



On the care and feeding of virtual assistants: Automating conversation review with AI

Ian Beaver¹ | Abdullah Mueen²

¹ Verint Systems Inc., 175 Broadhollow Rd, Ste 100, Melville, NY 11747, USA

² University of New Mexico, 210 University Blvd NE, Albuquerque, NM 87106, USA

Correspondence

Ian Beaver, Verint Systems Inc.
Email: ian.beaver@verint.com

Abstract

With the rise of intelligent virtual assistants (IVAs), there is a necessary rise in human effort to identify conversations containing misunderstood user inputs. These conversations uncover error in natural language understanding and help prioritize improvements to the IVA. As human analysis is time consuming and expensive, prioritizing the conversations where misunderstanding has likely occurred reduces costs and speeds IVA improvement. In addition, less conversations reviewed by humans mean less user data are exposed, increasing privacy. We describe Trace AI, a scalable system for automated conversation review based on the detection of conversational features that can identify potential miscommunications. Trace AI provides IVA designers with suggested actions to correct understanding errors, prioritizes areas of language model repair, and can automate the review of conversations. We discuss the system design and report its performance at identifying errors in IVA understanding compared to that of human reviewers. Trace AI has been commercially deployed for over 4 years and is responsible for significant savings in human annotation costs as well as accelerating the refinement cycle of deployed enterprise IVAs.

BACKGROUND

Intelligent virtual assistants (IVAs) such as Amazon's Alexa or Apple's Siri along with specialized company-specific agents for customer service and sales support are exploding in popularity (Ram et al. 2018). The continued adoption of IVAs is contributing to a growing problem. How do we refine an IVA's knowledge effectively and efficiently? As IVA use as well as the number of tasks an IVA is expected to perform increases, there is a corresponding jump in the number of human-computer interactions to be reviewed for quality assurance. Therefore, discovering a means to expedite review and analysis of these interactions is critical.

Without scalable and efficient methods of automated conversation review, IVA designers must rely solely on human reviewers to validate expected behavior of the IVAs.

As this is a manual and time-consuming process, the reviewers are only able to view a limited number of interactions. The result is also subjective since reviewers may disagree on the user intention for any given turn in a conversation. In addition, as the IVA improves, errors in communication appear less often in a random sample due to their dwindling numbers. A recent challenge is public outcry over the human review of IVA conversation logs for the purpose of language understanding verification, due to privacy concerns. By the use of an automated system for conversation review, problematic interactions can still be surfaced without exposing the entire set of logs to human reviewers, minimizing privacy invasion.

In this article, we discuss a scalable system to process all conversations and autonomously mark the interactions where the IVA is likely misunderstanding the user. This system has been commercialized in Trace AI, a Verint



Systems product. Trace provides cost savings to companies deploying IVAs by reducing the time human reviewers spend looking at conversations with no misunderstandings present. It also enables a shorter IVA refinement cycle as problems are surfaced quickly and more reliably than a random sample or confidence metric based review. Trace has been deployed in the refinement of enterprise IVAs since 2016. Initially created as an application internal to Verint Systems, recently we have made the system available to external partners due to its success at reducing annotation costs.

Common to all IVAs is a natural language understanding (NLU) component (Ram et al. 2018). The NLU maps user inputs, or conversational *turns*, to a derived semantic representation commonly known as the *intent*, an interpretation of a statement or question that allows one to formulate the best response. The collection of syntax, semantics, and grammar rules that defines how input language maps to an intent within the NLU is referred to by us as a *language model*. The language model may be trained through machine learning methods or manually constructed by human experts (Zhao and Eskenazi 2016). Manually constructed symbolic models require humans to observe and formalize these language rules while machine-learned models use algorithms to observe and approximate them.

Regardless of implementation details, to improve the language models and for quality assurance, human-IVA interactions need to be continuously reviewed. Improvements include the addition of vocabulary and new rules or the revision of existing rules that led to incorrect mappings within the language model. For machine-learned models, identification of incorrect understanding can highlight confusion within the model and prioritize areas of further training. The main focus of misunderstanding detection is on intent classification. It is in the NLU component that the breakdown of communication will begin, assuming adequate automatic speech recognition, if speech is used as an interface. The detection of speech recognition error and recovery is well covered in literature (Ogawa and Hori 2015; Kim, Ryu, and Lee 2016) and outside the scope of Trace.

Even if intent classification errors can be recovered from and the conversation completes successfully, discovering the cause of the misunderstanding can lead to improvements in the IVA, and therefore user experience, and more intelligent error resolution strategies (Bohus and Rudnicky 2005). For example, in the following conversation, several communication errors on the part of the IVA are apparent.

User: I cannot set up my mothers Email from my home computer. I am an authorized user for her account.
Agent: Would you like to create a new Email address or set up your existing Email address with an Email program such as Outlook, Apple Mail, etc...?

User: yes
Agent: What else can I help you with?
User: Cannot set up Email
Agent: I have found more than one answer to your question. Please select the link below that interests you.
User: Set up Email
Agent: Would you like to create a new Email address or set up your existing Email address with an Email program such as Outlook, Apple Mail, etc...?
User: Set up Email with an Email program
Agent: To set up your Email with an Email program select your Email program below.
User: Outlook and Outlook Express
Agent: To send and receive Email with your new address you will need to enter some basic information into your Email program. On the page I have opened for you click the link for the Outlook program you are using for specific setup instructions.

In this conversation, the IVA initially cannot tell if user is trying to set up and existing Email or create a new one. When the user replies with “yes,” the IVA does not understand what the user is referring to and replies with a generic offer to be of further assistance. This leads the user to repeat their original query but in a much clarified form, lacking the backstory and justification present in the original query. The IVA is still unsure of the best response so it asks the user to clarify the task, which leads it back to its original response. At this point the user very clearly repeats their original query but with the specifics the IVA needs to understand the task. From there on the conversation progresses again with the IVA walking the user through the steps needed to set up and existing Email account using the Outlook mail application. While this leads to a successful resolution of the task, it is clearly not an efficient resolution and most users would not be satisfied with the experience. Therefore, review is needed even in conversations where the attempted tasks were successfully completed.

