Intelligent Conversational Agents for Ambient Computing: A keynote at SIGIR 2022

Ruhi Sarikaya
Amazon Alexa AI
USA
rsarikay@amazon.com

Abstract

We are in the midst of an AI revolution. Three primary disruptive changes set off this revolution: 1) increase in compute power, mobile internet, and advances in deep learning. The next decade is expected to be about the proliferation of Internet-of-Things (IoT) devices and sensors, which will generate exponentially larger amounts of data to reason over and pave the way for ambient computing. This will also give rise to new forms of interaction patterns with these systems. Users will have to interact with these systems under increasingly richer context and in real-time. Conversational AI has a critical role to play in this revolution, but only if it delivers on its promise of enabling natural, frictionless, and personalized interactions in any context the user is in, while hiding the complexity of these systems through ambient intelligence. However, current commercial conversational AI systems are trained primarily with a supervised learning paradigm, which is difficult, if not impossible, to scale by manually annotating data for increasingly complex sets of contextual conditions. Inherent ambiguity in natural language further complicates the problem. We need to devise new forms of learning paradigms and frameworks that will scale to this complexity. In this talk, we present some early steps we are taking with Alexa, Amazon's Conversational AI system, to move from supervised learning to self-learning methods, where the AI relies on customer interactions for supervision in our journey to ambient intelligence.

Date: 14 July 2022.

1 Background for Ambient Computing

For decades, the paradigm of personal computing was a desktop machine. Then came the laptop, and finally mobile devices so small we can hold them in our hands and put them in our pockets, which felt revolutionary. All these devices, however, are alike in that they tether you to a screen. You need to physically touch them to use them, which does not seem natural or convenient in a number of situations. So what comes next?

The most likely answer is the Internet of Things (IoT) and other intelligent, connected systems and services. What will the interface with the IoT be? Will you need a separate app on your phone for each connected device? Or when you walk into a room, will you simply speak to the device you

want to reconfigure? At Alexa, we're betting that conversational AI will be the interface for the IoT. And this will mean a shift in our understanding of what conversational AI [Sarikaya, 2017].

In particular, the IoT creates new forms of context for conversational-AI models. By "context", we mean the set of circumstances and facts that surround a particular event, situation, or entity, which an AI system can exploit to improve its performance.

In particular, context can help resolve ambiguities. Here are some examples of what we mean by context:

- **Device state:** If the oven is on, then the question "What is the temperature?" is more likely to refer to oven temperature than it is in other contexts.
- **Device types:** If the device has a screen, it's more likely that "play Hunger Games" refers to the movie than if the device has no screen.
- **Physical/digital activity:** If a customer listens only to jazz, "Play music" should elicit a different response than if the customer listens only to hard rock; if the customer always makes coffee after the alarm goes off, that should influence the interpretation of a command like "start brewing".

2 Self-Learning

IoT is the fabric of ambient computing and we believe Conversational AI will be the natural interface to interact with these systems. Then, the key question to answer is how to train the conversational agents, such as Alexa, in the ambient computing set-up. Our answer is self-learning. By self-learning, we mean a framework that enables an autonomous agent to learn from customer-system interactions, system signals, and predictive models.

User-system interactions can provide both implicit and explicit feedback. Alexa already handles both. If a customer interrupts Alexa's response to a request - a "barge-in", as we call it - or rephrases the request, that's implicit feedback: when aggregated across multiple users, it suggests that the request wasn't processed correctly the first time. Customers can also explicitly teach Alexa how to handle particular requests. This can be customer-initiated, as when customers use Alexa's interactive-teaching capability, or Alexa-initiated, as when Alexa asks, "Did I answer your question?"

The great advantages of self-learning are that it doesn't require data annotation, so it scales better and protects customer privacy; it minimizes the time and cost of updating models; and it relies on high-value training data, because customers know best what they mean and want. Within self-learning area, we have a few programs targeting a different application of self-learning including automated ground truth annotation generation, defect reduction, teachable AI, and root cause of failure determination.

