Automatic Online Evaluation of Intelligent Assistants

Jiepu Jiang[1], Ahmed Hassan Awadallah[2], Rosie Jones[2], Umut Ozertem[2],
Imed Zitouni[2], Ranjitha Gurunath Kulkarni[2] and Omar Zia Khan[2]

[1] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
[2] Microsoft Redmond, WA USA

jpjiang@cs.umass.edu, {hassanam, rosie.jones, umuto, izitouni, raguruna, omarzia.khan}@microsoft.com

ABSTRACT

Voice-activated intelligent assistants, such as Siri, Google Now, and Cortana, are prevalent on mobile devices. However, it is challenging to evaluate them due to the varied and evolving number of tasks supported, e.g., voice command, web search, and chat. Since each task may have its own procedure and a unique form of correct answers, it is expensive to evaluate each task individually. This paper is the first attempt to solve this challenge. We develop consistent and automatic approaches that can evaluate different tasks in voice-activated intelligent assistants. We use implicit feedback from users to predict whether users are satisfied with the intelligent assistant as well as its components, i.e., speech recognition and intent classification. Using this approach, we can potentially evaluate and compare different tasks within and across intelligent assistants according to the predicted user satisfaction rates. Our approach is characterized by an automatic scheme of categorizing user-system interaction into task-independent dialog actions, e.g., the user is commanding, or making a selection, or confirming with the system. We use the action sequence in a session to predict user satisfaction and the quality of speech recognition and intent classification. We also incorporate other features to further improve our approach, including features derived from previous work on web search satisfaction prediction, and those utilizing acoustic characteristics of voice requests. We evaluate our approach using data collected from a user study. Results show our approach can accurately identify satisfactory and unsatisfactory sessions.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *evaluation/methodology, interaction styles, voice I/O*.

Keywords

Voice-activated intelligent assistant, evaluation, user experience, mobile search, spoken dialog system.

1. INTRODUCTION

Intelligent assistants are becoming a prevalent feature on mobile devices. They provide voice control and feedback to mobile device functions (e.g., making phone calls, calendar management, finding places). Users can also search the web or even chat with intelligent assistants. While these novel applications are useful and attractive for users, it is challenging to evaluate and compare them due to the large variability of tasks.

Evaluation is a central component of many related applications, e.g., search engines, Q&A systems, and recommendation systems. These applications are usually evaluated by comparing system-generated answers with "correct" answers labeled by human annotators. For example, in web search, we annotate relevant webpages and evaluate using metrics such as mean average precision (MAP) and normalized discounted cumulative gain (nDCG) [14].

However, intelligent assistants differ from these applications in that they can involve a wide variety of tasks, ranging from making phone calls and managing calendars, to finding places and answers to general questions, and web search. These tasks have different forms of "correct" answers. It is expensive to evaluate each task separately using different human judgment and metrics. It is also difficult to use one single setup to evaluate all tasks. In addition, the tasks performed can be personal in nature and the performance of the system depends heavily on users. These factors make it challenging to conduct manual ground-truth-based evaluation.

To solve these challenges, we adopt approaches similar to recent studies of user satisfaction prediction in web search [1, 3, 6, 7, 17, 32]. These studies developed alternative evaluation approaches by finding and using correlation between explicit ratings of user experience and implicit behavioral signals such as click and dwell time [4, 7]. However, we cannot simply apply user behavior signals in web search to evaluate intelligent assistants due to the wider range of tasks. These tasks may involve different user intents, diverse topics, and distinct user interaction modes. For example, the process of making a phone call, or navigating to a place, involves a dialog style conversation between user and system, with user requests and system responses very different than those in web search. In addition, lots of voice interaction exist in intelligent assistants, and it is important to use voice signal to assess user experience.

We introduce a model for evaluating user experience in voice-activated intelligent assistants. We consider satisfaction as the major indicator of user experience because our study shows that it is consistently correlated with changes in user interests towards the products. Our model predicts whether the user has satisfactory or unsatisfactory experience with an intelligent assistant based on user interaction patterns. Once the model has been trained, it can evaluate real traffic of intelligent assistants without human judgments of correct answers for tasks. This makes it a useful and cheap evaluation approach for companies, who have abundant user traffic and logs. Our model includes a sub-model for automatically classifying user-system interaction into dialog actions, a Markov model over action transitions, as well as features related to requests, responses, clicks, and those using acoustic signals.

Our contributions can be summarized as follows:

- An accurate model for predicting user satisfaction with an intelligent assistant and its components, i.e., speech recognition and intent classification.
- A scheme of categorizing user-system interaction into task-independent dialog actions, and a model to automatically map different actions to this scheme.
- Analysis of user behavior and patterns indicating user experience in intelligent assistants.

The rest of the article introduces our approaches and findings.

