Natural Language Engineering (2024), 1–26
doi:10.1017/S1351324923000372

ARTICLE

Actionable conversational quality indicators for improving task-oriented dialog systems

Michael Higgins[1], Dominic Widdows[1,2], Beth Ann Hockey[1], Akshay Hazare[1], Kristen Howell[1], Gwen Christian[1], Sujit Mathi[1], Chris Brew[1], Andrew Maurer[1†], George Bonev[1], Matthew Dunn[1] and Joseph Bradley[1]

[1]LivePerson Inc. New York, NY, USA and [2]IonQ Inc. College Park, MD, USA
Corresponding author: Dominic Widdows; Email: widdows@ionq.com

(Received 22 September 2021; revised 7 July 2023; accepted 8 July 2023)

Abstract

Automatic dialog systems have become a mainstream part of online customer service. Many such systems are built, maintained, and improved by customer service specialists, rather than dialog systems engineers and computer programmers. As conversations between people and machines become commonplace, it is critical to understand what is working, what is not, and what actions can be taken to reduce the frequency of inappropriate system responses. These analyses and recommendations need to be presented in terms that directly reflect the user experience rather than the internal dialog processing.

This paper introduces and explains the use of Actionable Conversational Quality Indicators (ACQIs), which are used both to recognize parts of dialogs that can be improved and to recommend how to improve them. This combines benefits of previous approaches, some of which have focused on producing dialog quality scoring while others have sought to categorize the types of errors the dialog system is making. We demonstrate the effectiveness of using ACQIs on LivePerson internal dialog systems used in commercial customer service applications and on the publicly available LEGOv2 conversational dataset. We report on the annotation and analysis of conversational datasets showing which ACQIs are important to fix in various situations.

The annotated datasets are then used to build a predictive model which uses a turn-based vector embedding of the message texts and achieves a 79% weighted average f1-measure at the task of finding the correct ACQI for a given conversation. We predict that if such a model worked perfectly, the range of potential improvement actions a bot-builder must consider at each turn could be reduced by an average of 81%.

Keywords: Dialogue; Natural language interaction; Task-oriented dialogue; Customer service

1. Introduction and outline

Customer service dialog systems have become widely used in many everyday settings, but still make frustrating errors that are sometimes obvious to a human—for example, failing to understand a customer's message and asking an irrelevant question as a result. In practice, tuning these systems to limit these behaviors is an expensive and time-consuming art. This paper describes the design, implementation, and early results of an approach to improving overall dialog system quality by recognizing and addressing such individual failures. This is done by combining individual Actionable Conversational Quality Indicators (ACQIs) with a running interaction quality score (IQ), to show which problems were identified, what steps can be taken to fix them, and how these

[†]Work done before joining Amazon

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

issues affected an overall assessment of the user experience. IQ is a standard conversational quality measure developed by Schmitt, Schatz, and Minker (2011), Schmitt and Ultes (2015). This paper introduces ACQIs, which are designed so that each conversational quality indicator has associated recommended actions, but do not require explicit knowledge of the dialog system architecture.

The paper starts by explaining some of the background on how task-oriented dialog systems are built and maintained (Section 2): while there are several online tools that support this, many readers may be unfamiliar with their use. Previous works on evaluating dialog system performance (discussed in Section 3) have investigated the use of conversation-level quality metrics and individual turn-level assessments of good and bad interactions. In particular, the IQ score of Schmitt *et al.* (2011), Schmitt and Ultes (2015) is discussed in this section, and the combined use of ACQIs and IQ score plays a major role throughout the rest of the paper. Section 4 introduces the datasets that are used for examples and experiments throughout the rest of the paper.

Section 5 explains the heart of this paper: the design of the ACQI taxonomy.[a] This explains the motivation and decisions behind ACQIs, including how they are made to be actionable, explainable, and to give feedback that is specifically tailored to the dialog system in question. ACQIs are agnostic of dialog system architecture, relying only on textual input, but able to be mapped to any number of associated actions. Section 6 describes the annotation work for ACQI datasets, analysis of the ACQI distributions, and the work on combining ACQI and IQ scoring to distinguish those parts of the dialog that need particular improvement.

Section 7 describes experiments on automatically predicting ACQIs based on the annotated datasets and features extracted from the dialog text. We demonstrate that correct ACQI labels can be predicted with a weighted average f1-score of 79% and demonstrate the effectiveness of textual features to predict labeled IQ scores (1–5) with an average accuracy of 60%. Analyzing the distribution of ACQIs and suggested actions when IQ drops shows that the number of potential improvement strategies to evaluate could be reduced by up to 81%, depending on the accuracy of the ACQI and IQ classifiers. We argue that tools built using these approaches could improve the effectiveness and reduce cognitive burden on bot-builders.

2. Conversational AI systems and challenges for bot-builders

Since the year 2000, dialog systems (often called chatbots) have gone from mainly research demonstration systems to include various user-facing commercial offerings. Dialog systems fall into three broad categories (Deriu *et al.* 2020): conversational agents, question answering systems, and task-oriented systems (which are the topic of this paper). Each category has corresponding dialog quality measurement strategies. Conversational agents, which often receive the most attention in news articles when released, are typically unstructured and open-domain, with no particular objective other than an engaging conversation. Success in conversational agents is frequently measured by how long users are willing to continue interacting. Question answering systems can be evaluated by considering the accuracy of answers given.

Task-oriented systems, which are the focus of this article, typically have a rigid structure and a limited scope. Most commercial and many research dialog systems have an underlying modular structure as in Fig. 1. These systems are built to resolve the consumer's issue, answer questions, route to an appropriate representative, or guide the user through a task as efficiently as possible. Like question answering systems, task-oriented systems have clear "failure" cases: just as a question answering system can fail to answer a question, a task-oriented system can fail to complete a task.

[a] The term "taxonomy" is used here to mean an agreed categorization, and is used for classifications of dialog systems, intents, and failure states, in addition to the quality indicators and actions of the ACQI taxonomy. Tree-like "taxonomic" relationships can exist between these categories but are not a must.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

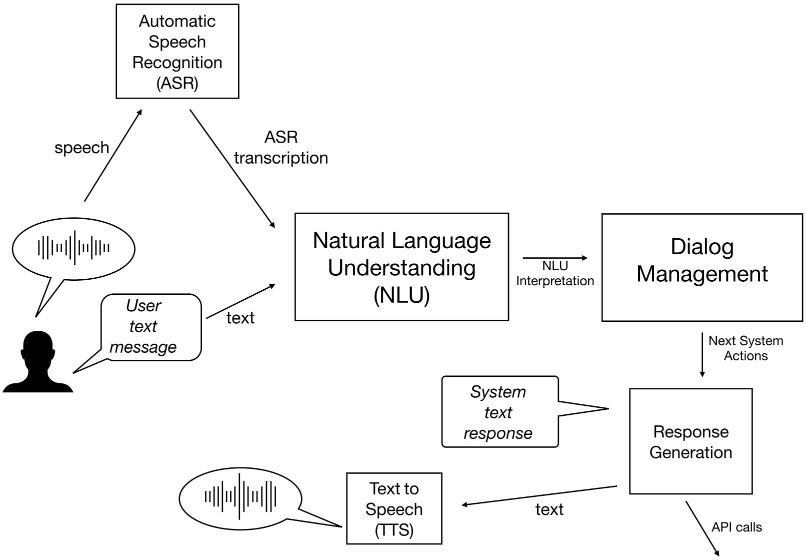


Figure 1. Typical dialog system architecture illustrating components for both spoken and text-based input and output.

Task-oriented systems are prevalent in customer service: ideally, they can automate simple tasks like routing requests to the right agent, thus freeing up human customer service specialists to handle more demanding situations. Several technology companies offer dialog system services to support such chatbots. These include offerings from large and general technology companies such as Microsoft LUIS, IBM Watson, Google DialogFlow, and from more specialist providers such as Salesforce, Intercom, and LivePerson.

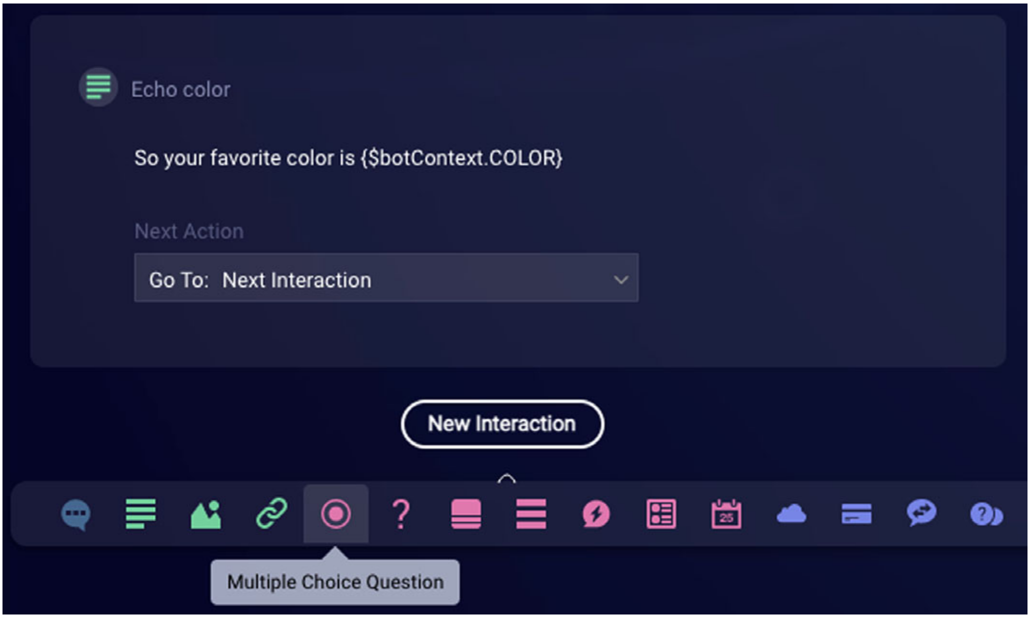
While there are several differences between these platforms, there are some typical themes:

- The platforms provide various tools and widgets for customers to build and deploy their own dialog systems.
- The dialogs built in this way are "designed" or "scripted." The process of building a dialog involves declaring various steps, inputs the user can be encouraged to make at those steps, and what action the system should take in response.
- This process admits the possibility of error or failure states, when the input from a user is something the system (knowingly or unknowingly) cannot process successfully.