Existing IVA refinement processes

IVAs for customer service are deployed in a specific language domain such as transportation, insurance, product support, or finance. Periodically, human reviewers are given a sample of recent conversations collected from a live IVA for quality assurance. The reviewers need to be familiar with any domain specific terminology. This poses difficulty in the utilization of crowd-sourced platforms such as Appen¹ or Mechanical Turk² as workers must be vetted to ensure they have proper knowledge of the domain and associated terminology. One strategy is to create a series

of tests that workers must pass before accessing the task. Another strategy injects conversations with known labels to the sample and removes reviewers that score poorly on them. The sample to be reviewed can be selected in a variety of ways. If a particular event is important to analyze, such a user requesting an escalation to a human, all conversations containing the event are selected. Such samples are biased and may miss many other important failure scenarios, so for a more holistic view a random sample can be taken. Another strategy selects interactions where the NLU confidence score is lower than some predetermined threshold. In this case, reviewers rely on the system itself to indicate where error lies. While low confidence is potentially more effective than a random sample at finding poor interactions, this requires trusting the very system that is being evaluated for errors. This also creates a dependency on the underlying system implementation that makes it difficult to compare the performance of different IVAs, or, if the system design is ever modified, the same IVA over time.

Sampled conversations are manually graded in an effort to find intents which need improvement. The reviewers may use various grading means, but a star rating system such as one-to-five stars is common (Kuligowska 2015). The result of this review process is a set of conversations along with their grades which are passed to *domain experts*. Domain experts are typically trained in natural language processing and/or machine learning and are responsible for the construction and modification of language models. Only poorly graded conversations require in-depth analysis by domain experts to determine the necessary changes to a given language model and implement them to produce an updated model, which is then deployed to the live IVA. The faster this review-correction cycle completes, the more quickly the IVA can adapt to changes in products or domain language or learn additional topics that users are attempting to converse about.

SYSTEM DESIGN AND COMPONENTS

The core of our system is a learned model of features for predicting intent classification errors in conversational turns. This model is used to generate a score per turn representing the risk of intent misclassification. This risk score is used to rank turns for priority review where humans vote on if each turn was misunderstood. The system will also vote on its own if the turn was misunderstood to reduce or eliminate the need for human voting. After a training period, the system voting performance can be comparable to that of humans and instead of a just a sample of the logs, produces voting outcomes for all conversations. The human and system voting outcomes are then used to gen-

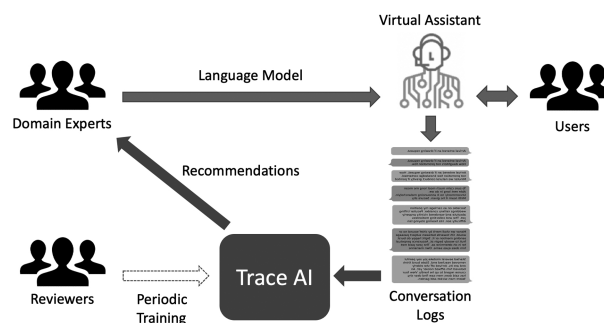


FIGURE 1 The intelligent virtual assistant (IVA) refinement process incorporating Trace AI

erate suggested actions to fix the human-identified errors in the language model. Figure 1 shows how Trace fits into the existing refinement cycle.

Trace is designed with three primary functions. The first is detecting numerous features of intent error and aggregating these features into a risk score. The risk analysis engine applies various heuristics and classifiers to detect indications of misunderstanding in each conversational turn and score them between [0,1]. As each indication is independently scored, and each conversation is independent, this task is done in parallel on a compute cluster for scalability. This modular approach also allows for easy experimentation with new detection mechanisms and automatic enabling/disabling of individual detectors based on IVA-specific properties.

Once each turn is annotated with all applicable risk indicators, the risk score for a particular turn is calculated as the weighted sum of all indicator scores in that turn. Weights are initialized to 0.5 and tuned over time using odds ratios. The odds ratio represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure (Szumilas 2010). As reviewers grade turn-intent mappings, the odds of each risk indicator predicting a misunderstanding is recalculated and the weight of that risk indicator is adjusted, improving the scoring model. As indicators may be domain-dependent, weights are adjusted per domain.

The other two functions of Trace are provided by a Django web application with interfaces designed for two types of users. The first type are the domain experts who create projects, linked to a live IVA, and select a time range over which to do analysis. Once they have defined a project and a time range for review, Trace prioritizes all conversational turns within that range by their risk score. The second type are the human reviewers whose work flow involves logging into a project (a collection of conversations from a live IVA) and reviewing turns that have been prioritized by risk score. They read each turn

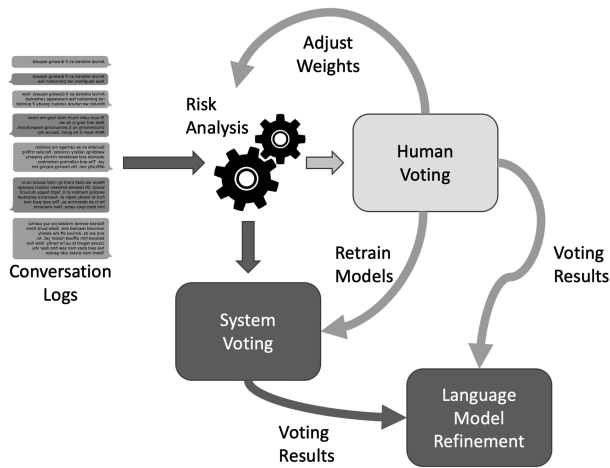


FIGURE 2 The flow of data and annotations through Trace

in the context of the conversation and vote on whether or not they agree with the intent chosen by the NLU in the live IVA. If a reviewer does not feel they have enough information or domain knowledge to decide, they may also vote *Unsure*. Each turn is shown to a minimum of three reviewers to reduce subjectivity and provide domain experts with a sense of inter-reviewer agreement.

As shown in Figure 2, these reviewer decisions are applied as training data to both tune the individual risk indicator weights and to retrain the system voting component, and are displayed to the domain experts to evaluate the current performance of Trace against humans on the IVA under review. Once the domain experts are satisfied with the voting performance of Trace, they can choose to cease the human voting until further training is desired.

The reviewer interface

A screen shot of the voting interface is shown in Figure 3. In the left-hand column, the intent the reviewer is currently voting on is displayed along with additional information about the current intent to give insight. The label of the intent is displayed at the top followed by a text description of its purpose which is maintained by the domain experts. If the reviewers do not fully understand the purpose of an intent, they can submit questions to the domain experts by clicking on the comment bubble below the description text. The experts can then update the description to clarify the purpose of the intent so that voting is accurate.

Next, a set of sample of conversational turns that have been previously human-validated to belong to this intent are displayed. This is to give the reviewer some intuition on the language intended for the current intent. Following that is a list of related intents to help the reviewer decide

if a more suitable intent currently exists in the language model. Both lists are searchable to speed analysis. Finally, controls to navigate through the intents to be reviewed and, at the bottom, metrics on how many turns have been completed by the reviewer and all reviewers combined on the currently displayed intent are shown.