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741669

2. RELATED WORK

There are a number of areas of related work relevant to the research described in this paper. These include (1) methods and metrics for the evaluation of search systems, (2) inferring satisfaction from observed search behavior and (3) dialog act modeling and classification in conversational speech. We cover these in turn in this section.

User behavior modeling has been used extensively for evaluating search systems [1, 4, 6, 7]. Traditionally search system have been evaluated using retrieval metrics such as MAP and nDCG [14], where a collection of documents, queries and human labeled relevance judgments are used to evaluate search system performance. These metrics are expensive to collect and potentially noisy, given that third-party judges have limited knowledge of the individual user's intent. Additionally, these metrics are query-based. Previous research has shown that search tasks often contain multiple queries related to the same information need [7]. Unfortunately the connections between these queries are ignored by these metrics. Session-based DCG (sDCG) [15] does consider the session-context, but still requires manual relevance judgments.

Another line of research has focused on using implicit feedback from user behavior to evaluate search engine. These methods have lower cost, are more scalable, and sourced from the actual users.

Early research on implicit feedback [4] used an instrumented browser to determine if there was an association between explicit ratings of user satisfaction and implicit measures of user interest and identified the measures that were strongly associated with user satisfaction. Huffman and Hochster [13] found a strong correlation with session satisfaction using a linear model encompassing the relevance of the first three results returned for the first query in a search task, whether the information need was navigational, and the number of events in the session. Hassan et al. [7] developed models of user behavior to accurately estimate search success using action sequences of user behavior and showed that this yields better performance compared to models derived from the query-URL relevance of top-ranked results for the first query in a task. Follow-up studies showed that satisfaction ratings can be collected in-situ from users [9] and that action sequence models can be learned in a semi-supervised manner from both labeled and unlabeled data [6]. Ageev and colleagues [1] augmented this approach with additional search features. They also used a game-like strategy for collecting labeled data by asking participants to find answers to questions using web search. All these methods focus on analyzing user behavior when interacting with traditional search systems.

In this work, we extend this line of work by presenting the first study, to the best of our knowledge, of user behavior patterns when interacting with intelligent assistants. We study the action sequences performed by the users and jointly model them with the actions performed by the system to predict user satisfaction. We also study features specifically related to voice input and propose methods to analyze the root cause of dissatisfaction.

Perhaps the biggest difference between traditional web search and intelligent assistants is their conversational nature. In many scenarios, intelligent assistants can refer to the previous requests to understand the user better; e.g. "show me weather in mountain view" followed by "how about in palo alto", or "Italian restaurants nearby" and "which ones are 4-stars or above". Therefore spoken dialog systems research is closely related to intelligent assistants. Spoken dialog systems interpret and respond to spoken commands by implementing dialog strategies [2], and the field has seen steady progress over the past two decades [29]. Since they use speech as the primary (or only) form of user communication, they provide error correction mechanisms to account for the potential errors in the automatic speech recognizer (ASR) output. Recently, partially observable Markov decision Processes (POMDP) has established itself as a solid foundation for managing dialogues, and a comprehensive review can be found in [33].

Since they also support other forms of interactions, intelligent assistants differ from traditional spoken dialog systems. In addition to voice system response, intelligent assistants provide answers or options in the display, and users can type in the requests and select a displayed result or option. In this sense, intelligent assistants are related to multi-modal conversational systems [11, 20, 30].

Note that many different taxonomies of dialog acts have been proposed [28]. We do not intend here to propose a new one, but rather to model user and system interaction with the goal of predicting user satisfaction. Our model of system interaction and user is designed independently from any dialog model the system uses. Hence, it is independent of any specific implementation. In contrast with work on learning dialog model transitions [27] we do not attempt to model the most likely dialog sequence, but to use the dialog sequence to model user satisfaction. Our work differs from previous work in offline evaluation of dialog systems [31], as we do not require manual transcription of speech, and thus once trained, our models can be run online, at-scale, evaluating voice assistants in an automated, unsupervised fashion.

3. INTELLIGENT ASSISTANTS

Intelligent assistants are emerging and evolving applications lacking a precise definition. Related products are usually referred to as "intelligent personal assistants", "mobile assistant", "virtual personal assistant", "voice assistant" etc. Also it is unclear to what ends and how frequently people use them. To clarify the goal of evaluation, we need to first study their functionalities and the scenarios they support. This section studies these questions. Due to too many related applications, we restrict our scope as follows:

1. We only consider intelligent assistants on mobile devices.

2. We do not consider proactive suggestions based on personal information, such as displaying flight status automatically if the user received an email about it, or showing traffic and time-to-leave reminders based on daily commute patterns. Evaluating this functionality is outside our scope since it requires long term studies to collect personal information from users' emails, GPS signal, etc.