A simple example might be that the dialog is at a stage where the user is asked to pick from a list of available options, such as "Enter 1 for English, Ingrese 2 para Español." In this case, any entry other than the numbers "1" or "2" would be a problem. This problem can be addressed in a few ways. A traditional keyboard interface might say "Option unrecognized, please type 1 or 2." A form-filling approach might be to use a "Select from Menu" element instead of a "Prompted Text Entry" element, so that the customer can only give input that corresponds to the actions the dialog system can take right now. An example from the LivePerson Conversation Builder is shown in Fig. 2. The dialog developer is about to add a new interaction to the conversation so far, they have a choice of widgets including those for text entry, scheduling, payments, and there are various formats for asking questions and tracking the responses. There are analogous features in other bot-building platforms.

It is crucial to note that dialog system platforms are typically used by customer support specialists, not the customers themselves. In the rest of this paper, these users will be described using the colloquial but industry-wide term "bot-builders." This is emphasized because one of the ways

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press



Echo color

So your favorite color is {$botContext.COLOR}

Next Action

Go To: Next Interaction

Figure 2. A dialog system-building interface where the bot-builder is about to add a multiple-choice question.

to improve these dialog systems is *to provide tools that make bot-builders more effective*. The work described in this paper shows that two established approaches to measuring conversations, scoring individual actions and assessing the conversation quality as a whole, can be combined to provide such a tool that helps bot-builders to identify and fix particular pain points in a deployed dialog system. The intuitively appealing assumption that useful actionable feedback about dialog systems can be produced through (semi-) automated methods is supported by Hockey *et al.* (2003), which shows that users of a task-oriented dialog system who received actionable feedback in failure cases outperformed the control group that did not.

3. Previous approaches to evaluating dialog systems, including IQ

Meaningful evaluation of automated dialog system performance has long been recognized as crucial to progress in dialog system research (Deriu *et al.* 2020) and is an active area of development, including the recent introduction of the “sensibleness and specificity” metric of Adiwardana *et al.* (2020). Since the introduction of dialog system-building platforms as described in Section 2, it has also become a daily concern for customer service bot-builders.

One of the most common industry measures for dialog system effectiveness is the *automation rate* or *containment rate*. Containment rate directly affects the cost savings that a business may make through automation, for example *“Over three years and a conservative 25% containment rate, the cost savings is worth more than $13.0 million to the organization”* (Forrester Research 2020). However, analysis of containment or automation rates is still relatively rare in the research literature and is mainly found in evaluation of speech/ voice-driven dialog systems (Pieraccini *et al.* 2009). The tradeoff between the automation rate and the number of setbacks a user can encounter is analyzed by Witt (2011), again with spoken dialog. The relative lack of research literature on containment rates can be easily attributed to different settings and incentives. Task-oriented commercial dialog systems are often developed to support existing customer service operations staffed by human agents, whereas most dialog systems in research settings do not have human agents to respond to escalations, so the systems cannot escalate, and containment cannot be measured.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

For the task-oriented dialog systems that are our primary concern, there is an underlying concept of giving a "right" response. Early evaluation work based on this idea compared the actual responses to a predefined key of reference answers (Hirschman *et al.* 1990). The portion of matches to the key gave the measure of performance. Well-known weaknesses of this approach include being specific to particular systems, domains, and dialog strategies. More portable strategies that measure inappropriate utterance ratio, turn correction ratio, or implicit recovery (Polifroni *et al.* 1992; Shriberg, Wade, and Price 1992; Hirschman and Pao 1993; Danieli and Gerbino 1995) are intellectual ancestors of the ACQI part of our approach, in that they identify events that are indicators of the quality of the conversation. Both of these early approaches share the limitation of being unable to model or compare the contribution that the various factors have on performance.

The PARADISE approach (Walker *et al.* 1997) overcomes this limitation by using a decision-theoretic framework to specify the relative contribution of various factors to a dialog system's overall performance. This and other ideas introduced by PARADISE, such as separating the accomplishment of a task from how the system does it, support evaluation that is portable across different systems, domains, and dialog strategies. We have tried to emulate some of these best practices by defining ACQIs that avoid being too implementation-specific, though without insisting that all ACQIs must be relevant to all dialog systems (see Section 5). Also, for our IQ models, in contrast with much work on IQ (Schmitt *et al.* 2011; 2012; Schmitt and Ultes 2015; Stoyanchev, Maiti, and Bangalore 2017), we have chosen not to utilize any features extracted from the dialog systems themselves, such as semantic parse from the natural language understanding system (NLU) or features from an automatic speech recognition (ASR) component because they may not be available (e.g., there are no ASR features in a text-only system, and not all NLU produces semantic parses). Instead in this paper, we rely solely on the text of the dialog, partly to facilitate comparison, and partly in order to improve portability across a wider variety of dialog systems (though this remains challenging, as shown in the experiments below).

3.1. IQ score

More recent work has developed several dialog quality measurement strategies which are categorized by Bodigutla, Polymenakos, and Matsoukas (2019) as follows: sentiment detection; per-turn dialog quality; explicitly soliciting feedback from the user; task success; and dialog-level satisfaction ratings as in PARADISE (Walker *et al.* 1997). These methods are useful but have well-known limitations: for example, sentiment analysis on messages misses many problems, there is response bias in user polling, and negative outcomes weigh more in the consumer's mind than positive ones (Han and Anderson 2020).

The per-turn dialog quality approaches such as Response Quality (Bodigutla *et al.* 2020) or IQ (Schmitt *et al*. 2011; Schmitt, Ultes, and Minker 2012; Schmitt and Ultes 2015) are based on turn-by-turn expert annotator ratings on a five-point scale that contribute to a running evaluation of the dialog up to that point.

The method in this paper most directly builds on the IQ method of Schmitt *et al.* (2011), Schmitt and Ultes (2015). We also draw from later work of Stoyanchev, Maiti, and Bangalore (2019) which applies the Schmitt *et al.* (2011), Schmitt and Ultes (2015) IQ method to customer service dialogs. One reason we preferred IQ over the similar RQ approach is that with three positive states and two negative, RQ's scale is less useful for identifying problems, and we would be surprised if in the data we are studying, the "excellent" rating was ever used. IQ's "dissatisfaction scale" with mostly negative ratings was better but still not ideal for our purposes. This led us to alterations of the IQ scale, which are discussed in detail in Section 6.2. We also preferred IQ to RQ because IQ has been shown to correlate with user satisfaction (Schmitt and Ultes 2015) and because of the availability of the IQ annotated LEGOv2 dataset and comparable prior work. In IQ annotation of a conversation, one point is typically added for a good interaction and a point is

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 1. Summary description and statistics on parts of LEGOv2 and LivePerson datasets annotated and used in this work

Bot	Vertical	Bot functions	Avg # turns	# Conversations
LEGOv2	Travel	Transactional	13.67 ± 11.25	541
Junior Sales Assistant	Retail	Data gathering, routing	4.95 ± 2.13	130
Help and Route	Travel	Routing, FAQ	5.89 ± 3.71	130
Router	Tech	Routing	2.25 ± 1.45	130
Food Expert	Food industry	Transactional (Ordering)	5.44 ± 3.31	130

subtracted for a bad interaction, with some exceptional cases, for example when the dialog "obviously collapses." A benefit of this method is that it can help to identify *where* there are problems in a dialog system, which is noted as a contrast and improvement over previous methods by Schmitt *et al*. (2011):

While the intention of PARADISE is to evaluate and compare SDS or different system versions among each other, it is not suited to evaluate a spoken dialog at arbitrary points during an interaction. (Schmitt *et al*. 2011), Section1)

Our work can be seen as an extension of the IQ approach in our use of a running dialog quality measure. Adding ACQI improves over IQ alone by recommending how to resolve a problem instead of only identifying where it exists.

4. Datasets used in this work

The experimental work in the paper uses two main datasets, which were annotated and used in the prediction experiments in subsequent sections. Statistics about these datasets are summarized in Table 1.

4.1. LEGOv2

LEGOv2 (Schmitt *et al*. 2012; Schmitt and Ultes 2015) is a parameterized and annotated corpus derived from the CMU Let's Go database from Raux *et al.* (2005, 2006). This corpus was developed by the Dialogue Systems Group at Ulm University and was used in the development of the IQ score of Schmitt *et al*. (2011), Schmitt and Ultes (2015), discussed in Section 3.

The data consist of phone-mediated customer service interactions between an automated dialog system and callers drawn from the general population in the vicinity of Pittsburgh, Pennsylvania. The LEGOv2 dataset represents many of the characteristics that are still challenging when deploying task-oriented spoken language systems "in the wild":

- The callers are drawn from the general population.
- The task-at-hand is authentic: callers have a presumed need to access bus information.
- The callers are using standard personal or public telephones in real-world settings that include such challenges as third-party speech, television programs in the background, very variable audio quality, and irritated and or amused speech.

Things that have changed since the initial creation of LEGOv2 include:

- Speech recognition technology is far more capable now than it was when Let's Go began.
- The public is much more familiar with conversational AI systems.
- Both speech-mediated and text-mediated dialog systems are commercially important and widely deployed, so lessons learned from Let's Go have greater potential impact.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Previous approaches for modeling quality in LEGOv2 have included features automatically extracted from the dialog system (Asri *et al.* 2014; Rach, Minker, and Ultes 2017; Stoyanchev *et al.* 2017; Ultes 2019) or have used meta-data (Ling *et al.* 2020). We have chosen to use only features derived from the text itself from a transcribed version of LEGOv2, as this approach is also applicable to text-based dialog systems including those developed by LivePerson and avoids depending on particular system implementation decisions for features.

4.2. Datasets from livePerson dialog systems

An approved[b] collection of transcripts of text-based customer service conversations with LivePerson dialog systems was also extracted and annotated.