On the right-hand side, the user turn from a conversation with an IVA is shown followed by voting buttons. Keyboard shortcuts are provided to speed voting. The entire conversation with the current turn highlighted is displayed to give the reviewer the conversational context needed to determine if the responding intent was appropriate. Notice that nowhere does the actual response *text* from the IVA appear. The response is not shown in order to separate the evaluation of the NLU component from that of the natural language generation (NLG) component.

In an IVA, the NLG is responsible for taking the action determined by the dialog management component, which is derived by considering the detected intent as well as other environmental variables such as the current state of the dialog, and formulating a natural language response to return to the user. Poor response wording can appear to the user as a misunderstanding when, in reality, the NLU component understood the intent but the generated response was inaccurate. In addition, in large IVAs it is possible that the NLU and NLG are maintained by different people or altogether different departments or roles. With commercial IVAs, the response text may have to go through an approval process or even a legal review to ensure that the information the IVA is giving the user is accurate and legal. Therefore, it is critical that errors in the NLU are separated from errors in the NLG so that the appropriate corrections can be identified. Once it has been established that the NLU is performing acceptably the NLG can be evaluated separately, which is outside the scope of Trace.

The analysis interface

After the risk analysis and voting processes are complete, Trace provides voting data and additional recommendations to the domain experts to facilitate language model development. To optimize domain experts' time, Trace uses the voting outcomes to determine a recommended action per turn as shown in Table 1 and visualized in Figure 4. These recommendations help the domain experts quickly determine what to do with the voting results for a particular turn and allow them to group the results by a similar action, such as display all turns where there was no reviewer consensus to discover if any intent descriptions need clarification.

To prioritize language model repair work by the impact it will have on the live IVA, the set of turns and their

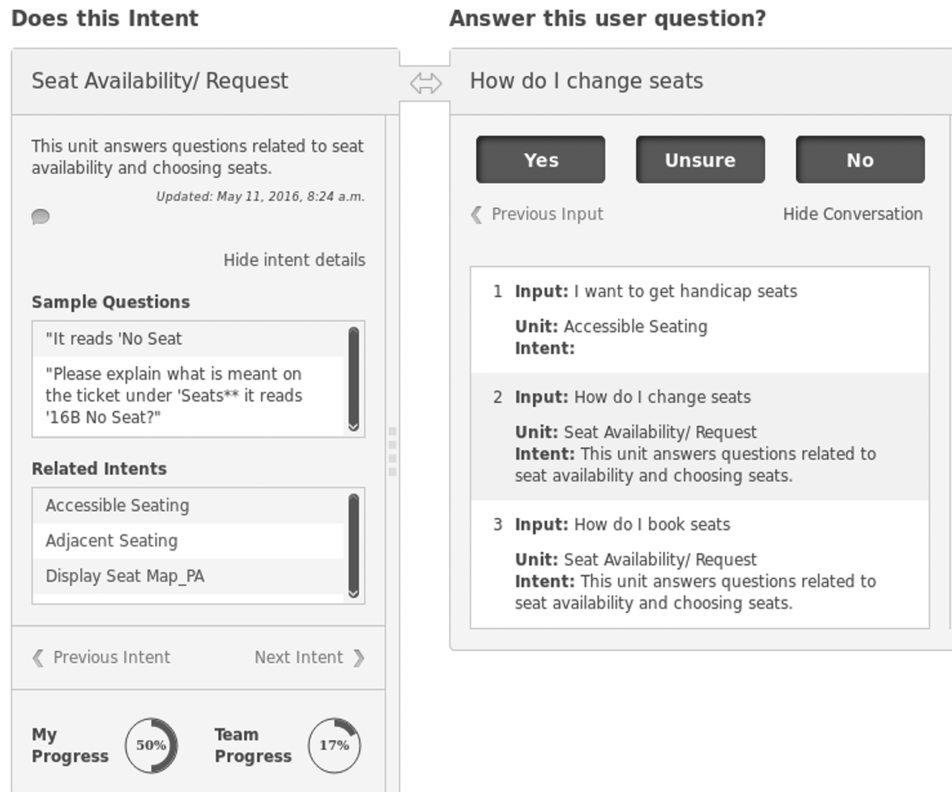


FIGURE 3 The Trace voting interface used by reviewers

Voting Results

Filter Results

Units▼	Input Types▼	Exported▼	Export Date▼	Action Required▼	Voter▼	<div><input type="checkbox"/> Yes <input type="checkbox"/> Unsure <input type="checkbox"/> No</div>	Re-release Inputs
Input	Unit Hit▲	Input Type▲	Voting Results			Action▲	
book a cheap flight	Cost of tickets	Current	1	0	2	Analyze: Wrong Intent	
what are the restrictions for a lowest available fare?	Cost of tickets	Current	0	1	2	Analyze: Wrong Intent	
What prevents the Oil companies from selling you gas at reduced prices instead of gorging themselves by selling it at high rates to China?	Cost of tickets	Current	0	0	3	Analyze: Wrong Intent	
dates for low fares	Cost of tickets	Current	1	1	1	Analyze: No Consensus	
I can find a cheaper airfare on another website, can you match it?	Cost of tickets	Current	2	0	1	Analyze: Conflicting Outcomes	
show me the prices	Cost of tickets	Current	3	0	0	Add as Test Question	

FIGURE 4 Analysis interface within Trace used by domain experts to view voting results and reviewer agreement

voting outcomes are first grouped by responding intent and then ordered by the frequency of response within the conversation logs. A screen shot of this prioritization from the analysis interface is shown in Figure 5. Each bar is the total volume of user turns that was matched a specific intent in the conversation logs by the live IVA. The red portion is the count of misunderstood inputs, as determined by

an adjustable threshold on the risk score, assigned to that intent. The gray portion is the count of correct inputs.

By looking at this chart, domain experts can quickly determine which malfunctioning intents have a greater impact on user experience with the IVA. If two intents have a similar ratio of misunderstood inputs, the intent with the higher response frequency would be prioritized for repair