3.1 Functionality

We picked five representative intelligent assistant products, including Apple Siri, Google Now, Microsoft Cortana, Samsung S Voice, and Nuance Dragon. After extensive usage of these applications, we summarize three major functionalities of intelligent assistants:

1. Device+dialog. This includes using voice commands to access device functions and other tasks using dialog style interaction. For example, users can say "call James" to make a phone call, or ask "do I have any meetings tomorrow?" to check calendars directly. Rich information dialogs, such as "weather in Florence" to check the weather, are also included here. All the five products provide this functionality, but the specific supported features may differ.

2. Web search. All the five products support web search using voice input. Apart from Siri, the other four support a combination of voice and text input. Also, it is usually the last resort to handle user requests: if the intelligent assistant fails to understand a request, it will handle it as a query for web search.

3. Chat. Users can talk to intelligent assistants for fun. Many intelligent assistants have pre-defined interesting responses for popular user questions. For example, if the user asks Cortana "who is your father", the response could be "technically speaking, that'd be Bill Gates. No big deal." All products apart from Google Now support chat.

Another angle of comparison is the types of input they support. All five products support voice input, and all except Siri support text input.

Table 1. Top 5 requests (speech recognition results) and proportion of each domain in a sample of Cortana user logs in April, 2014. Web search and other requests takes 30.7% and 0.6% of the data (do not show examples of requests here due to limited space).

Chat (21.4%)	Device Control (13.3%)	Communication (12.3%)	Location (9.2%)	Calendar (8.7%)	Weather (3.8%)
tell me a joke	play music	call	where am I	set alarm	in Celsius
do you like clippy	play	call mom	find a library	show my alarms	do I need a coat
hello	open facebook	call my wife	I'm hungry	wake me up	what's the weather
sing me a song	open whatsapp	text	where I am	wake me up in twenty minutes	what's the weather like
what's your name	stop music	call my mom	take me home	remind me	what's the weather today

3.2 Usage Analysis

We further study the scenarios of using intelligent assistants. We limit our scope to Cortana due to data access. We randomly sampled 70K sessions from Cortana's user logs during April, 2014. Note that during this time-range, Cortana was demonstrated for the first time, was not yet commercially available, and early-adopter developers were trying out Cortana to explore the functionality. Here a session refers to a sequence of requests from the same user, in which the interval of two adjacent requests does not exceed 30 minutes. This setting is similar to many previous web search studies [10, 23]. We annotated the underlying tasks of the user requests and summarized them into several domains. Web search and chat are two open domains. We also categorize requests to device+dialog functions into the following topics:

- Device control, e.g. launch apps, and play music.
- Communication, e.g. make phone calls, send text message.
- Location, i.e., find or navigate to certain places.
- Calendar, e.g. check calendar, create reminder, and set alarm.
- Weather, i.e., check weather conditions.
- Other: all other supported requests, e.g. taking notes.

Table 1 shows the five most frequent requests and proportion of requests for each domain in the sampled Cortana log (we do not show examples of web search and other requests due to limited space). About half of the requests (47.9%) are accessing device+dialog functions. Web search and chat take 30.7% and 21.4% respectively. In the following discussion, we also refer to them as "device+dialog function tasks", "web search tasks", and "chat tasks".

Note that the domains of requests and their popularity largely depend on the features supported by specific intelligent assistants and the way they are implemented. Therefore, the statistics in Table 1 may not be generalized to other intelligent assistants. Besides, they are also not necessarily representative of Cortana's requests today because the log is sampled from the very early period when Cortana was first put to public test. However, this is the best we can access at the time of the study. Despite the limitations, these domains and topics are still representative because all the five products support them (except that Google Now does not support chat). Therefore, our approach should cover these tasks and domains.

3.3 Goal of Evaluation

Based on the analysis in this section, we come to the following goals of evaluating intelligent assistants.

1. The evaluation approach should be able to work well on the three major tasks as well as the five popular domains.

2. We should evaluate not only the intelligent assistant as a whole, but also its important components separately. Intelligent assistants need to first recognize user requests (may include voice input) and then classify the intent (e.g., identify task and context information). We consider two components in this paper, i.e., automatic speech recognition (ASR) and intent classification.

3. The evaluation measure should be generic and task-independent. This is important because, unlike many applications, it is difficult and expensive to collect ground truth data to evaluate intelligent assistants, since they do not have a consistent form of correct answers for different tasks. For example, for "call James", the correct answer is to understand the intent and the correct person, but for "remind me for a meeting tomorrow at 3pm", the correct answer is the intent and the event information (e.g., theme and time). Without a task-independent evaluation measure, we would need to collect ground truth data for each scenario, which is expensive.

Table 2. Examples of task descriptions.

Type	Description
Device-Dialog Function	You are stuck in very heavy traffic and it seems you will be late for your meeting with James. Use Cortana to send James a text message explaining the situation (James is a contact stored in your phone).
Web Search	Check the exchange rate between US dollars and Australian dollars.
Chat	Talk to Cortana as if she is a real person. Try to make a conversation with her for about 2 minutes.