2.1 Verity

At Alexa, we have launched a multiyear initiative to shift Alexa's ML model development from a manual-annotation-based to a primarily self-learning-based approach. We call this initiative Verity [Gupta et al., 2021]. The fundamental challenge of Verity is that we need to convert

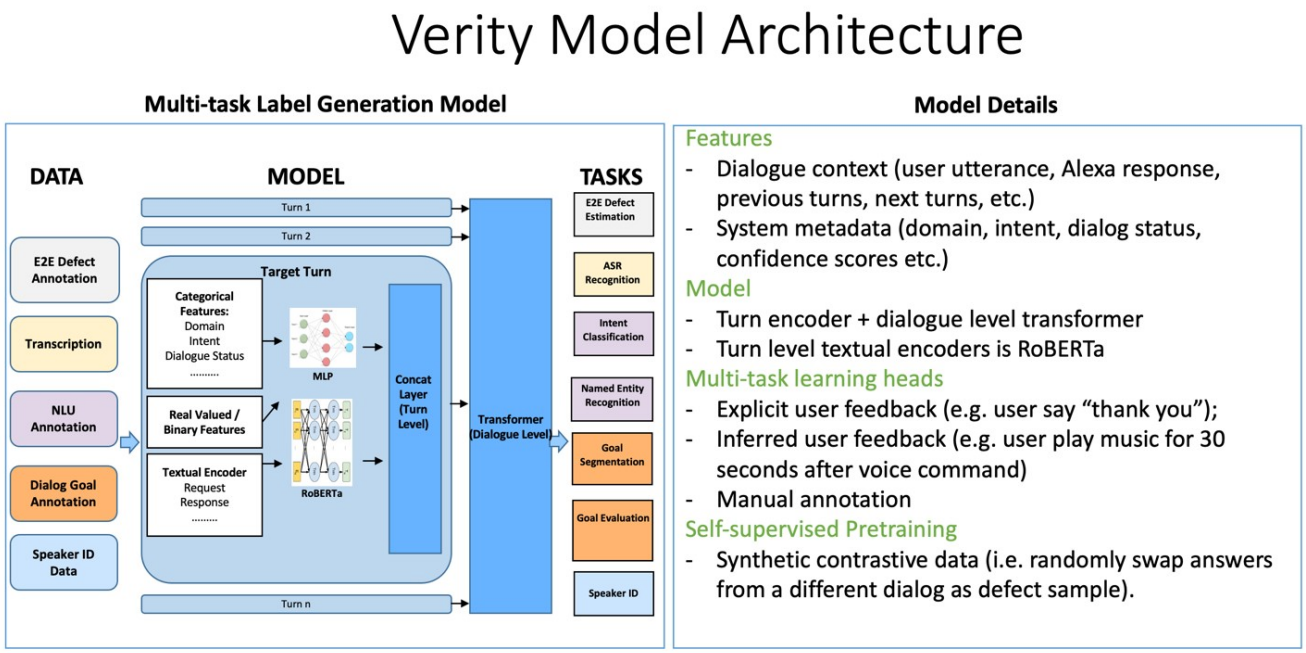

Figure 1: The ground truth generation model converts customer feedback, which is often binary or low dimensional, into high-dimensional synthetic labels.

customer feedback, which is often binary or low dimensional (yes/no, defect/non-defect), into high-dimensional synthetic labels such as transcriptions and named-entity annotations.

Accordingly, Verity has two major components: (1) an exploration module and (2) a feedback collection and label generation module. In Fig. 1, we show the architecture of the label generation model:

The input features include the dialogue context (user utterance, Alexa response, previous turns, next turns), categorical features (domain, intent, dialogue status), numerical features (number of tokens, speech recognition and natural-language-understanding confidence scores), and raw audio data. The model consists of a turn-level encoder and a dialogue-level Transformer-based encoder. The turn-level textual encoder is a pretrained RoBERTa model [Liu et al., 2019].