We have intentionally chosen a set of bots that serve different functions, come from different industries, and have different overall quality (as measured by final IQ score). To this end, 130 conversations were chosen from each of 4 dialog systems, giving a total of 520 conversations, a comparable number of conversations to those used from the LEGOv2 dataset, though the LivePerson conversations themselves on average have fewer turns (Table 1).

These dialog systems often respond with structured content, such as embedded HTML with buttons, toggles, or drop-down menus. In these cases, we represent the text of the options separated by "***," for example *"BUTTON OPTIONS *** Main Menu *** Pick a Color *** Pick a different item."*

5. The design of the ACQI taxonomy

This central section describes the design of the ACQI taxonomy and how it extends the IQ system introduced above.

A running score like IQ allows bot-builders to identify dialog system responses where there is a meaningful decrease in score, but does not provide direct diagnosis of the problems that may be there. If nothing other than the running score is available, bot-builders have little option other than to manually review, form their own taxonomy of failure causes, and come up with appropriate fixes. This process in practice typically takes days or weeks to complete and is prioritized largely by operator intuition.

To make the problems explicit and to suggest solutions, we introduce *Actionable Conversation Quality Indicators* or ACQIs. ACQIs highlight moments in chatbot conversations that impact customer experience. In previous work, Finch and Choi (2020) list 21 dimensions across 20 publications that human evaluators have used to measure the quality of open-domain dialog systems. While these dimensions are not used directly in our work, they partially inspire our taxonomy of ACQIs (Table 5, leftmost column). For instance, our "Doesn't Understand" ACQI is inspired by Coherence (Luo *et al.* 2018; Wu *et al.* 2019), Correctness (Liu *et al.* 2018; Wang *et al.* 2020), Relevance (Moghe *et al.* 2018; Lin *et al.* 2019; Qiu *et al.* 2019), Logic (Li and Sun 2018) and Sensibleness (Adiwardana *et al.* 2020). Other elements of our taxonomy are informed by Jain *et al.* (2018) who provide a set of best practices when developing dialog systems for messaging-based bots. We separated "misunderstanding" into "Input Rejected," "Ignores Consumer," and "Does Not Understand" categories, because each of these requires different actions that can mitigate the understanding issue. Such separation of failure states highlights the design-for-actionability of the ACQI taxonomy.

Table 2 shows our complete ACQI taxonomy, including descriptions, and a targeted range of potential actions for bot-builders to take. In addition to making the ACQIs actionable, the taxonomy aims to make the issues aggregable, so that bot-builders can analyze aggregated statistics

[b] The approval and annotation process includes checking all aspects of security and legal compliance and scrubbing personally identifiable information from messages.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 2. ACQIs with their associated actions in example text and spoken dialog systems (NLU = Natural Language Understanding. ASR = Automatic Speech Recognition)

ACQI	Description	LivePerson action	LEGOv2 action
Does Not Understand	The bot says that it does not understand the consumer's response and says something like "I didn't get that" or "I don't understand." This also includes when the bot's response is incorrect and it is clear that the bot has misunderstood what the consumer said	Add/Remove Features, Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add FAQs, Add/Update Bot Statements, Add/Update Bot Prompts, Add/Update Max No Match Behavior	Add/Remove Features, Update ASR Training Data, Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add/Update Consumer-Requested Repeat, Add FAQs, Add/Update Bot Statements, Add/Update Bot Prompts, Add/Update Max No Match Behavior
Input Rejected	The bot does not accept a consumer response to menu options, forms, or other structured content, such as "1, 2, 3, 4," "a, b, c, d," or "yes, no"	Improve Response Flexibility, Add/Update Bot Statements, Add/Update Bot Prompts, Add/Update Max No Match Behavior	Improve Response Flexibility, Add/Update Bot Statements, Add/Update Bot Prompts, Add/Update Max No Match Behavior
Ignored Consumer	The dialog system does not recognize a free text consumer response and asks an unrelated question	Set Delay Expectations, Enable/Disable Context Switching, Add FaaS code, Confirm Successful Bot Handoff, Handle Backend Errors	Set Delay Expectations, Enable/Disable Context Switching, Add/Update Turn-Taking Signals, Add FaaS code, Add/Update Max No Input Behavior, Handle Backend Errors
Restart	When explicitly asked for by the consumer and only used when the conversation starts over within the same engagement with the bot. Includes when the bot goes back to the main menu	Add/Remove/Update Confirm Consumer Statement(s), Add/Update Conversational Navigation, Add/Update Help, Add/Update Max Restart Behavior	Add/Remove/Update Confirm Consumer Statement(s), Add/Update Conversational Navigation, Add/Update Help, Add/Update Max Restart Behavior
Bad Transfer	The dialog system attempts to transfer the consumer to an agent, but either leaves them hanging or abruptly ends the chat. It might also fail to tell them EARLY enough in the conversation that there are no agents available at that hour	Set Transfer Expectations, Confirm Successful Transfer	Transfer was not possible in the CMU system leading to the LEGOv2 dataset, since its only use is out of hours. Set Transfer Expectations makes sense, but the issue never arose, because Bad Transfer was never used
Unable to Resolve	The bot explicitly states that it cannot provide the information the consumer requested and does not offer to transfer to an agent	Add/Remove Features, Offer External Solutions, Set Transfer Expectations	Add/Remove Features, Offer External Solutions, Set Transfer Expectations
Ask for Information	The bot asks the consumer for information	Remove Unnecessary Questions, Set Delay Expectations, Enable/Disable Context Switching, Add FaaS code, Confirm Successful Bot Handoff, Handle Backend Errors	Remove Unnecessary Questions, Set Delay Expectations, Enable/Disable Context Switching, Add/Update Turn-Taking Signals, Add FaaS code, Add/Update Max No Input Behavior, Handle Backend Errors

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 2. Continued

ACQI	Description	LivePerson action	LEGOv2 action
Provide Assistance	Bot responds to consumers' query, attempting to fulfill their intent	Add/Remove/Update Validate Bot Resolution, Add/Remove Features, Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add FAQs, Add/Update Bot Statements, Add/Update Bot Prompts	Add/Remove/Update Validate Bot Resolution, Add/Remove Features, Update ASR Training Data, Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add FAQs, Add/Update Bot Statements, Add/Update Bot Prompts
Ask for Confirmation	Bot asks for user confirmation of input. Includes when the bot asks if the information provided is correct	Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add/Update Bot Statements, Add/Update Bot Prompts	Update ASR Training Data, Update NLU Taxonomy, Update NLU Training Data, Add/Remove/Update Confirm Consumer Statement(s), Add/Update Bot Statements, Add/Update Bot Prompts

about conversations and prioritize fixing the most prevalent issues accordingly. By exposing predicted ACQIs in an appropriately aggregated format, we empower bot-builders to make more data-driven decisions when improving their dialog systems.

The ACQIs and the taxonomy dimensions are derived from analyzing the user experiences directly at the message level, as well as: literature on the (human) evaluation of open-domain dialog systems, consultation with experts, and ongoing feedback from expert users involved in bot-building. The ACQIs in our taxonomy can have negative, neutral, or positive impact on the conversation (see Section 6.4 for an exploration of positive, negative, and neutral changes in IQ for each ACQI). The ACQIs are dialog system architecture-independent, in that we use only the text of the user's input and the system's output for annotation and training predictive models. A text version of the input and output, the actual language of the conversation, is available in nearly all systems since even in spoken systems the ASR produces a written representation of its result. This approach is in contrast to using system-specific indicators, such as backend errors that depend on a particular type of functionality or ASR confidence scores that depend on having ASR. Moreover, the taxonomy is such that not all ACQIs are relevant to all dialog systems: for example, the LEGOv2 system does not support transfer to a human agent, so "Bad Transfer" cannot occur (though the associated action "Set Transfer Expectations" can still potentially be useful, for example by indicating a need to inform the user there are no human agents available).

Because IQ + ACQI will be aggregated and used to improve consumer experience (CX), we put an explicit emphasis on CX and actionability. In order to connect our ACQIs to actions available to bot-builders, we carried out a series of interviews with bot-builders and assembled a repertoire of tuning actions based on their practices, with descriptions and examples for each action. We identified 28 distinct actions bot-builders take at LivePerson. In the case of LEGOv2, we consulted with 2 domain experts and identified 31 actions.

Once the action set was defined, we next needed to attach the actions to ACQIs. A LivePerson expert in improving dialog systems created a proposal mapping each action onto any applicable ACQI(s). The expert then conducted interviews with LivePerson bot-builders to validate the mappings. After validation, we transposed the relationship so that for each of our ACQIs there was an assigned set of actions that the bot-builder may make (Table 2). The mapping showed that our ACQIs could guide bot-builders to take 23 of 28 possible actions for LivePerson systems and 25 of 31 possible actions for LEGOv2.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 3. Average minimum and final IQ scores from annotation

Bot	Ave final IQ score	Ave min IQ score
LEGOv2	3.78	2.44
Junior Sales Assistant	3.43	3.15
Help and Route	3.79	3.44
Router	3.74	3.61
Food Expert	4.08	3.48

5.1. Desirable properties of a ACQI taxonomy

A good ACQI taxonomy should be actionable, easy to understand, have highly bot-dependent ACQI incidence rates, and have a significant impact on consumer experience. In the following sections, we will justify these properties and provide measurement strategies where appropriate. For some ACQIs and dialog systems, it may be appropriate to bypass the need for annotation by automatically extracting ACQIs from the system logs. An example for this would be an ACQI that indicates that the NLU returned a confidence score less than a predefined threshold and responded with a request for the user to rephrase their intent (Table 3).

5.1.1. Actionable

Without actionable ACQIs, bot-builders are left to the time-consuming process of reviewing large amounts of transcripts and/or a careful analysis of the model's feature space, from which they can attempt to deduce what mitigation strategies are appropriate. Unfortunately, many features are not actionable from the bot-builder's perspective. For instance, conversation length can be highly predictive of conversation quality, but from their perspective, there are no clear steps to uniformly reduce conversation length. The problem is that conversation length is not the cause of a bad conversation, it is a consequence of it.

