scenarios, such as adversarial attacks on ML models [32], it is important to consider a broader range of stakeholders.

In considering each of the above-mentioned cases, we note that the platform is extensible, and that teams can adapt and extend the Playbook to accommodate such scenarios.

7.3 Standardizing Terminology

Though participants noted that the AI Playbook had the potential to standardize communication across teams and disciplines, one major challenge was revealed: people struggled even to express their own conceptualizations and characterizations of some AI systems. Let us consider the following scenario:

Upon a user opening a chat window, an intelligent chatbot shows a recommended set of actions even before the user has entered any text.

Is this a conversational system or a recommendation system? Is the AI feature initiated by the user (by opening the chat window) or by the system? Our findings suggest that answers to these questions may vary widely even among industry experts in AI product design. The lack of precision and consistency poses challenges for developing a tool like the AI Playbook, but, more importantly, could lead to potential misunderstandings and confusion among practitioners. We thus envision future work in standardizing the language used to discuss concepts that arise at the intersection of interaction design and machine learning.

7.4 Expanding Accountability

Finally, though the AI Playbook provided concrete actionable guidance, our participants identified the benefits of being even more forceful, holding designers and developers accountable for addressing specific scenarios. Specifically, it was suggested that the AI Playbook be integrated into compliance checkers and task management systems that already gate the distribution of code within the company. Alternatively, the Playbook report could be used as a checklist, where issues are prioritized and addressed prior to moving a design from one phase of development to the next. This raises the additional interesting possibility of dynamically adapting to specific phases of development.

8 CONCLUSION AND ACKNOWLEDGMENTS

Despite the promise of AI to improve the user experience of language-based AI products, errors will remain an inevitable byproduct of the inclusion of AI features. While it is important to prevent or eliminate failures in constructing machine-learned models, practitioners need tools and techniques to plan for unexpected failures in early prototyping and testing of AI user experiences.

To this end, our paper contributes an empirical understanding of the existing prototyping practices of teams developing intelligent language-based software products, as well as the practical challenges that product teams face in so doing. Based on these findings, we present a taxonomy of common errors, and an interactive tool the AI Playbook—that provide practical guidance on which error scenarios to consider in their designs, and how to simulate such errors in their prototypes.

Through the AI Playbook, our research demonstrates the potential to minimize technical debt by surfacing non-ideal scenarios early in the design process. The benefits of such tool extend beyond prototyping and testing and shows promising returns in design practice, such as promoting standardization and collaboration across disciplines.

We thank all of our study participants and researchers who contributed to our well-rounded understanding of natural language failures and their implications for prototyping practice. We believe that exciting opportunities await HCI researchers and practitioners to expand our knowledge and articulation of AI failures.

REFERENCES

- [1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In Conference on Human Factors in Computing Systems - Proceedings. Association for Computing Machinery, New York, New York, USA, 1–13. https: //doi.org/10.1145/3290605.3300233
- [2] Dan Bohus and Alexander I Rudnicky. 2005. Sorry, I Didn't Catch That! An Investigation of Non-Understanding Errors and Recovery Strategies. In 6th SIGdial workshop on discourse and dialogue.
- [3] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [4] Bill Buxton. 2010. Sketching user experiences: getting the design right and the right design. Morgan kaufmann.
- [5] Herbert H Clark. 1996. Using language. Cambridge university press.
- [6] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. 2017. Calendar.help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop. In Conference on Human Factors in Computing Systems - Proceedings, Vol. 2017-May. Association for Computing Machinery, New York, NY, USA, 2382–2393. https://doi.org/10.1145/3025453.3025780
- [7] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX design innovation: Challenges for working with machine learning as a design material. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 2017-May. Association for Computing Machinery, New York, NY, USA, 278– 288. https://doi.org/10.1145/3025453.3025739
- [8] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. 2018. Automatic speech recognition errors detection and correction: A review. In Procedia Computer Science, Vol. 128. Elsevier B.V., 32–37. https://doi.org/10.1016/j.procs.2018. 03.005
- [9] Jerry Alan Fails and Dan R. Olsen. 2003. A design tool for camera-based interaction. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery (ACM), New York, New York, USA, 449– 456. https://doi.org/10.1145/642611.642690
- [10] Norman M. Fraser and G. Nigel Gilbert. 1991. Simulating Speech Systems. Computer Speech and Language 5, 1 (1 1991), 81-99. https://doi.org/10.1016/0885-2308(91)90019-M
- [11] Google PAIR. 2019. People + AI Guidebook. https://pair.withgoogle.com/ guidebook/