We pretrain the model in a self-supervised way, using synthetic contrastive data. For instance, we randomly swap answers from different dialogues as defect samples. After pretraining, the model is trained in a supervised fashion on multiple tasks, using explicit and implicit user feedback. We evaluate the label generation model on several tasks. Two of these are goal segmentation, or determining which utterances in a dialogue are relevant to the accomplishment of a particular task, and goal evaluation, or determining whether the goal was successfully achieved.

As a baseline for these tasks, we used a set of annotations each of which was produced in a single pass by a single annotator. Our ground truth, for both the model and the baseline, was a set of annotations each of which had been corroborated by three different human annotators. Our model's outputs on both tasks were comparable to the human annotators': our model was slightly more accurate but had a slightly lower F1 score. Since we can run the model on the entire

utterance traffic, we can set a higher threshold and exceed human performance significantly and still achieve much larger annotation throughput than the one manually labeled data.

In addition to the goal-related labels, our model also labels utterances according to intent (the action the customer wants performed, such as playing music), slots (the data types the intent operates on, such as song names), and slot-values (the particular values of the slots, such as "Purple Haze"). As a baseline for slot and intent labeling, we used a RoBERTa-based model that didn't incorporate contextual information, and we found that our model outperformed it across the board.

2.2 Self-learning-based Defect Reduction

Three years ago, we deployed a self-learning mechanism that automatically corrects defects in Alexa's speech recognition and interpretation of customer utterances based purely on implicit signals. This mechanism - unlike Verity - doesn't involve retraining Alexa's natural language understanding models. Instead, it overwrites those models' outputs, to improve their accuracy. There are two ways to provide rewrites [Ponnusamy et al., 2020]:

- **Precomputed rewriting** produces request-rewrite pairs offline and loads them at runtime. This process has no latency constraints, so it can use complex models, and during training, it can take advantage of rich offline signals such as user follow-up turns, user rephrases, Alexa responses, and video click-through rate. Its disadvantage is that at runtime, it can't take advantage of contextual information.
- **Online rewriting** leverages contextual information (e.g., previous dialogue turns, dialogue location, times) at run time to produce rewrites. It enables rewriting of long-tail-defect queries, but it must meet latency constraints, and its training can't take advantage of offline information.

2.2.1 Precomputed rewriting

We've experimented with two different approaches to precomputing rewrite pairs, one that uses pretrained BERT models and one that uses absorbing Markov chains.

In Fig. 2 we illustrate the BERT-based approach [Wang et al., 2021]. At left is a sample dialogue in which an Alexa customer rephrases a query twice. The second rephrase elicits the correct response, so it's a good candidate as a rewrite of the initial query. The final query is not a rephrase, and the rephrase extraction model must learn to differentiate rephrases from unrelated requests.

We cast rephrase detection as a span prediction problem, where we predict the probability that each token is the start or end of a span, using the embedding output of the final BERT layer. We also use timestamping to threshold the number of subsequent customer requests that count as rephrase candidates.

The second pre-computed rewriting method, described in Fig. 3, is absorbing Markov chains to extract rewrite pairs from rephrase candidates that recur across a wide range of interactions. A Markov chain models a dynamic system as a sequence of states, each of which has a certain probability of transitioning to any of several other states. An absorbing Markov chain is one that