For example, for the dialog system "Food Expert," the initial turn was predicted to have ignored a consumer's initial intent. This can lead to longer conversations, as can an order with complicated modifications, but the associated dialog quality improvement strategies for the operator are quite different. ACQI allows for separation and sizing of these situations: the second is varied and complicated, whereas the first is a single problem with a negative effect on IQ. From the bot-builder's perspective, this one is a relatively easy fix: they just need to make sure that intent recognition is applied to the first customer utterance.

5.1.2. Easy to understand

For bot-builders, understanding is a necessary condition for fixing an issue. The terms used in the ACQI taxonomy are deliberately chosen to be familiar to bot-builders and have been refined with use to add clarity where requested. The relative success of this effort is reflected in the encouraging inter-annotator agreements reported in Table 4 below. However, this finding is potentially influenced partly by the use of experts in bot-building and conversation modeling as annotators. This has yet to be corroborated with less experienced bot-builders.

5.1.3. Reusable where appropriate

The ACQI taxonomy in Table 2 is designed to fit a wide breadth of task-driven dialog systems, and the considerable overlap between the LivePerson and the LEGOv2 columns reflects the fact that

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 4. Annotator agreement for annotating IQ and ACQI, showing linear weighted Cohen kappa (LWCK), unweighted average recall (UAR), Spearman rank correlation (ρ) for IQ and Cohen kappa (CK) for ACQI. Following Schmitt and Ultes (2015), we take the average agreement across each pair of annotators

	IQ			ACQIs
Bot	UAR	LWCK	ρ	CK
LEGOv2	0.61	0.65	0.76	0.71
Junior Sales Assistant	0.46	0.34	0.21	0.64
Help and Route	0.55	0.50	0.53	0.67
Router	0.39	0.49	0.64	0.78
Food Expert	0.60	0.63	0.71	0.60
All (macro-averaged)	0.52	0.52	0.57	0.68

many of the ACQIs and actions are pertinent to both settings. However, the ACQI taxonomy for two dialog systems should only be shared insofar as this makes sense for the two systems. A key example is that many research systems do not have the ability to transfer the dialog to a human agent, so “Bad Transfer” is not applicable. Another example is the “Unable to Resolve” ACQI: this situation is common in many dialog systems, but the specific reasons why a resolution is not possible depends on what other backend capabilities the dialog system can call upon. For example, two dialog systems may both understand that a user wishes to reschedule an appointment, but if they have different levels of access to calendar data and booking operations, this may be resolvable by one system and not the other.

5.1.4. Bot dependent ACQI reports

As the purpose of the ACQI taxonomy is to guide bot-builders to appropriate fixes, it is imperative that the rates at which ACQIs are present are bot-dependent. Teams working on very specific dialog systems may wish to use a subset or create their own smaller taxonomy. For instance, if the dialog system does not allow for transfers (this is the case for LEGOv2) “Bad Transfer” should be excluded from the taxonomy. There is clearly a design tension between these bot-specific concerns and the desire for the taxonomy to be reusable where appropriate. This tension is a challenge from the point of view of dialog system research, though it is natural in the context of customer service as a whole. Taking again the example of a request to reschedule an appointment: the backend integration to reliably automate such requests is nearly always considerably larger than the work of adapting the ACQIs themselves. Manual maintenance and adaptation of the ACQIs taxonomy is often a comparatively smaller concern for commercial than for research dialog systems, whereas the need for ACQIs dedicated to the needs of various situations is sometimes more pressing.

5.1.5. Impact on customer experience

Each ACQI should have an actionable relevance to CX. As we demonstrate in Section 6.4, several elements of the taxonomy only indicate a poor customer experience in particular circumstances. For example, the “Ask for Confirmation” is more likely to indicate a negative CX when it occurs multiple times within a given dialog.

This concludes the summary of the ACQI taxonomy itself. In practice, we found that ACQIs were most reliably predicted and effectively used in combination with a running quality score. This work is described next.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

6. Combining ACQIs with IQ in conversational datasets

The section describes the work done on annotating ACQIs along with IQ, which turned out to be necessary for distinguishing those ACQIs that warrant action. This is due to the fact that dialog context matters. For instance, a system attempting to correct a misunderstanding of a malformed user statement is quite different than the system failing to understand an unambiguous answer to a direct question.

6.1. Observations from annotating ACQIs

Our initial approach to building a model for finding ACQIs and making associated recommendations to bot-builders was based on the assumption that particular ACQIs are bad for the user experience and should be avoided. This turned out not to be the case. As the conversations from the LivePerson datasets were annotated, we observed that ACQIs alone may be bad or good or neither. Asking for confirmation (i.e., "Ask for Confirmation" ACQI) contributes to a good conversation when the user input is genuinely vague, but results in a poor conversational experience when the user input were clear and the system should have understood. For example, asking the user to confirm that "next Wednesday" refers to a particular calendar date is sometimes helpful (especially if today is Tuesday!). Asking if the user's response was "3" if the user just selected "3" can by contrast be obvious and irritating. If a consumer uses "Restart" one time, it can be seen as a good signal that the consumer requested to go back and the system responded appropriately, but if it is used more frequently it can be a signal that their request is not being properly handled. Based on ACQI designations alone, bot-builders cannot always be sure whether a corrective action is needed. Essentially, ACQIs are context-dependent and combine (with context and themselves) nonlinearly. In this work, we explore the context dependency via the relationship with IQ. We leave a more structured statistical modeling framework for future work.

6.2. Annotating ACQIs and IQ together

To mitigate the issue of ACQI instances not being universally good, bad, or neutral, it was decided to combine ACQIs with IQ scores. There are several differences between the guidelines given in Schmitt *et al.* (2011), Schmitt and Ultes (2015) and our own. Most importantly, the removal of all guidelines relating to how much the score can be increased/decreased. Given restrictive guidelines, the change in IQ is almost entirely (99.7%) in increments of 0,1, or -1. From observations in negativity bias (Rozin and Royzman 2001), we know that humans are biased toward giving greater weight to experiences with undesirable behaviors. Allowing larger changes gives annotators the ability to reflect this. The impact of the removal of these guidelines can be seen in Fig. 3. In spite of these changes, our annotator agreement has slightly increased with $\rho = 0.69, 0.72$ for them and $\rho = 0.76$ for us.

We also altered the "dissatisfaction scale" used by Schmitt and Ultes (2015) (5-satisfactory, 4-slightly unsatisfactory, 3-unsatisfactory, 2-very unsatisfactory, 1-extremely unsatisfactory) to 5-good, 4-satisfactory, 3-bad, 2-very bad, 1-terrible. While still retaining a dissatisfaction bias, we added the "good" category as we wanted to allow for the possibility of systems having good interactions. While we largely agree with Schmitt and Ultes (2015) that a dissatisfaction scale has advantages for the task of identifying problems in a conversation, we find utility in having an additional positive option. Identifying good interactions can provide models for bot-builders of what works and the scale should be more robust to a future in which these systems perform well more of the time. The range of our IQ scores allows *room* for improvement even if currently the improvement is infrequent or not technically feasible. In addition to wanting to build on the IQ approach's success, we stayed with a five-point scale rather than a simpler 3-point scale such as "good/neutral/bad" because the five-point scale fit the range of interactions better. Conversational

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

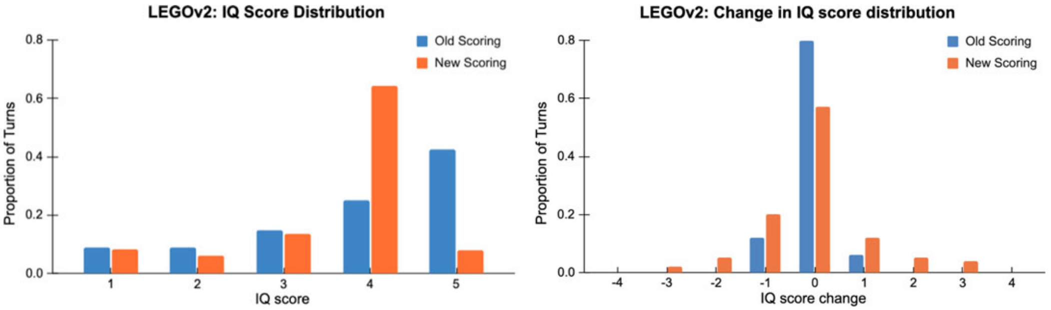


Figure 3. IQ score distribution between IQ annotations in Ultes *et al.* (2015) and the work reported in this paper.

interactions move the conversation forward smoothly ("good") or clumsily ("satisfactory") or something has gone wrong, for example the system has misunderstood and/or done the wrong action ("bad," "very bad" or "terrible"). It is not clear what a neutral interaction would be, and reduction to a binary positive/negative labeling is too simplistic. With a range of "bad" values, bot-builders can use those values to prioritize which problem to address first.

Finally, the annotation guideline from Schmitt and Ultes (2015) dictating that each conversation should start with a satisfactory rating was removed. Although annotators were encouraged to start conversations with a "Satisfactory" rating, they were permitted to assign other ratings when motivated, due in large part to issues we observed wherein a consumer could initiate a conversation, and sometimes have the bot ignore or misunderstand their first message.

6.3. Annotation and inter-annotator agreement

Annotation of the 531 LEGOv2 conversations and 520 LivePerson conversations was carried out by three experienced annotators employed by LivePerson. Annotators were given instructions (see Appendix A) and a small initial annotation job. The results of the initial job were quality assured by LivePerson taxonomy experts, and feedback was given to individual annotators to increase alignment and consistency. After QA and feedback, the annotators were given larger jobs, and those jobs were measured for agreement. Following Schmitt *et al.* (2011), Schmitt and Ultes (2015), we took the median score of our 3 annotators as ground truth. For 3-way ties with ACQIs (6.4% of turns), we chose to use the most common label for that particular dialog system out of the labels the annotators have chosen. So as in Schmitt *et al.* (2011) and Schmitt and Ultes (2015), we use third parties rather than the users of the dialog system to judge the turn-by-turn quality, and like Schmitt *et al.* (2011), Schmitt and Ultes (2015) our annotators are expert. The biggest differences between their annotation work and ours is that the guidelines for the running IQ score are simplified, and we include an additional annotation task for ACQI to recommend an appropriate fix. The averages of the minimum and final IQ scores are shown in Table 3, which shows that the LEGOv2 has the lowest dips in IQ score during a conversation, but by the end of the conversations, the IQ for the various bots is quite similar.