To make the approach task-independent, we evaluate intelligent assistants by solving the following classification problem:

Given user interaction information of a session, can we identify whether the user is satisfied or not with the intelligent assistant (or its speech recognition and intent classification)?

Using this approach, we can evaluate and compare intelligent assistants by the predicted percentage of satisfactory sessions. This makes the evaluation measure task-independent. As Section 6 will show, user satisfaction is a user experience measure with consistent correlation with changes in user interests towards the product. We conduct a user study to collect user behavior and satisfaction ratings in intelligent assistant tasks, which will be introduced in Section 4.

4. USER STUDY

This section introduces a user experiment to collect user behavior and ratings in different intelligent assistant tasks.

4.1 Participants

We recruited 60 participants through emails sent to a mailing list of an IT company located in the United States. All participants were college or graduate students interning at the company. Their average age was 22.97 (SD=3.45). Among these participants, 35% were female and 58.3% were native English speakers. We asked the participants to rate their spoken English proficiency on a 5-point Likert scale (5 means the highest). For those who are not English native speakers, the average rating was 4.24 (SD=0.66). They were reimbursed $10 gift card for participating in a half-hour experiment.

4.2 Tasks

Each participant finished 10 tasks, including: 4 web search tasks, 1 chat task, and 5 device+dialog function tasks (1 for each of the five domains). This makes the distribution of tasks and domains similar to those shown in Table 1. In total we developed 12 web search tasks, 15 device+dialog function tasks (3 for each domain), and 1 chat task. We rotated the assignment of web search tasks and device+dialog function tasks such that 20 participants worked on each unique task. The chat task is usually open-ended, so we assigned the same chat task (presented in Table 2) to all participants. We also rotated task sequence using a Latin Square. Table 2 shows examples of task descriptions. Detailed descriptions of tasks can be accessed at www.cs.umass.edu/~jpjiang/cortana/.

Table 3. An example of user requests, system responses, and annotated action types in a satisfactory (SAT) session.

#	User Request	System Response	User Action	System Action
1	"Cortana call James"	"Sure, call James Smith mobile, is that right?"	Command	Confirm
2	"Yes"	"Calling James Smith, mobile." [call the contact]	Yes	Execute

Table 4. An example of user requests, system responses, and annotated action types in a dissatisfactory (DSAT) session.

#	User Request	System Response	User Action	System Action
1	"Where is the nearest pharmacy"	"Here are 8 pharmacies near you." [show options on the screen]	Command	Option
2	"Send me the directions to block sponsee" (Show me the directions to Clark's pharmacy)	"Sorry, I couldn't find anything for 'Send me the directions to block sponsee.' Do you wanna search the web for it?"	Select	Confirm
3	"No"	"Here are 8 pharmacies near you." [show options on the screen]	No	Option
4	"Directions to Clark's pharmacy"	"OK, getting you directions to Clark's Pharmacy." [navigation]	Select	Execute

4.3 Device and Environment

Participants worked on the tasks using a Windows phone with the latest version of Windows Phone 8.1 and Cortana installed. If the task needed to access some device resources or functions, they were installed to make sure users would not encounter problems. The experiment was conducted in a quiet room, so that we can reduce the disturbance of environment noise. Although real application environment often involves noise and interruption, we eliminated those factors to simplify the experiment.

4.4 Procedure

The participants were first asked to watch a video introducing Cortana (about 3 minutes long) and to complete a background survey. Then, they worked on a training task and 10 formal tasks. We instructed them that they could stop a task when they had accomplished the goal or if they became frustrated and wanted to give up. Finally, they answered a feedback survey and a short interview. The total experiment time was about 30 minutes.

For each task, we first verbally described the task scenario to the participants. The participants were not shown the task description while they are working on the task, because in an earlier pilot study, many participants directly read the sentences shown in task descriptions as requests. To encourage them to use free form commands and queries, we switched to verbal descriptions. When the participants worked on the task, they were allowed to issue both voice and text requests, reformulate requests, and interact with results (e.g., tapping a result to read more details). After terminating a task, they answered questions regarding their experience in this task session. In this paper, we mainly used their answers to the following three questions:

- How satisfied are you with your experience in this task?
- How well did Cortana recognize what you said?
- How well did Cortana understand your intent?

Responses to the three questions are referred to as ratings of user satisfaction, ASR quality, and intent classification quality. We did not specifically instruct participants the definition of intent classification, and the ratings are purely based on user's own understanding. Our goal is to predict these ratings in order to evaluate intelligent assistants. Participants answered these questions using a 5-point Likert scale (5 is the best and 1 the worst). In addition, we collected user ratings of frustration, goal success, and effort in each task. We also asked participants to report their interests in using Cortana twice: before they started with any task, and after they finished all the tasks. This helps us understand the relationship between user experience in individual tasks and changes in user interests towards an intelligent assistant product over time.