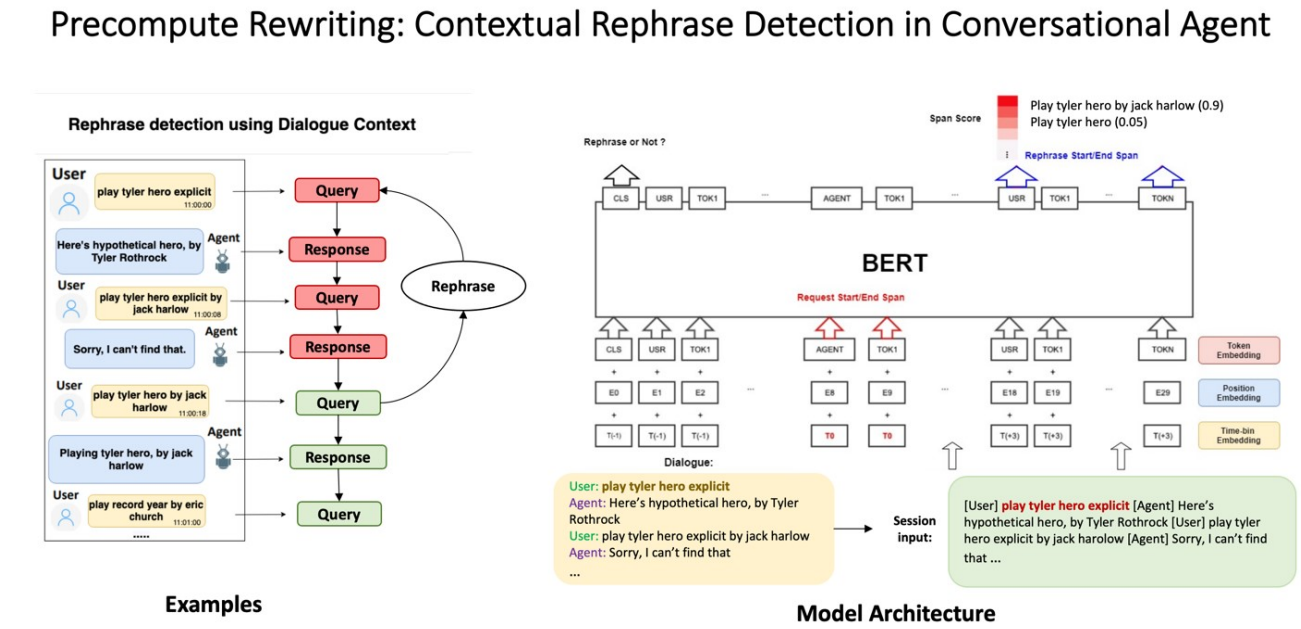

Figure 2: The contextual rephrase detection model casts rephrase detection as a span prediction problem, predicting the probability that each token is the start or end of a span.