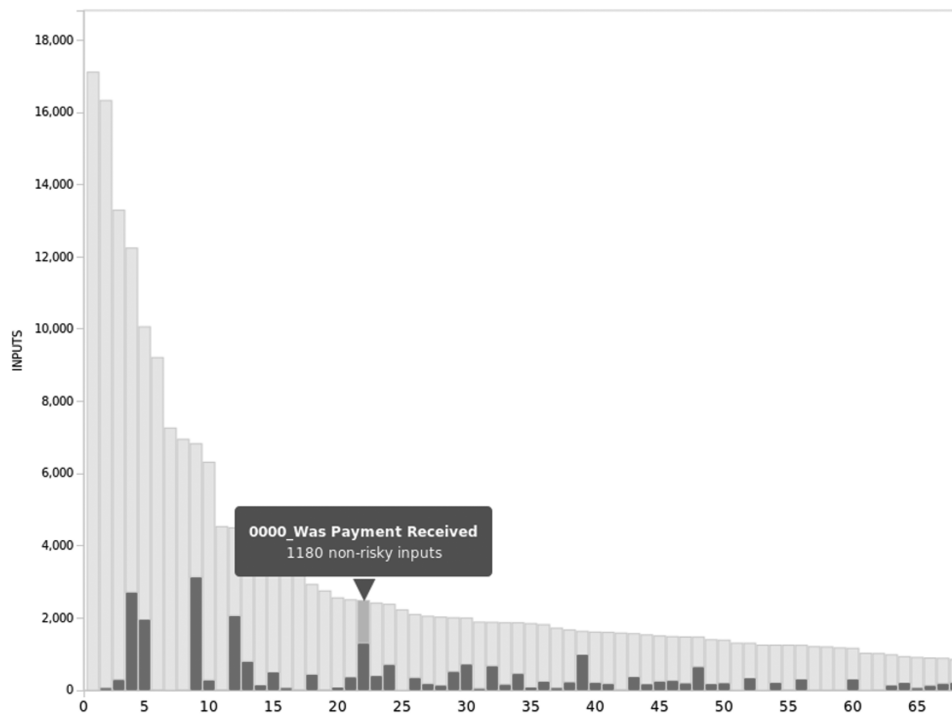


FIGURE 5 The ratio of misunderstood to correct inputs

TABLE 1 Voting outcomes and recommended actions

Circumstance	Recommended action
A turn-to-intent map is voted to be correct	None. These are added as training and regression samples for Trace.
A turn-to-intent map is voted to be incorrect	Fix or retrain the language model to prevent the turn from reaching the associated intent.
The reviewer majority votes <i>Not Sure</i>	Determine if the intent was appropriate for the turn or if a new intent should be created.
There is no reviewer consensus	Determine if the intent was appropriate for the turn or if the intent description needs clarification.
Voters are conflicted as they approved the turn in more than one intent	Clarify definitions of both intents and re-release for voting.

as its malfunction will have a larger impact on overall user experience.

To help domain experts quickly analyze the voting results and voter consensus, the analysis interface also provides the tabular view as shown in Figure 4. The filters at the top provide the ability to explore the results from many angles such as per intent, per voter, date range, recommended action, etc. In the left-hand column, the original user turn text is displayed. In the next column is the

intent that the reviewers evaluated the text against. The “Input Type” column shows whether the intent evaluated was from the current NLU or a different source, such as regression tests used in developing the language model or annotated human–human live chat logs. Trace is designed in such a way that it can perform misunderstanding analysis on any textual data labeled with intent or topic, as the risk analysis engine is not dependent on any implementation details on how the intent or topic was derived. The “Voting Results” column provides a visual indicator of the voting outcome and inter-reviewer agreement. The final column on the right-hand side is the recommended action from Table 1. Filtering this table by an action type will quickly surface all turns where a particular action should be performed.

From this view, the domain experts can quickly find areas of the language model that need attention and export the text data with any detected risk indicators and voting results. They can then use this data along with the NLU-specific means to make the necessary changes in the language model.

INDICATORS OF INTENT MISCLASSIFICATION

This section describes the current individual indicators of intent classification error that the risk analysis engine tests

for. These indicators were derived from literature review on miscommunication in conversation between humans combined with our own empirical evidence and tribal knowledge gathered over 18 years of enterprise IVA development. Some of these indicators operate at the conversation level while others at the turn level as there exists both global and local features of miscommunication in a natural language conversation (Higashinaka et al. 2015).

Conversation level features

The following features apply risk equally across all turns within the single conversation where they are present. These features are used to detect miscommunication over the course of the conversation and elevate the risk score for turns in conversations where miscommunication was likely to have occurred.

I do not know (IDK) occurred in conversation

An IDK occurs when the language model does not find an intent that satisfies the user query with a high enough confidence. The IVA may respond with something like “I’m sorry, I didn’t understand you. Please reword your question.” If a conversation contains one or more IDK responses, this may indicate that the user is talking about some subject the IVA has no knowledge of. Either the correct intent does not exist in the IVA’s knowledge base or it does exist and the mapping to that intent is faulty. However, with the former, sometimes an IDK is appropriate. For example, the user could ask something off topic such as what the IVA’s favorite sports team is. Utterances triggering acceptable IDK responses are classified as out-of-domain and out-of-application-scope utterances in (Bohus and Rudnicky 2005). Therefore, not every IDK response is a missed intent. IDKs must be weighed and combined with other indications of risk into an overall risk score for the turns they occur in.

Same intent(s) hit

The same intent is hit more than once within the conversation (not necessarily successively). This is an indication of risk within a customer service conversation because it is unlikely the user would want to see the same response repeated. If these are successive in a conversation they are considered to be sequential hits. This usually indicates that the response to the first input did not satisfy the user; he or she is rewording the question to get a different response. If the system has the initiative, this may mean that the system

is repeating a prompt, a common indication of miscommunication (Aberdeen and Ferro 2003).

Intent tie occurred in conversation

The responding intent for one or more turns in the conversation had a nearly identical score as one or more different intents. This indicates confusion in the NLU around the input language for the tying intents. If a conversation contains such ties, it may surface subject matter that is not well defined in the language model.

User rating scores

Users may be asked for feedback on how helpful the IVA was for their session. However, feedback is not reliable in customer service as we have observed users who give negative feedback if the IVA rightly upholds business rules. For example, business rules may prevent transferring a ticket to a different passenger, and, when a user attempts to do so, the IVA will not let them. In retribution, the user grades the conversation poorly but the IVA was not at fault. The user may also say the IVA was unhelpful when the NLU was indeed working correctly, but the response text was poorly worded. Therefore, this feedback is only an indication of risk in our system and not the final determination of understanding error as was used in (Jiang et al. 2015).

Conversation should have escalated

An escalation occurs when a user requests an alternative channel than the IVA for the completion of a task. Sometimes escalations are due to communication failures, but they can also be triggered by business rules for complex tasks or identity verification. Whether or not there was explicit user request for an escalation in the conversation, the algorithm described in (Freeman and Beaver 2017) has determined that the conversation should have been escalated due to IVA failures to complete the task at hand.

Sentiment change over time

The QA^{RT} system presented in (Roy et al. 2016) monitors live customer service dialogs and provides supervisors with visualizations and summaries of ongoing chats. It employed features in the categories of customer behavior (emotion and sentiment) as well as compliance (greeted customer, used customer name, assurance, etc.). As the QA^{RT} system is monitoring human-human chats



there is no concept of intents nor does it directly detect misunderstanding. However, change in sentiment and emotion proved useful for indicating a misunderstanding occurred between human speakers and was therefore implemented in Trace. This indicates the user began the conversation with positive or neutral sentiment, but by the end of the conversation their sentiment was negative. This may be caused by either the IVA preventing them from completing a task due to business rules or due to IVA misunderstanding.