5. METHOD

This section introduces our approach. We assume the existence of a classifier that can accurately classify user sessions into three task types, i.e., device+dialog function, web search, and chat tasks. This assumption is reasonable because such classifiers have been readily implemented in most intelligent assistant products (such that they can handle requests for different types of tasks). We do not discuss how to implement such a classifier in this paper, because we focus on the evaluation models for different tasks. We train separate evaluation models for each task due to large task variability. When evaluating a session, we first classify its task type and then adopt the task's evaluation model to predict user satisfaction and the quality of ASR and intent classification.

This section introduces approaches and features for evaluating intelligent assistants. First, we introduce a way of characterizing user interaction in device+dialog function tasks. We classify requests and responses into action types and use action sequence to predict user satisfaction. Then, we introduce generic features for all tasks. We put more focus on device+dialog function tasks because few previous work addressed the challenge.

5.1 User and System Action

In device+dialog function tasks, users interact with intelligent assistant in a way similar to spoken dialog systems. A task session includes one to many rounds of user-system interactions. In each round the user issues a request and the system gives a response.

Table 3 shows an example of a satisfactory (SAT) device+dialog function task session. The user completed the task without any obstacles. In contrast, the system makes mistakes in speech recognition and/or intent classification in dissatisfactory (DSAT) sessions. It requires extra user interaction to correct the mistakes. Table 4 shows an example. The system did not recognize the user request in the second round and gave a useless suggestion for web search. It costs the user two more rounds to correct the error.

We categorize requests and responses into different types. We refer to these types as action types. Table 3 and Table 4 also show annotated action types. These action types are high level characterization of requests and responses ignoring the detailed contents. For example, we annotate both "Cortana call James" and "Where is the nearest pharmacy" as "Command", i.e., commanding the intelligent assistant to execute certain operations, despite the details of the operation are different.

These actions may indicate the status of the session. For example, when the system asks the user to confirm an operation (e.g. to call someone), a "Yes" action conveys the user's positive attitude to the system's response in the previous round and is indicative of SAT sessions, and vice versa. We show more analysis in Section 7.

Following previous work on dialog acts modeling [27], we define the following system (intelligent assistant) actions:

- **Execute**: executes an operation in this round.
- **Confirm**: asks the user whether or not to execute an operation.
- **Question**: asks the user a question for specific information.
- **Option**: provides a list of options and wait for user selection.
- **WebSearch**: searches the web using request content.
- **Error**: reports system error to the user, e.g., cannot understand the request, cannot find an answer, network error, etc.
- **NoAction**: does nothing and returns to the default interface. In Cortana, it happens when user declines to execute an operation.

And the following user actions:

- **Command**: commands the system to perform an operation.
- **Yes/No**: agrees or declines the system's confirmation.
- **Answer**: answers the system's questions.
- **Select**: selects an option provided by the system.

Here we define system actions based on the actual operation of the intelligent assistants in each round. For a specific product, one can simply define a rule to map operations to action types. In contrast, we only know limited content of user request. We need a classifier to identify user action types. For this purpose, we annotate the collected data and train a classifier for user action types using the following features:

- **QLength**: the number of words in the user request (ASR result). QLength is useful because we notice that "Yes/No" and "Select" are usually short, while "Command" and "Answer" are longer.
- **HasWordX**: whether the request (ASR result) includes a specific word X. We handcraft a list of "Yes" and "No" words for Cortana. The "Yes" words include: yes, yep, right, yeah, send, and call. The "No" words include: no, nope, and cancel.
- **PercWordX**: the percentage of a specific word X in the request. We use the "Yes" and "No" word list in PercWordX features.
- **IsPrevActionX and IsNextActionX**: whether the previous/next system action is X. This feature is important because user action is usually triggered by the previous system action, or triggers the next system action.

Note that different intelligent assistants may implement these actions differently. For example, they can notify the user a list of options by voice, or displaying on the screen, or both. When asking for user's confirmation of an operation, some assistants consider the user as agreeing with the operation if the user does not respond after a few seconds. Similarly, user behavior for each action may be distinct in different system. Therefore, it requires different rules and classifiers to predict action types in different systems. Whereas we believe these action types are generalizable to other products.

5.2 Modeling Action Patterns

We infer the quality of a session by modeling action patterns in the session. We assume that SAT and DSAT sessions (or sessions with SAT or DSAT ASR/intent classification quality) have different action patterns, e.g., "No" may be more common in DSAT sessions.

We represent a session S as a series of user and system actions, i.e., $S\{u_1, s_1, \ldots, u_n, s_n\}$, where u_i and s_i are the user and system action in the ith round. We model action pattern using a 3-gram model, i.e., each action depends on previous two actions. We also add START and END to the beginning and the end of a session.