- [12] John D Gould, John Conti, and Todd Hovanyecz. 1983. Composing Letters With a Simulated Listening Typewriter. *Commun. ACM* 26, 4 (4 1983), 295–308. https://doi.org/10.1145/2163.358100
- [13] Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015. Towards taxonomy of errors in chat-oriented dialogue systems. In Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue. 87–95.
- [14] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3290605.3300830
- [15] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In Conference on Human Factors in Computing Systems - Proceedings. ACM Press, New York, New York, USA, 159–166. https://doi.org/10.1145/302979.303030
- [16] IBM. 2019. IBM Design for AI: Conversation planning. https://www.ibm.com/ design/ai/conversation/planning/
- [17] J. F. Kelley. 1984. An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications. ACM Transactions on Information Systems (TOIS) 2, 1 (1 1984), 26–41. https://doi.org/10.1145/357417.357420
- [18] Walter S. Lasecki, Juho Kim, Nicholas Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 2015. Apparition: Crowdsourced user interfaces that come to life as you sketch them. In Conference on Human Factors in Computing Systems - Proceedings, Vol. 2015-April. Association for Computing Machinery, 1925–1934. https://doi.org/10.1145/2702123.2702565
- [19] Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. 2020. Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376445
- [20] Sean McGregor. 2020. Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database. arXiv preprint arXiv:2011.08512 (2020).
- [21] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency. Association for Computing Machinery, Inc, New York, New York, USA, 220–229. https: //doi.org/10.1145/3287560.3287596
- [22] Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. ACM Transactions on Information Systems 21, 2 (4 2003), 133-154.

https://doi.org/10.1145/763693.763694

- [23] Tim Paek. 2003. Toward a taxonomy of communication errors. In ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems.
- [24] Tim Paek and Eric Horvitz. 2000. Conversation as Action under Uncertainty. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'00). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 455–464.
- [25] Laurel Riek. 2012. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. Journal of Human-Robot Interaction (8 2012), 119–136. https://doi.org/10.5898/jhri.1.1.riek
- [26] Stephan Schlögl, Gavin Doherty, and Saturnino Luz. 2015. Wizard of Oz Experimentation for Language Technology Applications: Challenges and Tools. Interacting with Computers 27, 6 (11 2015), 592–615. https://doi.org/10.1093/iwc/iwu016
- [27] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology. ACM Press, New York, New York, USA, 159–168. https://doi.org/10. 1145/1294211.1294238
- [28] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T Iqbal, and Jaime Teevan. 2019. Sketching Nlp: A Case Study of Exploring the Right Things to Design With Language Intelligence. In Conference on Human Factors in Computing Systems -Proceedings. Association for Computing Machinery, New York, New York, USA, 1–12. https://doi.org/10.1145/3290605.3300415
- [29] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating how experienced UX designers effectively work with machine learning. In DIS 2018 - Proceedings of the 2018 Designing Interactive Systems Conference. Association for Computing Machinery, Inc, New York, NY, USA, 585–596. https://doi.org/10.1145/3196709.3196730
- [30] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-Examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376301
- [31] Qian Yang, Jina Suh, Nan Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In DIS 2018 - Proceedings of the 2018 Designing Interactive Systems Conference. Association for Computing Machinery, Inc, New York, NY, USA, 573–584. https://doi.org/10.1145/3196709.3196729
- [32] Tomáš Zemčík. 2020. Failure of chatbot Tay was evil, ugliness and uselessness in its nature or do we judge it through cognitive shortcuts and biases? AI and Society (2020). https://doi.org/10.1007/s00146-020-01053-4