Turn level features

The following features only apply risk to a single turn. However, they may still use features of the conversational context in their determination.

Triggers IDK response

If the response to this turn is an IDK, this may indicate that the user has asked about a subject the IVA does not have knowledge of.

Input contains backstory

Users may give backstory on their task that is unnecessary for determining the correct intent, such as in the example conversation shown previously in the background section. The presence of this language can add confusion in the NLU and result in an intent error (Beaver, Freeman, and Mueen 2020). For example, a user may tell the IVA that he or she needs to fly to Boston for a son's graduation party. The fact that the user has a son and is attending his graduation party is irrelevant to the task. The additional language can interfere with determining the user's primary task of booking a flight. We apply (Kim, Ryu, and Lee 2016) to segment intents in the text, and if the NLU is unable to determine the intent of a segment, we consider it the presence of out-of-domain/unnecessary language.

Precedes corrections

The following user turn contains error correction language such as "no, ..," "I said ..," ".. not what I ..". This is a common signal or error corrections humans employ in conversation (Bulyko et al. 2005; Freeman and Beaver 2017).

Abandonment

The user left the conversation immediately after the IVA asked them a question. This indicates that the IVA did not have all the information it needed to complete the task, but the user abandonment indicates it was likely trying to accomplish the *wrong* task and the user left in frustration.

Input contains multiple intents

If multiple intents are present it can add confusion to the NLU. We assume the IVA under review does not support multiple intents within a single turn as multi-intent parsing is still an unsolved problem for IVAs (Khatri et al. 2018). Using the method given in (Kim, Ryu, and Lee 2016), we detect if multiple intents are present in the user turn.

Triggers sequential hit or impasse

The turn hits the same intent as the previous turn. This usually indicates that the previous response did not satisfy the user, so he or she is rewording the question to get a different response but failed to do so. An *impasse* occurs when the same intent is returned more than two times in a row. In which case the IVA may respond with something like "I think you are asking for more information than I have."

Precedes escalation

As escalations may be due to previous IVA failures, risk is assigned to the turn preceding any escalation request.

Precedes unhelpful

The input directly preceded a turn stating the unhelpfulness of the IVA. This is a common reaction when the user is frustrated at the inability to make progress in their task.

Precedes profanity

The input directly preceded an interaction containing profanity. With a customer service or product support IVA, profanity is usually a sign of user frustration or irritation.

Precedes negative sentiment

If a turn contains negative sentiment, this may be due to the user's reaction to the previous IVA response. Therefore, risk is assigned to the preceding user turn.

Restated

If a turn is very similar to one or more following turns, this may indicate the user was dissatisfied with the response and rewords the question. Similarity is defined as a rephrasing of the same question or statement as measured by cosine similarity of sentence vectors; it may not have triggered the same intent in the IVA (Jiang et al. 2015).

Precedes IDK

We have observed that IDKs may follow misunderstood turns. This type of IDK can happen when the user reacts in surprise or frustration (“*What??*”) or changes the subject to complaining about the IVA due to the misunderstanding (“*This is the dumbest thing I have ever used!*”).

Triggers tie

The responding intent had a nearly identical classification score as one or more different intents. This indicates confusion in the language model around the input language.

Contains unknown words

The user turn contains words that are out of vocabulary for the underlying language model. This may indicate that the user is talking about some subject the IVA does not yet have knowledge of.

Should escalate point

There was no explicit user request for escalation in the conversation, but an algorithm (Freeman and Beaver 2017) determined that the conversation *should* have escalated at this point in the conversation due to task failures.

EVALUATION

The purpose of Trace is to reduce the human burden and costs in maintaining conversational agents. To demonstrate its utility, we measure its performance in automating

TABLE 2 Dataset statistics for the evaluation data

Dataset	# Conv	Total user turns	Textual user turns	Majority agreement
Train	2030	13,930	7270	6331
Telecom	1342	20,485	7313	5252
Airline	1611	9103	9103	6978
Average	1661	14,506	7895	6187

the reviewer voting process on real datasets from three live IVAs as well as performing a cost analysis of human review.

Data

Due to annotation budget for this study, we limited our average user turns per dataset to 8000. All turns in a conversation need to be reviewed. However, conversations have varying numbers of turns and, with multi-modal IVAs, not all user turns consist of natural language (Hewitt and Beaver 2020). For example, some user turns in a conversation may be events such as user interface clicks or web page navigations which the IVA responds to. Using the average natural language turns per conversation we estimated the sample size per domain. We then selected a random sample of full conversations, using the estimated sample size per domain, from the conversation logs of a live virtual agent we maintain in each domain.

All natural language turns were selected for voting and released to a group of 14 voters. Three votes per turn were required to control for subjectivity. Voters were all employees of Verint Systems who were trained on the Trace user interface and voting process prior to actually voting, and many were domain experts. After voting, the average number of turns per dataset with a clear majority (agree or disagree with the intent chosen by the live IVA) was 6187. If there was no clear majority, the turn was not used for evaluation. Although all three datasets had 14 voters, not all 14 were the same people; there were 17 unique voters overall. As the human reviewers voted on the user turns, the system logged the time required to review each turn and make a determination. The average overall time for a human reviewer to receive a user turn and place a vote was 11.12 s.

Evaluation dataset statistics are given in Table 2. Total user turns involve all forms of user input including clicking on controls and web page navigation events. Textual user turns are only those that were processed by the NLU component for intent classification. As we are only interested in the discovery of error in the NLU, it is these user turns that are evaluated by humans. Majority agreement

are the number of textual user turns where a majority (at least 2) of the three voters agreed. Note that reviewers can choose to vote *not sure* (see Figure 3), so a majority is not guaranteed.

From these counts, we can see that the telecommunications IVA is very interactive, less than half of user turns are actually in the form of natural language. This IVA responds to many user activities besides typed or spoken input. In contrast, the Airline IVA does not accept anything but typed or spoken input. The Train IVA appears a good balance of the two interaction styles. The Train IVA had the highest level of overall voter agreement, at 87 percent. The Airline had less at 76.7 percent followed by the Telecommunications IVA with 71.8 percent agreement. Inspecting the conversations and IVA knowledge bases, it appears these differences are due to the complexity of the IVA and the number of intents understood. The Train IVA has 930 distinct intents in its language model, compared to 1223 for the Airline IVA and 2173 for the Telecommunications IVA. Not surprisingly, the increase in possible intents to select from appears to decrease voter agreement on the correctness of an intent chosen by the IVA.