Let L be a target session quality label (e.g., SAT or DSAT). The evaluation problem can be solved by inferring the most likely label L for an observed action sequence S, as shown in Equation (1). We use θ_L for the 3-gram action model of sessions with label L. Then, we can calculate $P(S|L)$ as Equation(2), where u_0 and s_0 are START and u_{n+1} and s_{n+1} are END. Let (s, u, v) be three successive actions, we estimate $P(v|s, u)$ from the dataset as Equation(3), where: $c(s, u, v)$ is the raw count of (s, u, v); $P(v|u)$ and $P(v)$ are the bigram and unigram probabilities of v; α and β are smoothing parameters.

$$P(L\mid S)\propto P(S\mid L)P(L) \tag{1}$$

$$P(S\mid L)=\prod_{i=1}^{n+1}P(u_i\mid s_{i-1},u_{i-1};\theta_L)\cdot\prod_{i=1}^{n+1}P(s_i\mid u_i,s_{i-1};\theta_L) \tag{2}$$

$$\hat{P}(v\mid s,u)=\frac{c(s,u,v)+\alpha\cdot P_{ML}(v\mid u)+\beta\cdot P_{ML}(v)}{\sum_{v_i\in U}c(s,u,v_i)+\alpha+\beta} \tag{3}$$

The action sequence model itself can evaluate a session. We can use it with or without the prior probability factor $P(L)$. We found that using $P(L)$ can lead to better accuracy while dropping $P(L)$ can result in better F-measure. To combine with other features, we use the log probability of a session's action sequence as features, i.e., $\log P(S|\mathrm{SAT})$, $\log P(S|\mathrm{DSAT})$, $\log P(\mathrm{SAT}|S)$, and $\log P(\mathrm{DSAT}|S)$.

5.3 Generic Features

In addition to action sequence features, we introduce other features in this section. Most of the features are applicable to all three tasks, with only a few exceptions. Some of these features (e.g., click features) come from previous studies of user satisfaction and success in web search. Besides, we also adopt acoustic features due to the prevalence of voice interaction in intelligent assistants.

5.3.1 Click Features

In web search, click and click dwell time are useful indicators of document relevance [21, 22] as well as query and session level search success [1, 6, 7, 17, 32]. Here "click" means tapping a result item. The item can be a search result in web search tasks or in other types in device+dialog function tasks (e.g., a best answer from a list of candidates). Click features are not applicable to chat tasks. We include the following click features:

#click	Number of clicks per session and per request.
#click longer or shorter than *t*	Number of clicks by dwell time. We count clicks longer than 30s and those shorter than 15s.
rank_click	Average rank of clicks.
click_dwelltime	Average click dwell time.
time_firstclick	Average time to the first click of a request.

5.3.2 Request Features

Previous studies found some query characteristics are useful for predicting search success or difficulty. We use similar request features here. We first consider request type in a session. Shokouhi et al. [26] found that switching from voice to text queries correlates with low search quality in mobile search. Studies also found increased usage of query suggestions in difficult topics [24] and complex tasks [18]. Therefore, we use the number of requests by type (voice, text, suggestion) and voice-to-text switch as features.

Request content may also indicate user experience. For example: Hassan et al. found that high similarity of queries is an indicator of unsuccessful task [8, 10]; Jiang et al. [19] found that user may repeat a voice query if ASR error happens, and long queries are more likely to have ASR errors. We use request length and the similarity of adjacent requests (by ASR results) as features.

Another information we make use of is request dwell time. It is calculated as the time interval of adjacent requests. We consider average request dwell time as well as the number of requests with dwell time longer or shorter than certain thresholds.

#request	Total number of requests in a session.
#request_type	Number of requests of a specific type.
#request longer or shorter than *t*	Number of requests whose dwell time is longer or shorter than *t* (set to 30s, 50s, or 100s).
request_length	Average request length (by word).
request_dwell	Average request dwell time.
#voice2text	Number of transition from voice to text request.
edit_distance	Average edit distance of adjacent requests.
#common_word	Average number of common words in adjacent requests.
#common_req	Number of adjacent requests with the same ASR output.

5.3.3 Response Features

We consider responses of a specific type as features. In device+dialog function tasks and chat tasks, we use the number and percentage of voice responses. In web search tasks, we use the number and percentage of responses the intelligent assistant can provide instant answers as features. This is because in some tasks, user may prefer certain types of responses. For example, users may be frustrated if the intelligent assistant cannot respond them in voice in device+dialog tasks and chat tasks. Similarly, users may prefer responses of instant answers when searching for factual questions.