In spite of having similar overall IQ outcomes, the LEGOv2 dialog system has fewer neutral steps, and more positive and negative turns, whereas LivePerson dialog systems have more neutral turns. This is shown in Fig. 4.

We measured inter-annotator agreement to measure the clarity of our ACQI taxonomy and IQ rules when applied to real dialog systems. The results have Cohen's Kappa (CK) of .68, which is substantial agreement according to Landis and Koch (1977). See Table 4 for more details.

6.4. Analysis of combined ACQI and IQ annotations

By looking at the turn-to-turn change in IQ given the presence of a particular ACQI, we gain a more nuanced understanding of how a dialog system is performing. In previous attempts to

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

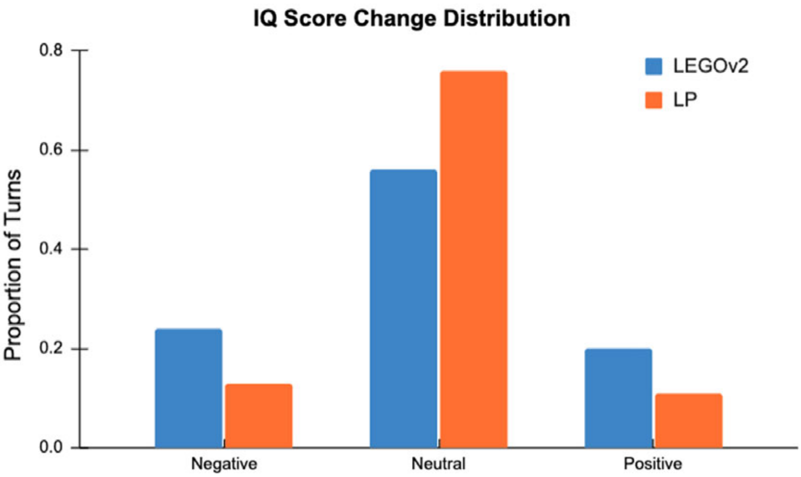


Figure 4. Distribution of negative/neutral/positive score changes grouped by LEGOv2 and LivePerson dialog systems.

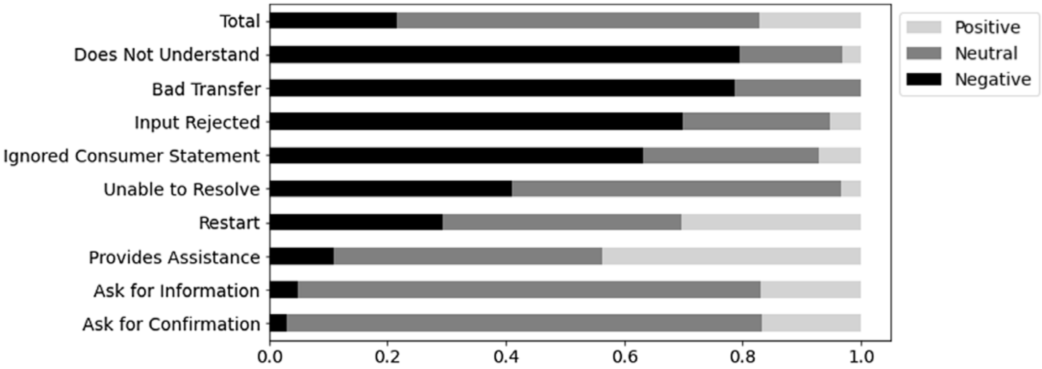


Figure 5. ACQIs from Table 2 along with the proportions of each that were aligned with positive, negative, and neutral changes in IQ. Note that for the above graphic, we excluded any turn whose preceding IQ score was a 1 or 5.

measure dialog quality, a lack of understanding and an inability to resolve a consumer's issue (task completion) are taken to be unequivocally bad. This is not always the case.

From Fig. 5, we see that with the exception of "Bad Transfer" (which was not annotated as positive), all of our ACQIs can be positive, negative, or neutral. "Does Not Understand," "Ignored Consumer Statement," and "Input Rejected" are usually negative, but can be reasonable responses if the consumer is typing things that are out of scope or incomprehensible. However, they are negative when the consumer's utterance should have been understood by the system. "Unable to Resolve" is positive when the system correctly identifies that the consumer's request is out of scope. "Provides Assistance" is positive if the assistance was requested and appropriate, but negative if it was not, or if the assistance has already been provided. Similarly, "Ask for Information" can be positive or negative, depending on the relevance of the requested information.

Analyzing the annotated datasets by ACQI and dialog system is also instructive (see Table 5). For most of the dialog systems, "Does Not Understand" is the largest category of ACQI associated with a decrease in score. The *Junior Sales Assistant* is an exception, and for this, 54% of the ACQIs are marked as "Provides Assistance," indicating that this bot is making too many improper transfers or inappropriate suggestions to the customer. While having the score alone would be somewhat valuable in locating this issue, the ACQI gives additional guidance on what steps can be taken to mitigate the undesirable behavior.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 5. Distribution of ACQIs given a decrease in IQ score

Dialog system	LEGOv2	Junior sales assistant	Help and route	Router	Food expert
Does Not Understand	0.655	0.196	0.631	0.667	0.557
Ignored Consumer Statement	0.093	0.009	0.046	0.000	0.076
Ask for Information	0.078	0.140	0.077	0.000	0.165
Input Rejected	0.057	0.009	0.000	0.000	0.013
Unable to Resolve	0.042	0.000	0.000	0.000	0.051
Ask for Confirmation	0.034	0.000	0.000	0.000	0.089
Provides Assistance	0.024	0.542	0.246	0.333	0.051
Restart	0.017	0.000	0.000	0.000	0.000
Bad Transfer	0.000	0.103	0.000	0.000	0.000

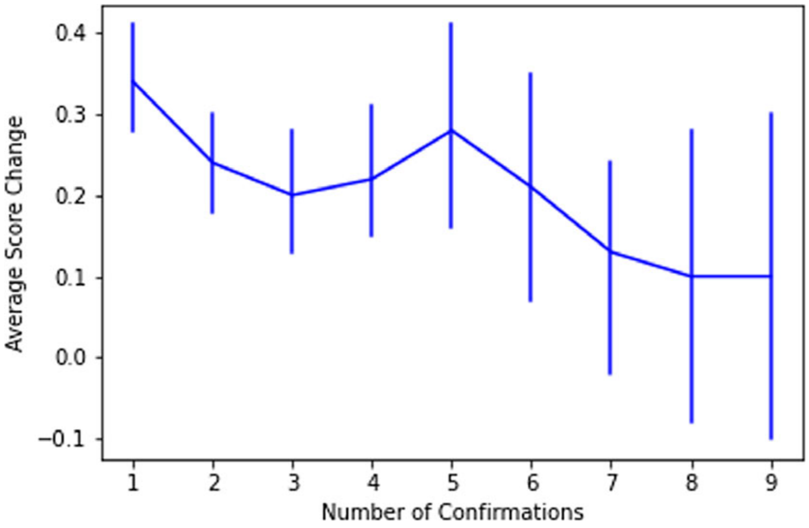


Figure 6. Relationship between number of confirmations and score change.

We can also analyze combinations of ACQIs and their effect on a conversation. For example, Fig. 6 shows the mean score change with 95% error bound for “Ask for Confirmation” ACQI. We observe that asking for confirmation once normally leads to an increase in IQ score, but after this the effectiveness decreases and asking for confirmation more than 6 times can be actively harmful. It is important to realize that the ACQIs combine nonlinearly. That is, a single instance of an ACQI in a conversation may be healthy, but multiple copies of it may be a negative indicator. The extent to which the ACQIs have known meaning is the extent to which structural statistical models can be built, allowing bot creators the ability to test and refine explicit hypotheses about how the conversational events actually aggregate to the dialog system user experience. The careful definition and annotation work described so far enables us to start quantifying such effects in ways that were not hitherto available. Analyzing such combinations of causes and effects in dialog systems will be extended in future work.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

7. Experimental validation of the predictive ability of the ACQI and IQ model

While the annotation and analysis so far were already able to provide some useful insights, the larger goal of these efforts is to build systems that can automatically highlight crucial ACQIs in a dialog system so that they can be fixed.

There is prior work in predicting IQ from labeled conversations, using a variety of methods. The original work of Schmitt *et al.* (2011) used Support Vector Machines, and more recent work has used stateful neural network models including LSTMs (Rach *et al.* 2017) and biLSTMs (Bodigutla *et al.* 2020).

7.1. Vectorized features from conversations

In common with these approaches, we adopted a vector representation for features. However, we deliberately restricted our model to use only features extracted directly from the conversation text and the annotation, so that the method could be more universally applicable, and in particular, to be able to use the same annotation setup and featurization processes for data from the LEGOv2 and LivePerson dialog systems.

For our text features (indicated with “text” in Table 6), we use the pretrained contextual sentence embeddings of Sentence-BERT (Reimers and Gurevych 2019). Our embedding dimension is 768 for each speaker’s response. When predicting ACQI and IQ, we concatenate the embeddings for both consumer and dialog system for the two most recent turns (the current turn and one prior). This results in a $4 \times 768 = 3072$ dimensional vector. When the previous utterance is unavailable, we use a zero vector of the appropriate dimension. This feature vector represents a system where the model uses only surface textual features for the previous two turns per user.

To compare with a system that also uses its observations and predictions of the rest of the conversation so far, we experimented with adding features derived from the ACQI labels. For the features “annotated-acqi” and “predicted-acqi,” we use cumulative counts on the one-hot encoding for the presence of ACQIs. These feature sets were also combined with the text features as “annotated-acqi+text” and “predicted-acqi+text,” for which we concatenated the cumulative counts to the contextual sentence embeddings with our cumulative ACQI counts.

7.2. Model training and prediction

To get a good representation of many ACQIs across the training and test sets, we performed nested cross-validation (following Krstajic *et al.* 2014), which reduces bias when testing different configurations. We implemented nested cross-validation using the *multi-label Scikit Learn* python package and methodology of Szymański and Kajdanowicz (2017), which supports balanced multi-label train/test splits. Our implementation used 5 cross-validation folds for the inner and outer loop, where we store the estimations for each sample in the test set of a fold and computed a final score over all of the folds.