Comparison metrics

Due to the multiple layers of random sampling used to create the datasets and gather the votes, fairly comparing humans to each other and Trace can be difficult. As the human voters did not see all of the user turns in a dataset, but were merely given a subset of turns ensuring each turn had three votes each, we cannot calculate a recall, and therefore an F1 score, for the humans. Furthermore, no two humans saw the exact same subset of the turns to ensure a pair of voters who only choose one value (always vote *Yes*, for example) could generate an inaccurate majority on an entire subset. Therefore, to compare the human reviewers to each other and to Trace we considered only the class unweighted (micro) and class weighted (macro) precision.

The micro-averaged precision gives a sense of how many “correct” votes a reviewer made over the sample size they reviewed. Equal weight is given to each turn classification decision without regard to class imbalance (Schutze, Manning, and Raghavan 2008). However, as the two classes are very imbalanced in this evaluation (only 14.45 percent are class *No* averaged over the three datasets) this can be misleading if viewed alone. In contrast, the macro-averaged precision gives a sense of effectiveness on small classes (Schutze, Manning, and Raghavan 2008). Taking these two measurements together we can get a sense of a classifier’s (human or machine) performance overall and performance equally favoring the under-represented class of misunderstandings.

TABLE 3 Human (H) voter mean precision \pm 95 percent on majority agreement compared with Trace (T) voting classifier that a turn’s intent was misclassified by domain

Dataset	H_{Macro}	T_{Macro}	H_{Micro}	T_{Micro}
Airline	0.85 ± 0.24	0.74	0.93 ± 0.12	0.89
Telecom	0.82 ± 0.24	0.59	0.89 ± 0.16	0.86
Train	0.79 ± 0.28	0.74	0.84 ± 0.38	0.83

Automating reviewer voting

Beyond prioritizing human reviewer time we wish to automate the entire voting process where possible. To do this we train a binary classifier to vote *Yes* or *No* for each $\langle \text{turn}, \text{intent} \rangle$ pair identical to the reviewer task. Humans would not be entirely replaced however, as the risk indicator weights and voting classifier would need periodic retraining to account for changes in the set of intents within the language model. In this way, Trace is a human-in-the-loop system which automates many of the human review tasks without entirely replacing the valuable human decision making.

To select the voting classifier, we performed an extensive evaluation of various classification methods on each dataset. The voting classifiers were trained using the unweighted risk indicator values as features and the majority decision as the outcome. If voters agreed that turn t belongs to the intent assigned by the IVA, the class is 1. If they disagree, the class is 0. For each turn with a voter consensus, we add a row to a feature matrix M , with a column for each risk indicator and a final column for the class.

This feature matrix M is then used to train a binary classification model using a stratified 30-fold cross validation. When a new turn is under review, the risk indicators are represented as a vector and fed to the voting classifier to predict the majority vote of *Yes* or *No*. The classifiers were trained and evaluated on each dataset in isolation. The classification method with the highest combined precision and fastest training time across all three datasets was chosen. Training time is an important consideration as Trace is continually retraining these models per dataset as human voting data is added. Our final selection for the voting classifier was a Random Forest model with 30 estimators, which only required 2 s on average to train over the 30 folds. Compare this to the Gaussian Process classifier which performed similarly but required 12 min on average to train.

Voting classifier evaluation

Having selected a voting classifier, we compare its performance on each dataset to the human voters in Table 3. For

each voter, we calculated the micro- and macro-precision of their votes to the majority vote. It is obvious from this table that as the IVA complexity increases, T_{Macro} compared to H_{Macro} suffers, although T_{Micro} does not seem affected indicating Trace's performance on class *Yes* still holds. Even in the more complex domains with thousands of intents in the language model, Trace maintains human-level performance in marking turns that were not misunderstood. As this is the much larger class, this demonstrates that Trace is still effective in reducing the volume of conversations necessary for human review even if it is not as accurate at identifying the misunderstood turns for more complex IVAs.

There is a bias favoring the humans here in that the gold standard was produced by the majority of human reviewers. The bias arises when a third reviewer votes on a user turn where there already exists one *Yes* and one *No* vote. In this case, the third reviewer is forming the majority either way they vote and cannot be penalized.

In light of this, the human voter precision scores given may be higher than a true outside observer predicting the existing majority vote as Trace is required to do in this evaluation. It is dangerous to try to correct for this by ignoring votes that form majority however, as turns will not be scored *for* the two humans choosing the majority, but will be scored *against* the one that did not. This gives more chances for penalty than reward. We also cannot only consider turns where all three reviewers agree as the human performance will always be perfect and turns with some disagreement are potentially harder cases we want to evaluate Trace on. Therefore, we only note that the bias exists and favors the human voters.

To measure the magnitude of possible bias, we counted how often a human voted to form the majority when there was an existing split vote and found that this situation occurred in 16.8 percent of the Airline votes, 22.7 percent of the Telecom votes, and 25.1 percent of the Train votes. The human performance numbers at predicting the majority can be considered potentially inflated by a maximum of these percentages, indicating that H_{Macro} for predicting the majority rather than forming it is likely lower than it appears in Table 3.

Annotation cost savings

In production, it costs our company \$0.10 per turn to generate an annotation from a human reviewer. For our telecom IVA, which responds to 1.8 million user turns per month, reviewing 5 percent for quality control requires 90,000 turns to be reviewed every month. Recall this is a very difficult domain in which there are over 2000 unique intents and complex intent classification logic, and is the

domain in which Trace demonstrated the lowest performance. At \$0.10 per turn the annotation cost for 90,000 is \$9000 per month for just one subjective review per turn. Using crowd-source platforms such as Mechanical Turk currently would cost \$0.54 per turn using qualified workers³ and paying them \$0.10 per turn, which would greatly inflate monthly annotation costs. In addition, this IVA has a historical intent error rate of 14.35 percent, meaning only 12,915 turns of the 90,000 actually need review, assuming the 5 percent is a truly random sample. We wish to minimize reviewing turns with no error as they are not used for IVA improvement directly.

Using Trace to prioritize the data for review by the risk score, as the voting classifier is not trustworthy in this domain (see Table 3), we have measured 28 percent of the riskiest 5 percent in a month to be actually misunderstood. This doubles the number of turns that truly needed review in the sample, and therefore we find a similar number of misunderstandings in a 2.5 percent sample from Trace as in a 5 percent random sample, saving 50 percent in monthly annotation costs for a *single* IVA.