We also incorporate features considering the content of response. We use the Jaccard similarity of results between adjacent responses as a feature. This helps us measure request similarity at result level. In addition, we consider the time to achieve the first execute action in the session in device+dialog function tasks. Intuitively, if it requires many rounds of interaction to command an intelligent assistant, the task session is unlikely satisfactory.

instant_answer/ voice_response	Number and percentage of responses provide instant answers/voice responses to the user.
jaccard_result	Jaccard similarity of results or items.
first_execute	Time (number of rounds) to the first "execute".

5.3.4 Acoustic Features

Since intelligent assistant involves lots of voice interaction, we also adopt acoustic features as a component of our approach. When encountering ASR errors, user may slow down the speed of speaking in requests [16, 19]. To detect such behavior, we compare the speaking rate in adjacent voice requests. The speaking rate is measured as the number of words (in ASR output) divided by the duration of the request. We measure the number and percentage of requests with speaking rate below a certain ratio as compared to the previous request. We set the ratio to 0.25, 0.5, and 0.75.

The second acoustic feature is the similarity of metaphone code between adjacent requests. Metaphone [25] is a way of indexing words by their pronunciation and it can help us detect ASR errors and enhance the accuracy of request similarity measures. For example, a request "WhatsApp" may be incorrectly recognized as "what's up", but their Metaphone codes are both "WTSP". In such cases, this phonetic similarity feature helps us detect repeated or similar requests that are missed by normal request similarity features (based on ASR outputs).

Besides, we adopt Huang et al.'s method [12] to measure ASR confidence and use the confidence of the voice requests as a feature. In short, ASR confidence gets higher when both acoustic and language model scores of the selected recognition hypothesis are significantly higher than the remaining hypotheses.

asr_confidence	Voice request speech recognition confidence score.
metaphone_sim	Metaphone code similarity between adjacent requests.
slow_request	The number and percentage of requests with slower speaking rates comparing to previous requests.

6. DATA

We collected information of 60 users in 600 sessions. We use the dataset to test how well our approach can correctly evaluate an intelligent assistant and its components in different types of tasks. Table 5 shows some data statistics. We consider the evaluation as binary classification and divide the sessions into binary classes by user ratings. Due to the large difference of rating distributions, we set the thresholds differently for each task type and rating type such that we can balance the proportion of positive and negative classes.

We compute the correlation of user satisfaction with the quality of ASR and intent classification. Results in Table 5 suggest that the quality of the two components do affect overall user satisfaction. Comparing two components, satisfaction relies more on the quality of intent classification but less on ASR quality. The degree of correlation also differs a lot in three tasks. We observed a moderate correlation (r=0.54) in device function tasks, but a weaker one in web search tasks (r=0.37), and an insignificant correlation in chat tasks ($p \geq 0.05$). This indicates different types of tasks rely differently on the two components. This also confirms our objective that it is necessary to evaluate not only the overall experience, but also each components in order to diagnose the system in deeper details.

In this paper, we use user satisfaction as measure for the overall quality of a session. There are other user-based measures, such as goal success [6, 7] and user frustration [3]. We do not know much about their differences and validity. Table 5 shows the correlation of satisfaction with frustration, goal success, and effort in three task types. Satisfaction is positively correlated with goal success, and negatively correlated with frustration and effort. The correlations are rather strong, showing these measures are related to each other.

Table 5. Some statistics of the collected user study data.

Statistics / Task Type	Device Function	Web Search	Chat
# Sessions	300	240	60
Average # requests in a session	2.44	2.59	8.95
Satisfaction vs. ASR Quality	0.54	0.37	-
Satisfaction vs. Intent Quality	0.77	0.76	0.70
ASR Quality vs. Intent Quality	0.49	0.35	-
Satisfaction vs. Frustration	-0.63	-0.75	-0.76
Satisfaction vs. Goal Success	0.69	0.67	0.70
Satisfaction vs. Effort	-0.67	-0.71	-0.63

"-" means the correlation is statistically insignificant ($p \geq 0.05$)
all other correlations are significant ($p<0.05$)

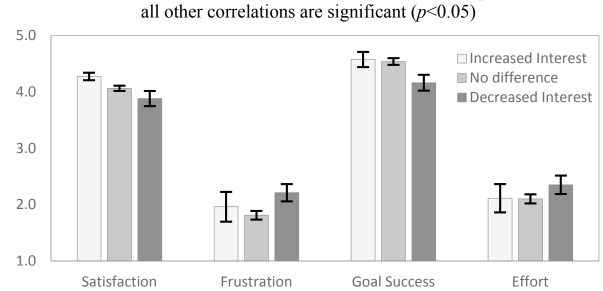

Figure 1. Average task-level ratings for users with increased, equal, or decreased interests after the experiments (10 tasks).