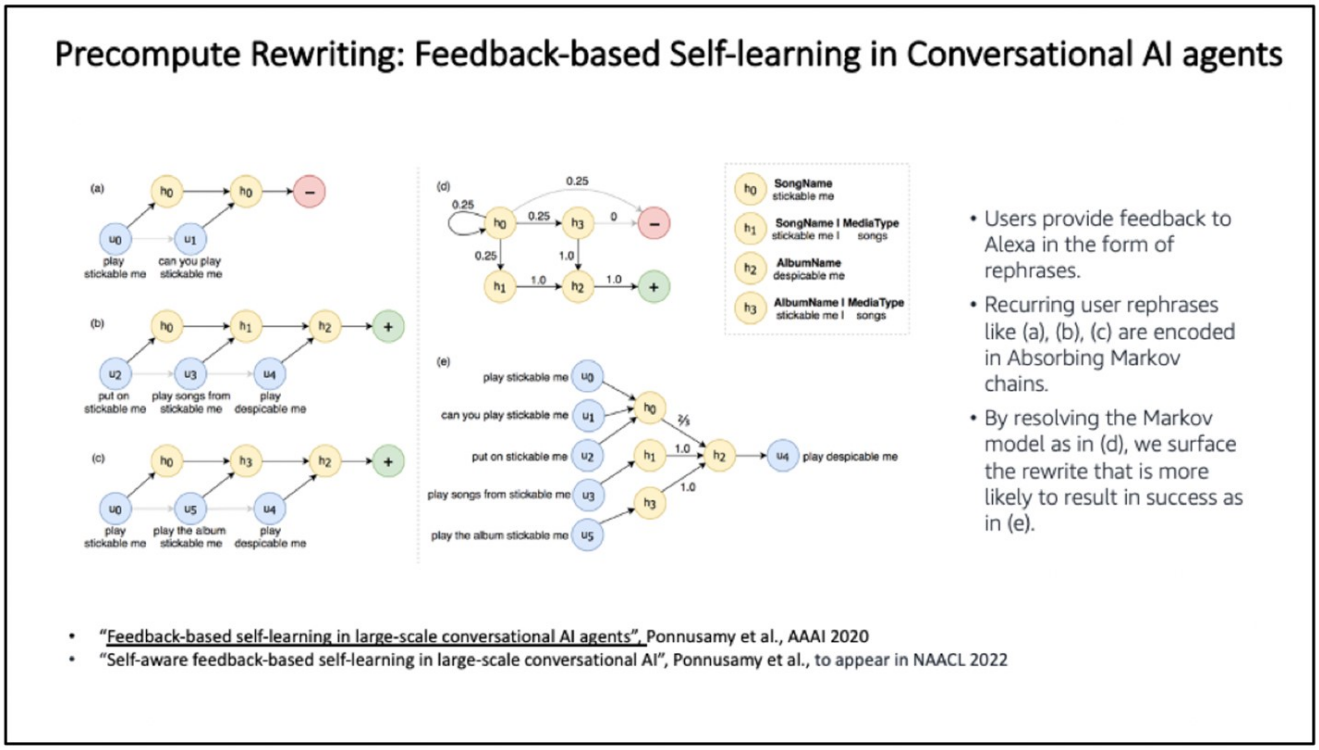

Figure 3: The probabilities of sequences of rephrases across customer interactions can be encoded in absorbing Markov chains.