We use the features described in Section 7.1 to train a variety of classifiers: For predicting ACQI we tested logistic regression, random forest, and xgboost and for IQ we tested the above and a linear regressor. We found that the best-performing text-based model for predicting both ACQIs and IQ was logistic regression and reported the results using this model in Sections 7.3 and 7.4. In this model, C (inverse regularization strength) is set to 0.01 and we use the “balanced” class weight setting in Scikit Learn. For prediction IQ, we also trained a BERT classifier (BERT-Base; Devlin *et al.* 2019) using back-propagation to tune the encoder weights. We use Hugginface’s bert-base-uncased, and tuned on subsets of the LEGO and LivePerson data. Our input for the BERT classifier is the concatenated text from the two most recent turns for the consumer and dialog system, separated with special [cust] and [bot] tokens to indicate the speaker.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 6. IQ Model Performance: linear weighted Cohen kappa (LWCK), unweighted average recall (UAR), Spearman rank correlation (ρ) for IQ. Model selection and hyper-parameter selection were accomplished by nested cross-validation (5 folds)

Dialog system	Classifier	ρ	UAR	LWCK
LEGOv2	BERT	0.786	0.603	0.646
	LR:TEXT	0.572	0.585	0.47
	LR:ANNOTATED-ACQI	0.469	0.499	0.404
	LR:PREDICTED-ACQI	0.296	0.396	0.242
	LR:ANNOTATED-ACQI+TEXT	0.622	0.597	0.529
	LR:PREDICTED-ACQI+TEXT	0.593	0.516	0.475
Junior Sales Assistant	BERT	0.614	0.387	0.600
	LR:TEXT	0.721	0.567	0.678
	LR:ANNOTATED-ACQI	0.147	0.429	0.349
	LR:PREDICTED-ACQI	0.529	0.399	0.494
	LR:ANNOTATED-ACQI+TEXT	0.84	0.599	0.797
	LR:PREDICTED-ACQI+TEXT	0.831	0.573	0.798
Help and Route	BERT	0.580	0.289	0.403
	LR:TEXT	0.569	0.503	0.495
	LR:ANNOTATED-ACQI	0.25	0.363	0.285
	LR:PREDICTED-ACQI	0.288	0.384	0.256
	LR:ANNOTATED-ACQI+TEXT	0.673	0.553	0.597
	LR:PREDICTED-ACQI+TEXT	0.682	0.434	0.583
Router	BERT	0.561	0.361	0.415
	LR:TEXT	0.621	0.346	0.585
	LR:ANNOTATED-ACQI	0.608	0.482	0.521
	LR:PREDICTED-ACQI	0.636	0.341	0.592
	LR:ANNOTATED-ACQI+TEXT	0.765	0.635	0.679
	LR:PREDICTED-ACQI+TEXT	0.841	0.374	0.791
Food Expert	BERT	0.552	0.416	0.435
	LR:TEXT	0.698	0.603	0.609
	LR:ANNOTATED-ACQI	0.574	0.469	0.493
	LR:PREDICTED-ACQI	0.514	0.435	0.448
	LR:ANNOTATED-ACQI+TEXT	0.765	0.635	0.679
	LR:PREDICTED-ACQI+TEXT	0.787	0.55	0.691

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 6. Continued

Dialog system	Classifier	ρ	UAR	LWCK
All Bots	BERT	0.618	0.411	0.490
	LR:TEXT	0.595	0.599	0.499
	LR:ANNOTATED-ACQI	0.462	0.511	0.417
	LR:PREDICTED-ACQI	0.326	0.408	0.274
	LR:ANNOTATED-ACQI+TEXT	0.651	0.612	0.561
	LR:PREDICTED-ACQI+TEXT	0.624	0.531	0.514

7.3. Predicting IQ: Results and discussion

For the IQ prediction experiment, results for each of the dialog systems using a BERT model with text-only input and a logistic regression classifier using features derived from text and ACQI scores (as described in Sections 7.1 and 7.2) are presented in Table 6. The results found can also be compared with the annotator agreement findings of Table 4.

Points to highlight include:

- The text-only prediction, which is the easiest to implement, achieves linear weighted Cohen's Kappa performance slightly lower but comparable to annotator agreement (average 0.49 vs. 0.52).
- Using only ACQI information (either annotated or predicted) leads to a loss in recall in all but one dialog system, and on average reduces recall by around 10% using annotated labels and 20% using predicted labels.
- Though the use of annotated labels along with text features leads to improved correlation with annotators (e.g., an average 6% improvement in kappa score), these expectations do not transfer reliably to the more realistic case of using predicted ACQI labels as features.
- The increase in average recall from using annotated labels is small (only 2%), and the use of predicted labels causes average recall to drop by 7%.
- The logistic regression models outperform BERT for all dialog systems except LEGOv2. This suggests that BERT underperforms on this task when training data is limited.
- The best Unweighted Average Recall (UAR) results are between the best result using BiLSTM using traditional cross-validation (0.78) and the best result using dialog-wise cross-validation (0.54) presented by Ultes (2019).

These experiments leave much room for optimization and improvement of various kinds, including trying different text featurizers, and the number of turns and relative weights of messages used in the vector encoding. The important findings are that we can predict the exact IQ score approximately 60% of the time and that the use of vector embeddings derived directly from the message texts is the most reliable practical method tested for building features.

7.4. Predicting ACQI: Results and discussion

For the ACQI prediction experiment, results for each of the dialog systems using only text-derived features and logistic regression as a classifier (as described above) are presented in Table 7.

Points to note include:

- The overall weighted average f1-score is 0.790. We are predicting the correct ACQI nearly 80% of the time overall.

Table 7. ACQI model performance

ACQI	Precision	Recall	F1-score	Support (Num. of Bot Turns)
Ask for Information	0.948	0.896	0.921	4144
Ask for Confirmation	0.777	0.781	0.779	2069
Does Not Understand	0.524	0.495	0.509	1417
Provides Assistance	0.884	0.902	0.893	1010
Ignored Consumer Statement	0.196	0.270	0.227	226
Unable to Resolve	0.599	0.807	0.688	176
Input Rejected	0.236	0.392	0.295	120
Restart	0.484	0.762	0.592	80
Bad Transfer	0.250	0.286	0.267	14
Macro avg	0.544	0.621	0.574	9256
Weighted avg	0.799	0.784	0.790	9256

- The accuracy of the classification depends significantly on the support of the class: common ACQIs are predicted much more accurately than rare ones.
- Because of this, the macro average performance (not weighted by support) is worse, with an f1-score of just 0.574.
- The f1-score for an ACQI can be cross-referenced against Fig. 5 and Table 5 to guide improvement efforts. For example, “Does Not Understand” occurs relatively frequently and with overwhelmingly negative impact on IQ score, but the f1-score for predicting this class is only 0.509. Improving ACQI classification performance for this class would therefore be especially impactful.

Learning curves in Fig. 7 show some of the change in performance with different training set sizes.

- It can be seen from that for both LEGOv2 and LivePerson dialog systems, macro and average performance on a 20% held-out set improves sharply for the first 125 training conversations.
- Given that training on only 125 conversations leads to near-peak performance, it offers the possibility of fast-tuning iterations.

We investigated the generalizability of the IQ and ACQI models using a productionized version of the underlying LivePerson model. These results can be found in Appendix B. We find that the models transfer well across systems within the LivePerson framework; however, the LivePerson model does not generalize well outside of the framework to the LEGOv2 data. Furthermore, the LEGOv2 model does not generalize well to LivePerson data. Further investigation into cross-domain generalizability is an important area for future work, particularly the differences between written and transcribed speech data, though we find it promising that different bots within a single framework can share a model.

7.5. Using ACQIs to simplify bot-tuning

As a final result, we estimate the extent to which predicting the correct ACQI could help bot-builders involved in bot-tuning. Referring back to the ACQI taxonomy in Table 2, without any

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

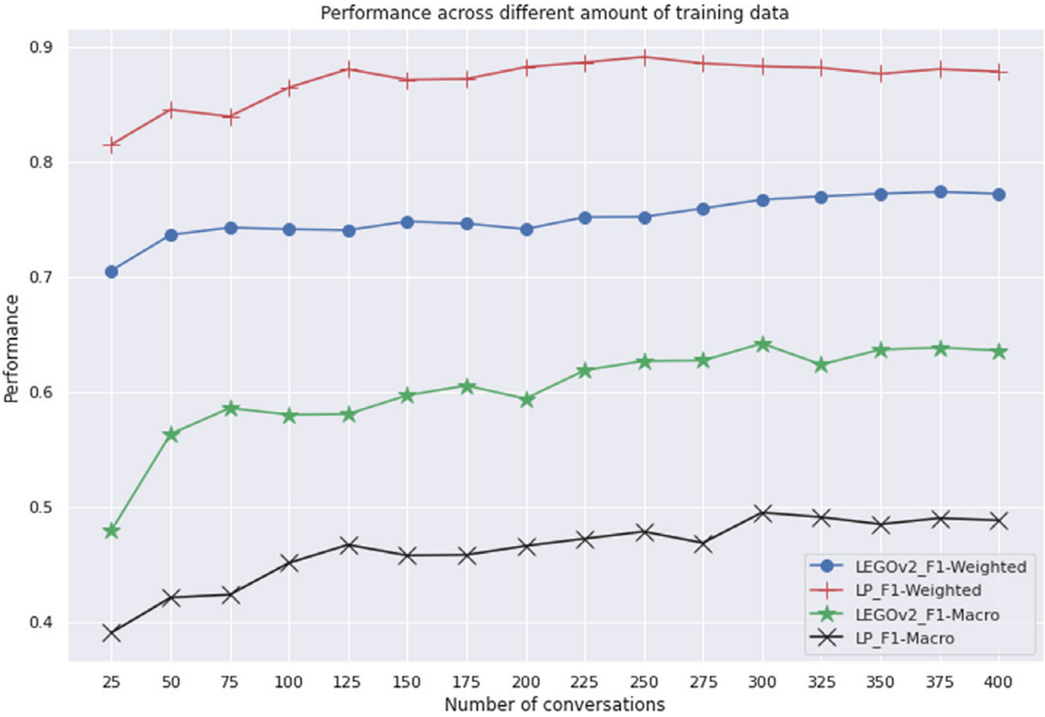


Figure 7. Performance of ACQI models based on number of training conversations.

extra contextual guidance, bot-builders have 28 possible action strategies with LivePerson dialog systems and 31 with LEGOv2 dialog systems. The unaided default case for bot-builders is exhaustive search of those action strategies.