In some domains with less complex language models, such as the Train IVA, Trace has similar performance to the average human reviewer (see Table 3). In these cases, we have been able to eliminate the human voters, with the exception of periodic batches to tune the risk indicator weights and voting classifier. The annotation cost savings for these IVAs have been closer to 90 percent. Considering that the Telecom IVA is a worst-case for Trace and that we maintain 40+ production IVAs and growing, by introducing Trace company-wide we save nearly 75 percent in monthly annotation costs, or roughly \$1.5 million yearly, over random samples.

Scalability

The ability for Trace to handle the conversation volume of production IVAs is also very important to us. One of the primary needs is the ability to review all of the conversations from a live IVA, not just a sample as humans do. If Trace is to be viable solution for a company that designs and builds IVAs, it must support not only the unfiltered log size of a single IVA but also of dozens. For the following tests, we deployed a single instance of Trace as four components on Amazon Web Services⁴. The components are a t3.xlarge webserver, two r5.2xlarge Celery nodes, three r5.2xlarge MongoDB nodes, and six m4.10xlarge Slurm⁵ HPC nodes.

We first performed a scale up test to measure how the analysis time increases as conversation log size increases on a fixed set of hardware. For testing, we used 4 weeks of data originating from the conversation logs from the Telecom IVA. This particular IVA handles customer

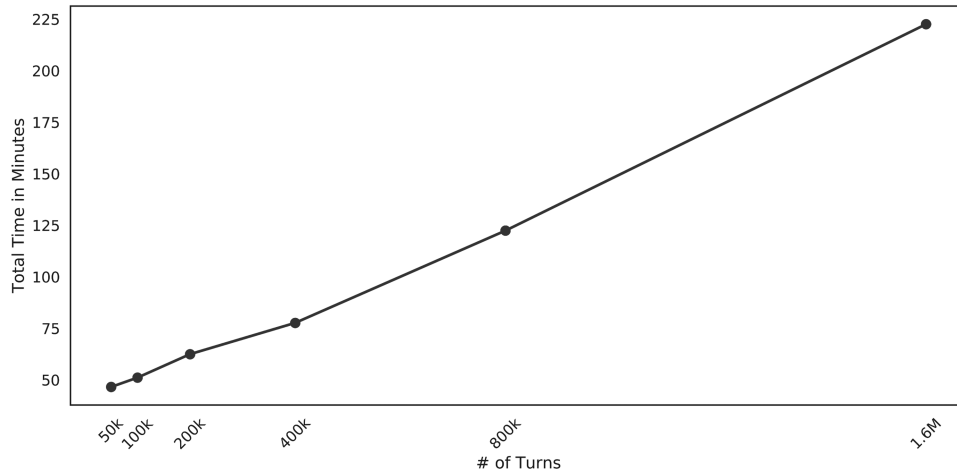


FIGURE 6 Analysis time as conversation log size increases

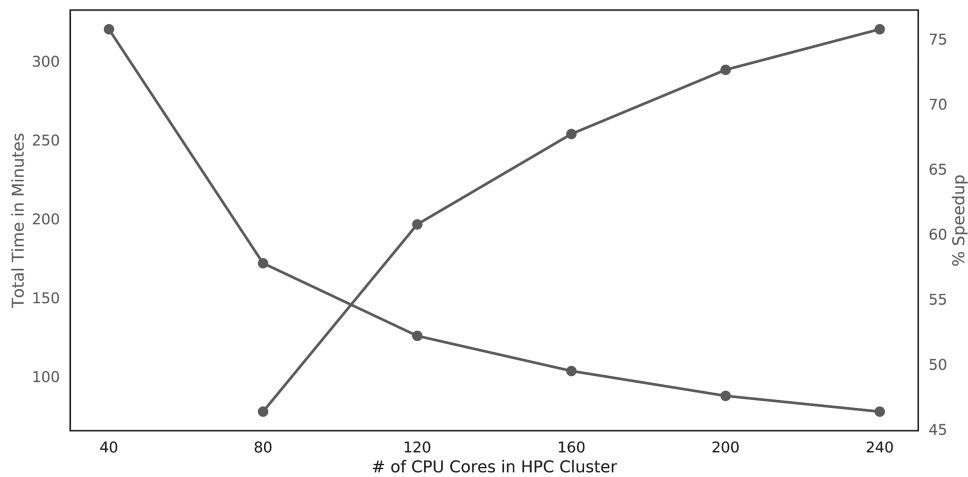


FIGURE 7 The performance impact on processing 400k turns increasing the compute cluster one 40-core node at a time

support issues on the website of a large telecommunications corporation. The logs were partitioned into increasing subsets of size 50k, 100k, 200k, 400k, 800k, and 1.6m turn-response pairs. Beginning with the smallest, each set was fed into Trace, with all six compute nodes enabled, and the total wall clock time was measured to complete the risk analysis and apply the voting classifier to all turns. Then the entire system was restarted to clear out any caches. After restarting, the next largest set was fed into the system and the wall clock time was recorded.

In Figure 6, we see that it takes roughly 48 minutes to complete analysis on 50,000 turn-response pairs. This volume of data is approximately a single days worth of customer interactions for this particular IVA. By 1.6 million turn-response pairs, Trace takes 225 min, or 3.75 h to complete its analysis. This is roughly 1 month worth of customer interactions from the live Telecom IVA.

Next, we perform a scale out test to ensure that the risk analysis process can be scaled as needed to meet volume demands. We start with a single 40-core compute node enabled and feed Trace 400k turns, roughly 1 week of conversations from the Telecom IVA, and measure the wall clock time to complete the risk analysis and apply the voting classifier to all turns. Then the entire system was restarted to clear out any caches. After restarting, an additional compute node is added to the cluster and the test is repeated.

In Figure 7, we see the results of this scaling test. With a single compute node, it takes roughly five and a half hours to process 1 weeks worth of data. With six compute nodes, Trace can process the data in roughly 1 h, at which point MongoDB and the network overhead begin to bottleneck linear scaling. With six compute nodes in our production Slurm cluster, the architecture has proven capable of

processing the conversation data influx of 40+ live IVAs concurrently.

CONCLUSION

We have presented Trace, a system to prioritize human reviewer time and the reduce annotation costs of maintaining production IVAs at scale. In addition, by minimizing the amount of human review necessary, we reduce the amount of user data exposed through the review process. In the best case, where human review can be reduced to periodic system reinforcement, the vast majority of conversations are not seen by humans, while still ensuring IVA quality.

We discussed the design of the risk analysis system, how human reviewers interact with the system, how the analysis is presented to the domain experts, and how it can help domain experts prioritize their time for language model repairs that will have the largest impact on user experience.

Trace relies greatly on previous research in human-computer interfaces, communication, and natural language processing in the development of its indications of risk. To our knowledge, there exists no other similar application for the improvement of deployed IVAs. Trace has been used in a production capacity for over 4 years, processing hundreds of millions of conversational turns per year.