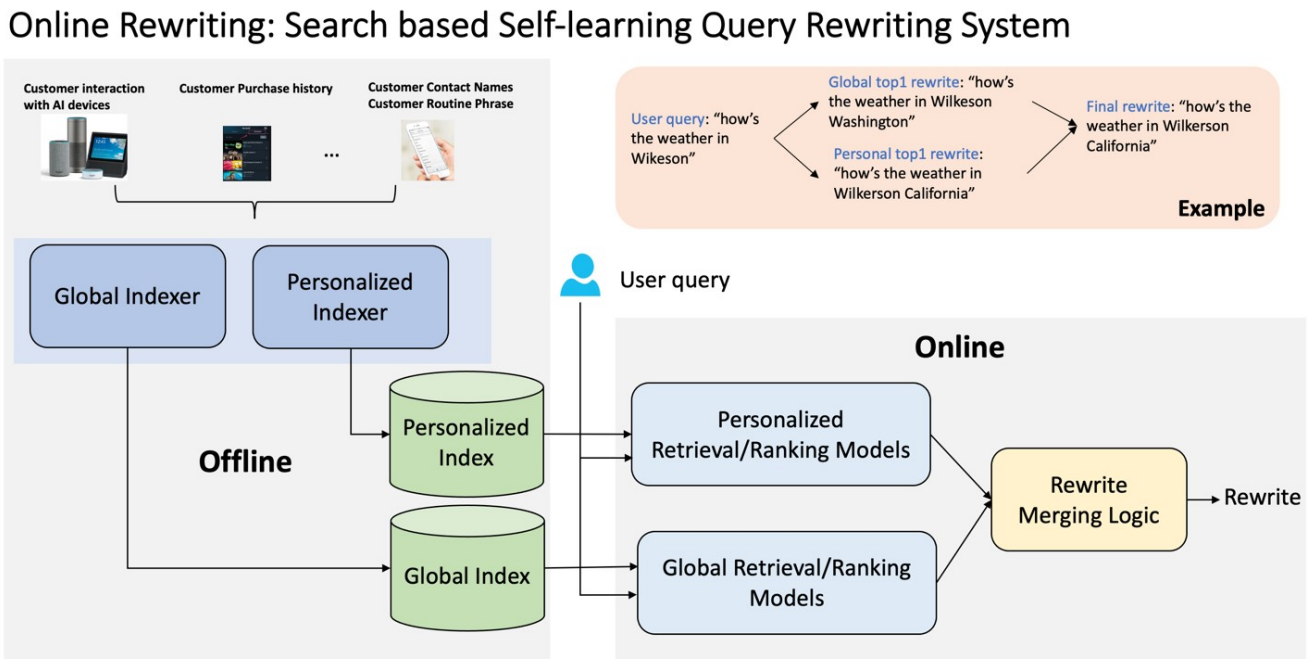

Figure 4: The online rewriting model's personal layer factors in customer context, while the global prevents overindexing on personalized cases.

has a final state, with zero probability of transitioning to any other, which is accessible from any other system state.

We use absorbing Markov chains to encode the probabilities that any given rephrase of the same query will follow any other across a range of interactions. Solving the Markov chain gives us the rewrite for any given request that is most likely to be successful.

2.2.2 Online rewriting

Instead of relying on customers' own rephrasings, the online rewriting mechanism uses retrieval and ranking models to generate rewrites [Cho et al., 2021]. We generate rewrites based on customers' habitual usage patterns with the agent. In the example shown in Fig.4, for instance, based on the customer's interaction history, we rewrite the query "What's the weather in Wilkerson?" as "What's the weather in Wilkerson, California?" — even though "What's the weather in Wilkerson, Washington?" is the more common query across interactions.

The model does, however, include a global layer as well as a personal layer, to prevent overindexing on personalized cases (for instance, inferring that a customer who likes the Selena Gomez song "We Don't Talk Anymore" will also like the song from Encanto "We Don't about Bruno") and to enable the model to provide rewrites when the customer's interaction history provides little or no guidance.

Both the personalized workstream and the global workstream include both retrieval and ranking models:

- The retrieval model uses a dense-passage-retrieval (DPR) model, which maps texts into a low-dimensional, continuous space, to extract embeddings for both the index and the query. Then it uses some similarity measurement to decide the rewrite score.
- The ranking model combines fuzzy match (e.g., through a single-encoder structure) with various metadata to make a reranking decision

We have deployed all three of these self-learning approaches; BERT-based, Markov-chain-based offline rewriting and online rewriting. All have made a significant difference in the quality of Alexa customers' experience. In experiments, we compared the BERT-based offline approach to four baseline models on six machine-annotated and two human-annotated datasets, and it outperformed all baselines across the board, with improvements of as much as 16% to 17% on some of the machine-annotated datasets, while almost doubling the improvement on the human-annotated ones.

The offline approach that uses absorbing Markov chains has rewritten tens of millions of outputs from Alexa's Automatic Speech Recognition (ASR) models, and it has a win-loss ratio of 3:1, meaning that for every one incorrect rewrite, it has 3 correct ones.

And finally, in a series of A/B tests of the online rewrite engine, we found that the global rewrite alone reduced the defect rate by 13%, while the addition of the personal rewrite model reduced defects by a further 4%.

2.2.3 Teachable AI

Query rewrites depend on implicit signals from customers, but customers can also explicitly teach Alexa their personal preferences, such as "I'm a Warriors fan" or "I like Italian restaurants."

Alexa's teachable-AI mechanism can be either customer-initiated or Alexa-initiated [Ping et al., 2020]. Alexa proactively senses teachable moments, when, for instance, a customer repeats the same request multiple times or declares Alexa's response unsatisfactory. And a customer can initiate a guided Q&A with Alexa with a simple cue like, "Alexa, learn my preferences". In either case, Alexa can use the customer's preferences to guide the very next customer interaction.