This was compared with the number of options that would be available using the predicted ACQI labeling. For this simple simulation, we made the following assumptions:

1. Each appropriate action is equally likely in the absence of IQ/ACQI.
2. If IQ is available, tuning is only required when the score decreases. If IQ is unavailable then all actions related to the system are relevant.
3. Given the presence of a decremented IQ score, each action is equally likely.
4. If an ACQI is available, all actions that are not assigned to at least one ACQI are included in our list of options that the bot-builder can make.
5. There is always a special action, ⟨`No Action`⟩, that may be applicable for the bot-builder.

The results of this exercise are presented in Table 8. We found that the largest single simplification comes from the use of IQ: if IQ can be modeled accurately, then the average number of recommended options is reduced from 28.6 to 9.73 (about 34%). Adding ACQI classifications as well reduces the average down to 5.4 (about 19% of the original number). This makes a hypothetical but strong case: if IQ and ACQI can be accurately predicted on a turn-by-turn level, the amount of effort it takes a bot-builder to diagnose problems and suggest possible solutions could be reduced by an estimated 81%. While this is an optimistic hypothesis, the potential reward is large enough to encourage more development in this area. Note that even with inaccurate ACQI classification and IQ scores, the search space is fixed so a bot-builder will do no worse than the default exhaustive search they would require without assistance. However, any correct ACQIs and IQs will reduce options for at least a portion of the cases. Although we have not explicitly modeled

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table 8. Average number of recommended actions per dialog system when there is no measurement strategy (None), IQ is available (assuming no actions required when score does not decrement), ACQI alone is available, and IQ + ACQI. 95% confidence intervals were calculated taking 1000 bootstrapped samples (at turn level) per dialog system

Dialog system	Potential actions	IQ	ACQI	IQ + ACQI
LEGOv2	31	7.61 ± .28	7.25 ± .04	**4.2** ± .14
Junior Sales Assistant	28	10.95 ± .99	14.29 ± .11	**5.99** ± .50
Help and Route	28	7.87 ± .83	14.49 ± .07	**4.57** ± .40
Router	28	13.26 ± 1.43	14.77 ± .16	**7.33** ± .80
Food Expert	28	8.98 ± .86	14.39 ± .09	**4.94** ± .47
All DS	28.6	9.73 ± .88	13.03 ± .09	**5.40** ± .46

this, in practice the bot-builders will be looking at aggregate statistics on ACQIs and IQ scores and that aggregation will make inaccuracies in the predictions less important. Even less-than-perfect classification and scores will identify multiple occurrences of a problematic ACQI and the general direction of its IQ score.

8. Conclusion

ACQIs are designed to provide bot-builders with actionable explanations of why their deployed dialog systems fail based on data from user interactions. We have explored the key desirable properties for building an ACQI taxonomy, based on recommendations from the literature, interviews, and collaboration with dialog system experts. Based on an annotated dataset of just over 1000 conversations, we have shown that ACQIs are particularly useful when combined with IQ, in particular so that the decision of whether to take a recommended action can be focused on places in the dialog where quality decreases.

The annotated datasets were used to train predictive models, which achieved a weighted average f1-score of 79% using features based just on vectorized embeddings of recent messages in order and logistic regression for classification. While these results are preliminary, such a classification model could be used to reduce the number of options for a bot-builder to consider by as much as 81%. Results like this should be directly useful to bot-builders for troubleshooting and refining their dialog systems: if the ACQI-based suggestions show up as a tooltip (similar to refactoring tips in software-integrated development environments), they may be useful in the majority of cases, while being easy to ignore in the remainder.

The prioritization of the bot-builder as a key user persona is the driving principle for much of this work. We hope that more research focused on making bot-builders more effective is encouraged and highlighted in the dialog system community, as a crucial route to optimizing the experience of dialog system users overall.

Acknowledgements. The authors would like to thank LivePerson Inc. for financial support while this work was carried out.

Competing interests. All authors were employed at LivePerson, Inc during the main work described in this paper. The authors declare no other competing interest.

References

Adiwardana D., Luong M.-T., So D.R., Hall J., Fiedel N., Thoppilan R., Yang Z., Kulshreshtha A., Nemade G., Lu Y., et al. (2020). Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977.

Asri L.E., Khouzaimi H., Laroche R. and Pietquin O. (2014). *Ordinal regression for interaction quality prediction*. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3221–3225.

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Bodigutla P.K., Polymenakos L. and Matsoukas S. (2019). *Multi-domain conversation quality evaluation via user satisfaction estimation*. In *33rd Conference on Neural Information Processing Systems, NeurIPS 2019*, Vacouver.

Bodigutla P.K., Tiwari A., Vargas J.V., Polymenakos L. and Matsoukas S. (2020). Joint turn and dialogue level user satisfaction estimation on multi-domain conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3897–3909.

Danieli M. and Gerbino E. (1995). *Metrics for evaluating dialogue strategies in a spoken language system*. In *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pp. 34–39.

Deriu J., Rodrigo A., Otegi A., Echegoyen G., Rosset S., Agirre E. and Cieliebak M. (2020). Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review* **54**(1), 1–56.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.

Finch S.E. and Choi J.D. (2020). *Towards unified dialogue system evaluation: A comprehensive analysis of current evaluation protocols*. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, pp.236–245. 1st virtual meeting.

Forrester Research (2020). The total economic impact of IBM Watson assistant. Technical report, Forrester, commissioned by IBM.

Han S. and Anderson C.K. (2020). Customer motivation and response bias in online reviews. *Cornell Hospitality Quarterly* **61**(2), 142–153.

Hirschman L., Dahl D.A., McKay D.P., Norton L.M. and Linebarger M.C. (1990). *Beyond class a: A proposal for automatic evaluation of discourse*. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden*, June 24-27, 1990, Valley, Pennsylvania.

Hirschman L. and Pao C. (1993). *The cost of errors in a spoken language system*. In *Proceedings of the Third European Conference on Speech Communication and Technology*, pp. 1419–1422.

Hockey B.A., Lemon O., Campana E., Hiatt L., Aist G., Hieronymus J., Gruenstein A. and Dowding J. (2003). *Targeted help for spoken dialogue systems*. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

Jain M., Kumar P., Kota R. and Patel S.N. (2018). *Evaluating and informing the design of chatbots*. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 895–906.

Krstajic D., Buturovic L.J., Leahy D.E. and Thomas S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics* **6**(1), 1–15.

Landis J.R. and Koch G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174.

Li J. and Sun X. (2018). *A syntactically constrained bidirectional-asynchronous approach for emotional conversation generation*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 678–683.

Lin Z., Madotto A., Shin J., Xu P. and Fung P. (2019). *MoEL: Mixture of empathetic listeners*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 121–132.

Ling Y., Yao B., Kohli G.S., Pham T. and Guo C. (2020). *IQ-net: A DNN model for estimating interaction-level dialogue quality with conversational agents*. In *Converse@KDD*.

Liu S., Chen H., Ren Z., Feng Y., Liu Q. and Yin D. (2018). *Knowledge diffusion for neural dialogue generation*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1489–1498.

Luo L., Xu J., Lin J., Zeng Q. and Sun X. (2018). *An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 702–707.

Moghe N., Arora S., Banerjee S. and Khapra M.M. (2018). *Towards exploiting background knowledge for building conversation systems*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 2322–2332.

Pieraccini R., Suendermann D., Dayanidhi K. and Liscombe J. (2009). *Are we there yet? research in commercial spoken dialog systems*. In *International Conference on Text, Speech and Dialogue*. Springer, pp. 3–13.

Polifroni J., Hirschman L., Seneff S. and Zue V. (1992). *Experiments in evaluating interactive spoken language systems*. In *Proceedings of the DARPA Speech and NL Workshop*, pp. 28–33.

Qiu L., Li J., Bi W., Zhao D. and Yan R. (2019). *Are training samples correlated? learning to generate dialogue responses with multiple references*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3826–3835.

Rach N., Minker W. and Ultes S. (2017). *Interaction quality estimation using long short-term memories*. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 164–169.

Raux A., Bohus D., Langner B., Black A. and Eskénazi M. (2006). *Doing research on a deployed spoken dialogue system: One year of let's go! experience*. In *INTERSPEECH*

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Raux A., Langner B., Bohus D., Black A. and Eskénazi M. (2005). *Let's go public! taking a spoken dialog system to the real world*. In *INTERSPEECH*.

Reimers N. and Gurevych I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3982–3992.

Rozin P. and Royzman E.B. (2001). Negativity bias, negativity dominance, and contagion. *Personality and Social Psychology Review* **5**(4), 296–320.

Schmitt A., Schatz B. and Minker W. (2011). *Modeling and predicting quality in spoken human-computer interaction*. In *Proceedings of the SIGDIAL. 2011 Conference*, pp. 173–184.

Schmitt A. and Ultes S. (2015). Interaction quality: Assessing the quality of ongoing spoken dialog interaction by experts–and how it relates to user satisfaction. *Speech Communication* **74**, 12–36.

Schmitt A., Ultes S. and Minker W. (2012). *A parameterized and annotated spoken dialog corpus of the CMU let's go bus information system*. In *Language Resources and Evaluation Conference (LREC)*, pp. 3369–3373.

Shriberg E., Wade E. and Price P. (1992). *Human-machine problem solving using spoken language systems (sls): Factors affecting performance and user satisfaction*. In *Proceedings of the DARPA Speech and NL Workshop*, pp. 49–54.

Stoyanchev S., Maiti S. and Bangalore S. (2017). *Predicting interaction quality in customer service dialogs*. In *IWSDS*.