Trace presents voting results and actions to the domain experts through the same interface regardless if the voter was human or machine (see Figure 4) and Trace votes on every turn. Therefore, the source of votes can be dynamically chosen based on current system performance in a particular language domain or human reviewer availability. As Trace is implemented as a web application, domain experts can easily use internal or external annotation sources for voting.

Our system uses only conversational features for misunderstanding detection and is not dependent on the implementation details of the underlying IVA or the domain of language it is deployed in. This combined with the flexibility of annotation sources, its ability to scale to real-world volumes of data, and its proven ability to lower costs makes it a beneficial application to our company or any company that maintains enterprise IVAs or chatbots.

ENDNOTES

¹ <https://appen.com/solutions/crowd-management/>

² <https://www.mturk.com>

³ From <https://requester.mturk.com/pricing> in July 2019

⁴ <https://aws.amazon.com/ec2/instance-types>

⁵ <https://slurm.schedmd.com/>

REFERENCES

- Aberdeen, J. & L. Ferro 2003. "Dialogue patterns and misunderstandings." In ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems. https://www.iscaspeech.org/archive/ehsd_2003/aberdeen03_ehsd.html
- Beaver, I., C. Freeman, and A. Mueen. 2020. "Towards awareness of human relational strategies in virtual agents." In Proceedings of the 34th AAAI Conference on Artificial Intelligence. Palo Alto, California: The AAAI Press. <https://aclanthology.org/2005.sigdial-1.14/>
- Bohus, D. & A. I. Rudnicky 2005. "Sorry, I didn't catch that!-An investigation of non-understanding errors and recovery strategies." In Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue, 128-143. Lisbon, Portugal: Special Interest Group on Discourse and Dialogue (SIGdial).
- Bulyko, I., K. Kirchhoff, M. Ostendorf, and J. Goldberg. 2005. "Error-correction detection and response generation in a spoken dialogue system." *Speech Communication* 45(3): 271-88.
- Freeman, C., and I. Beaver. 2017. "Online proactive escalation in multi-modal automated assistants." In The 30th Florida Artificial Intelligence Research Society Conference (FLAIRS), volume 30: 215-20. Palo Alto, California: The AAAI Press.
- Hewitt, T., and I. Beaver. 2020. "A case study of user communication styles with customer service agents versus intelligent virtual agents." In Proceedings of the 21st Annual SIGdial Meeting on Discourse and Dialogue. Association for Computational Linguistics.
- Higashinaka, R., K. Funakoshi, M. Araki, H. Tsukahara, Y. Kobayashi, and M. 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. Association for Computational Linguistics.
- Jiang, J., A. Hassan Awadallah, R. Jones, U. Ozertem, I. Zitouni, R. Gurunath Kulkarni, and O. Z. Khan. 2015. "Automatic online evaluation of intelligent assistants." In Proceedings of the 24th International Conference on World Wide Web, 506-16. ACM.
- Khatri, C., B. Hedayatnia, A. Venkatesh, J. Nunn, Y. Pan, Q. Liu & H. Song et al. 2018. "Advancing the state of the art in open domain dialog systems through the Alexa prize." arXiv preprint arXiv:1812.10757.
- Kim, B., S. Ryu, G. G. Lee, Kim Byeongchang, Ryu Seonghan & Lee Gary Geunbae. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications* 76(9): 11377-90. <http://doi.org/10.1007/s11042-016-3724-4>
- Kuligowska, K. 2015. "Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents." *Professionals Center for Business Research* 2. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2569637
- Ogawa, A., and T. Hori. 2015. "ASR error detection and recognition rate estimation using deep bidirectional recurrent neural networks." In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, 4370-4. IEEE.
- Ram, A., R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu & J. Nunn et al. 2018. "Conversational AI: The science behind the Alexa prize." arXiv preprint arXiv:1801.03604.
- Roy, S., R. Mariappan, S. Dandapat, S. Srivastava, S. Galhotra, and B. Peddamuthu. 2016. "QART: A system for real-time holistic quality assurance for contact center dialogues." In Proceedings of the 30th AAAI Conference on Artificial Intelligence, 3768-75. Palo Alto, California: The AAAI Press.

- Schutze, H., C. D. Manning, and P. Raghavan. 2008. *Introduction to Information Retrieval*, volume 39. Cambridge University Press.
- Szumilas, M. 2010. "Explaining odds ratios." *Journal of the Canadian Academy of Child and Adolescent Psychiatry* 19(3): 227.
- Zhao, T. & M. Eskenazi. 2016. "Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning." In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 1-10. Association for Computational Linguistics.

AUTHOR BIOGRAPHIES

Ian Beaver is Chief Scientist at Verint Systems, Inc. His research covers topics surrounding human-computer interactions such as gesture recognition, user preference learning, and communication with multi-modal automated assistants. He regularly presents his work at academic and industry conferences and has authored over 40 patents within the field of human language technology. Currently, he is leading a team working to optimize human productivity in text processing activities by way of automation and augmentation, fostering symbiotic relationships with machines. Beaver received his PhD in Computer Science from the University of New Mexico. He also holds a BS and MS in Computer Science with a Specialization in AI from Eastern Washington University. Contact ian.beaver@verint.com.

Abdullah Mueen is an Associate Professor in Computer Science at University of New Mexico. He joined UNM as an Assistant Professor in 2013. Earlier he was a Scientist in the Cloud and Information Science Lab at Microsoft Corporation. His major interest is in temporal data mining with a focus on two unique types of signals: social networks and electrical sensors. He has been actively publishing in the data mining conferences including KDD, ICDM, and SDM, and journals including DMKD and KAIS. He has won Junior Faculty Research Excellence Award from SOE at UNM, runner-up award in the Doctoral Dissertation Contest in KDD 2012, and the best paper award in SIGKDD 2012. His research is funded by NSF, DARPA, and AFRL. Earlier, he earned his PhD degree at the University of California at Riverside and BSc degree at Bangladesh University of Engineering and Technology. Contact mueen@unm.edu.

How to cite this article: Beaver, I., and A. Mueen. 2021. "On the care and feeding of virtual assistants: Automating conversation review with AI." *AI Magazine* 42: 29–42.
<https://doi.org/10.1609/aaai.12024>



It is the generosity and loyalty of our members that enable us to continue to provide the best possible service to the AI community and promote and further the science of artificial intelligence by sustaining the many and varied programs that AAAI provides. AAAI invites all members and other interested parties to consider a gift to help support the dozens of programs that AAAI currently sponsors.

For more information about the Gift Program, please use the donate option when renewing your membership, or write to us at donate21@aaai.org.

AAAI is a 501c3 charitable organization. Your contribution may be tax deductible.