2.2.4 Failure point isolation

Besides recovering from defects through query rewriting, we also want to understand the root cause of failures for defects. Conversational AI systems like Alexa depend on multiple models that process customer requests in stages. First, a voice trigger (or "wake word") model determines whether the user is speaking to the assistant. Then an ASR module converts the audio stream into text. This text passes to a Natural Language Understanding (NLU) component that determines the intent of user request. An entity recognition system recognizes and resolves entities, and a routing system selects the best skill (or application) that should serve the request [Kim et al., 2018]. The Natural Language Generation (NLG) system generates the best possible response. Finally, the text-to-speech (TTS) system renders the response into human-like speech.

For Alexa, part of self-learning is automatically determining, when a failure occurs, which component has failed. An error in an upstream component can propagate through the system, in which case multiple components may fail. Thus, we focus on the first component that fails in a way that is irrecoverable, which we call the "failure point".

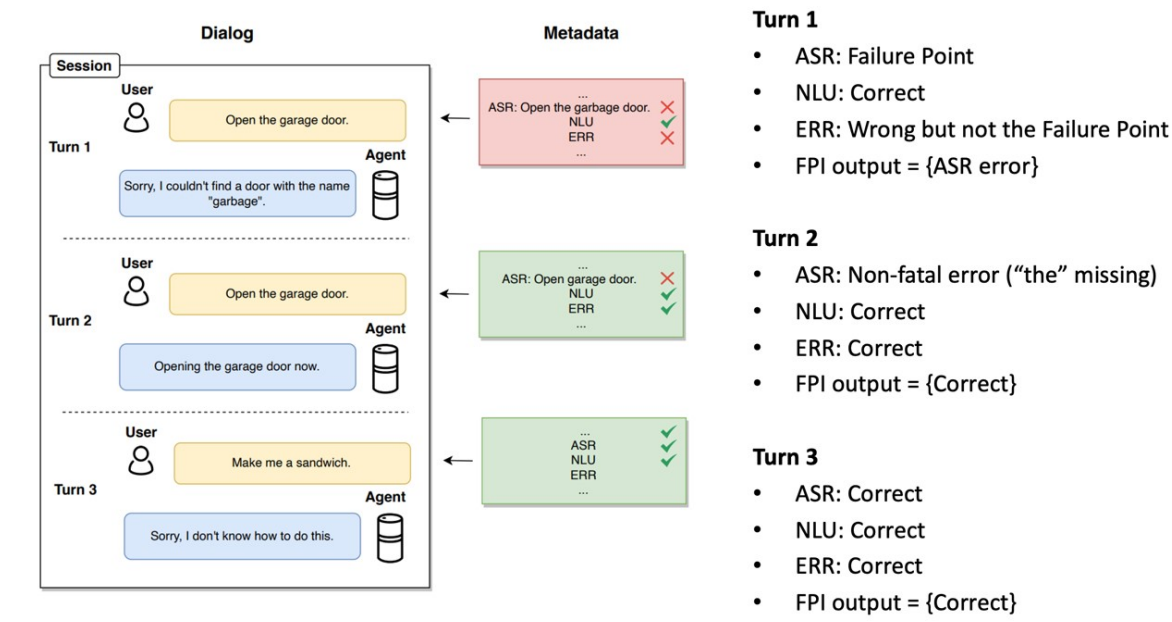

Figure 5: Failure point isolation identifies the earliest point in the processing pipeline at which a failure occurs, and errors that the conversational agent recovers from are not classified as failures.

In our initial work on failure point isolation, we recognize five error points as well as a "correct" class (meaning no component failed). The possible failure points are false wake (errors in voice trigger); ASR errors; NLU errors (for example, incorrectly routing "play Harry Potter" to video instead of audiobook); entity resolution and recognition errors; and result errors (for example, playing the wrong Harry Potter movie).

To better illustrate failure point problem, let's examine a multiturn dialogue given in Fig. 5. In the first turn, the customer is trying to open a garage door, and the conversational assistant recognizes the speech incorrectly. The entity resolution model doesn't recover from this error and also fails. Finally, the dialogue assistant fails to perform the correct action. In this turn, ASR is the failure point, despite the other models' subsequent failure.