Stoyanchev S., Maiti S. and Bangalore S. (2019). Predicting interaction quality in customer service dialogs. In *Advanced Social Interaction with Agents*. Springer, pp. 149–159.

Szymański P. and Kajdanowicz T. (2017). *A network perspective on stratification of multi-label data*. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. PMLR, pp. 22–35.

Ultes S. (2019). *Improving interaction quality estimation with BiLSTMs and the impact on dialogue policy learning*. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, Stockholm, Sweden. Association for Computational Linguistics, pp. 11–20.

Ultes S., Sánchez M.J.P., Schmitt A. and Minker W. (2015). Analysis of an extended interaction quality corpus. In *Natural Language Dialog Systems and Intelligent Assistants*. Springer, pp. 41–52.

Walker M.A., Litman D.J., Kamm C.A. and Abella A. (1997). *PARADISE: A framework for evaluating spoken dialogue agents*. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain. Association for Computational Linguistics, pp. 271–280.

Wang J., Liu J., Bi W., Liu X., He K., Xu R. and Yang M. (2020). *Improving knowledge-aware dialogue generation via knowledge base question answering*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9169–9176.

Witt S. (2011). *A global experience metric for dialog management in spoken dialog systems*. In *Proceedings of SemDial*, pp. 158–166.

Wu W., Guo Z., Zhou X., Wu H., Zhang X., Lian R. and Wang H. (2019). *Proactive human-machine conversation with explicit conversation goal*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 3794–3804.

Appendix A: Annotation Instructions

Annotation was done by in-house LivePerson annotators using an annotation tool illustrated in Fig. A1 to apply the scores in Table A1 and labels in Table A2. The tool provides "tooltip" on demand by hovering over items. For the annotation used in this work, the tooltip bubble included the ACQI (as "Bot State" and IQ labels (as "Quality"); hovering over those labels accesses the label definition. As described earlier, ACQI and IQ were annotated at the same time in the same pass. Note that "start" in the instructions to the annotators means a state before the conversation begins.

The following are the annotation instructions used by our expert annotators:

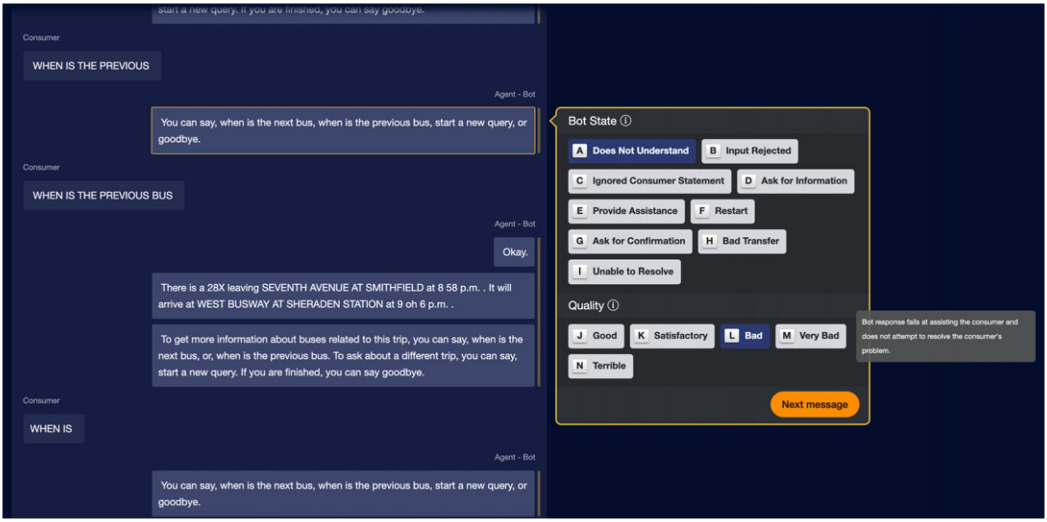
Assumptions for annotations

- *Annotation starts at "ask for information" and "satisfactory."*
- *Increment and decrement as much as necessary depending on how the conversation is going.*
- *The conversation is rated at the turn level and considering the conversation as it goes along.*

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table A1. Quality score annotation guidelines

Quality score	Definition
Good	Bot response is relevant and helpful, and brings the consumer significantly closer to resolution
Satisfactory	Bot response is somewhat helpful to the consumer but not completing the task or solving the problem
Bad	Bot response misses previous consumer response, yet might recover to help the consumer
Very Bad	Bot response fails at assisting the consumer and does not attempt to resolve the consumer's problem
Terrible	Bot response does not assist the consumer and makes another mistake in a series of mistakes. Customers may be so dissatisfied they could end the chat, complain angrily, and/or are likely never again to engage with the brand in chat



start a new query. If you are finished, you can say goodbye.

Consumer

WHEN IS THE PREVIOUS

Agent - Bot

You can say, when is the next bus, when is the previous bus, start a new query, or goodbye.

Consumer

WHEN IS THE PREVIOUS BUS

Agent - Bot

Okay.

There is a 28X leaving SEVENTH AVENUE AT SMITHFIELD at 8 58 p.m. . It will arrive at WEST BUSWAY AT SHERADEN STATION at 9 oh 6 p.m. .

To get more information about buses related to this trip, you can say, when is the next bus, or, when is the previous bus. To ask about a different trip, you can say, start a new query. If you are finished, you can say goodbye.

Consumer

WHEN IS

Agent - Bot

You can say, when is the next bus, when is the previous bus, start a new query, or goodbye.

Bot response fails at assisting the consumer and does not attempt to resolve the consumer's problem.

Figure A1. Annotation tool showing tooltips Bot State.

Please make sure to pick carefully between the Bot States "request confirmation" and "request information." These are easily confused, so make sure to take a moment and consider the difference between them. "Ask for confirmation" is only when the bot is asking to confirm something the customer has already said, while "ask for information" is a request for new information that the customer has not provided. Similarly, "input rejected," "does not understand" and "ignored consumer statement" are very easily confused. Please refer to the definition tool-tips for these 3 Bot States so that you are applying them correctly.

For the Quality Rating, we recommend everyone starting each conversation with "Satisfactory" and going from there. This will help everyone align on the same Quality Rating.

[In addition in training, the annotators were told to start (in their heads, not as an annotation) with Satisfactory as the starting point and increment up or down based on how the bot performs starting with the first turn, and then adjusting per turn as the bot does better or worse]

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table A2. ACQI options and descriptions

ACQI	Definition
Does Not Understand	The bot says that it does not understand the consumer’s response and says something like “I didn’t get that” or “I don’t understand”. This also includes when the bot’s response is incorrect and it is clear that the bot has misunderstood what the consumer said
Input Rejected	The bot does not accept a consumer response to menu options, forms, or other structured content, such as “1, 2, 3, 4,” “a, b, c, d,” or “yes, no”
Ignored Consumer Statement	The bot does not accept a free text consumer response and asks another question
Ask for Information	The bot asks the consumer for information
Resolve Consumer Query	Bot provides useful information or addresses or answers consumer queries. The consumer received an answer and the answer is relevant and actionable
Restart	When explicitly asked for by the consumer and only used when the conversation starts over within the same engagement with the bot. Includes when the bot goes back to the main menu
Ask for Confirmation	Bot asks for user confirmation of input. Includes when the bot asks if the information provided is correct
Bad Transfer	The bot attempts to transfer the consumer to an agent, but either leaves them hanging or abruptly ends the chat. It might also fail to tell them EARLY enough in the conversation that there are no agents available at that hour
Unable to Resolve	The bot explicitly states that it cannot provide the information the consumer requested and does not offer to transfer to an agent

Appendix B: Generalizability

Tables B1 and B2 represent generalization results across multiple LivePerson domains on an internal dataset. We considered four domains in the LivePerson internal dataset. For each domain,

Table B1. Generalization of ACQI models for production dialog systems across data from different industries within the LivePerson framework

Domain	Macro F1-score	Micro F1-score
Consumer	0.85	0.87
Telecommunications	0.79	0.87
Financial Services	0.83	0.84
High Tech	0.69	0.86

Table B2. Generalization of IQ models for production dialog systems across data from different industries within the LivePerson framework: Linear weighted Cohen kappa (LWCK), unweighted average recall (UAR), Spearman rank correlation (ρ)

Domain	LWCK	UAR	ρ
Consumer	0.65	0.69	0.69
Telecommunications	0.63	0.73	0.75
Financial Services	0.64	0.73	0.74
High Tech	0.59	0.64	0.59

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press

Table B3. Generalization of ACQI models between LivePerson and LEGOv2

Model	Train domain	Test domain	Macro F1-score	Micro F1-score
BERT	LEGOv2	LivePerson	0.12	0.25
lr:text	LEGOv2	LivePerson	0.10	0.42
BERT	LivePerson	LEGOv2	0.11	0.47
lr:text	LivePerson	LEGOv2	0.15	0.27

Table B4. Generalization of IQ models between LivePerson and LEGOv2: Linear weighted Cohen kappa (LWCK), unweighted average recall (UAR), Spearman rank correlation (ρ)

Model	Train domain	Test domain	LWCK	UAR	ρ
BERT	LEGOv2	LivePerson	0.05	0.27	0.06
lr:text	LivePerson	LEGOv2	0.04	0.27	−0.01
BERT	LivePerson	LEGOv2	0.08	0.21	0.18
lr:text	LEGOv2	LivePerson	0.16	0.32	0.27

conversations from that domain were held in the test set, and the models were trained on the conversations from the rest of the domains. This process was repeated for all four domains.

The knowledge transfer within the LivePerson domains was quite good. However, we also compared the models trained on LivePerson data and evaluated on LEGOv2 data and vice versa. Tables B3 and B4 show that the knowledge transfer between these two datasets is poor. Further investigation into the reason for this is an important area for future work.

Cite this article: Higgins M, Widdows D, Hockey B A, Hazare A, Howell K, Christian G, Mathi S, Brew C, Maurer A, Bonev G, Dunn M and Bradley J. Actionable conversational quality indicators for improving task-oriented dialog systems. *Natural Language Engineering* https://doi.org/10.1017/S1351324923000372

https://doi.org/10.1017/S1351324923000372 Published online by Cambridge University Press