On the second turn, the customer repeats the request. ASR makes a small error by not recognizing article "the" in the speech, but the dialog assistant takes the correct action. We would mark this turn as correct, as the ASR error didn't lead to the system failure.

The last turn highlights one of the limitations of our method. The user is asking the dialogue assistant to make a sandwich, which dialogue assistants cannot do yet. All systems have worked correctly, but the user is not satisfied. In our work, we do not consider such turns as defective.

On average, our best failure point isolation model achieves close to human performance on across different categories (92% vs human). This model uses extended dialogue context, features derived from logs of the assistants (e.g., ASR confidence), and traces of decision-making compo-

nents (e.g., NLU modules). We outperform humans in result and correct class detection. ASR, entity resolution, and NLU are in the 90-95% range [Khaziev et al., 2022].

The day when computing fades into the environment, and we walk from room to room casually instructing embedded computing devices how we want them to behave, may still lie in the future. But at Alexa AI, we're already a long way down that path. And we're moving farther forward every day.

3 Conclusions

Increasingly, Conversational AI systems will be making decisions under a richer and more complex ambient and personal context. The ground truth for these decisions will be context and person specific and thus hard to simulate in offline setting to train ML models through a supervised learning paradigm. One source of supervision that will scale with this complexity is the users themselves. Devising methods that learn from user interactions, their implicit and explicit feedback within their ambient state is a promising solution. We call this method self-learning and demonstrated its effectiveness on several tasks for Alexa, Amazon's conversational AI system.

Acknowledgement

The content of this work is combination of multiple papers authored by numerous Alexa AI scientists.

References

Eunah Cho, Ziyan Jiang, Jie Hao, Zheng Chen, Saurabh Gupta, Xing Fan, and Chenlei Guo. Personalized search-based query rewrite system for conversational ai. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 179–188, 2021.

Saurabh Gupta, Xing Fan, Derek Liu, Benjamin Yao, Yuan Ling, Kun Zhou, Tuan-Hung Pham, and Chenlei Guo. Robertaiq: An efficient framework for automatic interaction quality estimation of dialogue systems. In *2nd International Workshop on Data-Efficient Machine Learning (DeMaL)*, 2021.

Rinat Khaziev, Usman Shahid, Tobias Röding, RAKESH CHADA, Emir Kapanci, and Pradeep Natarajan. Fpi: Failure point isolation in large-scale conversational assistants. In *NAACL 2022*, 2022. URL https://www.amazon.science/publications/fpi-failure-point-isolation-in-large-scale-conversational-assistants.

Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya. Efficient large-scale neural domain classification with personalized attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2214–2224, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1206. URL https://aclanthology.org/P18-1206.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Qing Ping, Feiyang Niu, Govind Thattai, Joel Chengottusseriyil, Qiaozi Gao, Aishwarya Reganti, Prashanth Rajagopal, Gokhan Tur, Dilek Hakkani-Tur, and Prem Natarajan. Interactive teaching for conversational ai. In *NeurIPS 2020 Workshop on Human in the Loop Dialogue Systems*, 2020. URL https://www.amazon.science/publications/interactive-teaching-for-conversational-ai.

Pragaash Ponnusamy, Alireza Roshan-Ghias, Chenlei Guo, and Ruhi Sarikaya. Feedback-based self-learning in large-scale conversational ai agents. In *The Thirty-Second Annual Conference on Innovative Applications of Artificial Intelligence*, 2020.

Ruhi Sarikaya. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34(1):67–81, 2017.

Zhuoyi Wang, Saurabh Gupta, Jie Hao, Xing Fan, Dingcheng Li, Alexander Hanbo Li, and Edward Guo. Contextual rephrase detection for reducing friction in dialogue system. In *EMNLP 2021*, 2021. URL https://www.amazon.science/publications/contextual-rephrase-detection-for-reducing-friction-in-dialogue-system.