We further examine how task-level user experience relate to the changes of user interests in using the product. We measured user interests in using Cortana twice: at the beginning and the end of the user study. We divide the users into groups with increased, equal, or decreased interests. Figure 1 shows that three groups have significantly different task-level satisfaction ($p<0.05$). But we only observed significant differences in frustration, goal success, and effort between users with decreased interests and others. This indicates that satisfaction is a task-level measure consistently correlate to the changes of long term user adoption, but other three measures may only correlate with decreased user interests. This confirms that we should adopt user satisfaction as the measure for session quality.

7. DEVICE+DIALOG FUNCTION TASKS

This section evaluates our model in device+dialog function tasks. We first evaluate the action sequence model described in Section 5.2. Then, we evaluate other features. In all experiments, we generate 10 random divisions of the data and perform 10-fold cross validation on each division. This generates results for 100 folds in total. We report the mean value of F-measure and accuracy in the 100 folds. We test for significant difference using Welch t-test.

7.1 Action Type Classification

We first evaluate how well we can predict user action type. We manually labeled all the user actions to train a classifier and evaluate its accuracy. We use multi-class logistic regression for classification. Table 6 reports the effectiveness of the classifier. It shows that we can achieve satisfactory performance in predicting user action types. The average F-measure is as high as about 0.9. Comparatively it is less effective in identifying the "No" and "Select" actions (but it still has over 0.8 F-measure). This is probably due to there is limited data (only 2.8%) to train models for "No" and we did not make a word list for "Select". This shows that, provided

Table 6. Effectiveness of user action type classification.

Action Type	Proportion	F_1	Precision	Recall
Command	67.6%	0.956	0.973	0.940
Yes	6.7%	0.956	0.984	0.929
No	2.8%	0.815	0.867	0.768
Select	13.9%	0.849	0.793	0.913
Answer	9.0%	0.910	0.870	0.953
Micro-averaged	-	0.932	0.932	0.932
Macro-averaged	-	0.897	0.897	0.901

certain amount of training data, we can automatically identify action types defined in Section 5 accurately. If not specified, we use predicted user action types in following experiments.

7.2 Action Sequence Model

Table 7 shows results for action sequence model without using the prior probability of classes with both the human annotated and the predicted action types. The results show that using predicted actions can be as effective as using human annotated actions in all three prediction tasks. We did not observe significant difference of the two in predicting user satisfaction and the intent classification quality. The action sequence model using the predicted action types has only slightly worse average F-measure comparing to that using human annotated action types when predicting ASR quality (0.699 vs. 0.713, p<0.05). Again, this shows that the prediction of user action type is accurate and does not affect much on the performance of the action sequence model.

Due to few baseline approaches for evaluating device-dialog function tasks, we do not directly compare action sequence model to other approaches. But according to Wang et al.'s studies [32], state-of-the-art approaches predict user satisfaction in web search with the average F-measures ranging from 0.7 to 0.8. The action sequence model achieves similar performance in evaluating device function tasks of intelligent assistants. The model is relatively more effective in prediction of user satisfaction and intent classification quality (with 0.79 and 0.76 average F-measures), but worse in evaluating ASR quality (but still can achieve about 0.7 average F-measure). It is probably because many action patterns indicating SAT and DSAT sessions are not equally predictive of ASR quality. We will show examples in Section 7.1.3. Using the prior probability of classes leads to 2%-3% improvement in accuracy, but about 2% decline in F-measure. We do not further show details here.

7.3 Action Sequence Patterns

We further analyze typical action sequence patterns indicative of SAT or DSAT sessions. We look into the following four patterns:

- actions prior to the end of a session, i.e., (*, *, END)
- actions next to a user command, i.e., (Command, *, *)
- actions after system execution, i.e., (Execute, *, *)
- actions before system execution, i.e., (*, *, Execute)

For each action sequence, we calculate its probabilities in SAT and DSAT sessions. We use the probability ratio as an indicator of to what degree its occurrence favors the session is satisfactory. If the ratio is greater than one, occurrence of action sequence favors the session is satisfactory. We use human annotated action types for the accuracy of analysis. Table 8 shows results.

The (*, *, END) pattern helps us analyze how SAT and DSAT sessions terminate. It shows that SAT sessions are more likely to end with a system execution. The probability ratio for (Command, Execute, END), (Yes, Execute, END), and (Select, Execute, END) are all above 1. In comparison, ending with a system action other than "Execute" is a strong indicator of DSAT sessions. This is not surprising since by our definition, "Execute" is the only way to achieve the task goal in device-dialog function tasks. If the session terminates with other actions, it probably means that the user gives up without achieving the goal. However, note that even in DSAT

Table 7. Effectiveness using action sequence (without prior probability of classes) to evaluate device-dialog function tasks.

User Satisfaction	Avg F_1	Pos F_1	Neg F_1	Accuracy
Human Annotated Action	0.785	0.861	0.709	0.813
Predicted Action	0.793	0.866	0.719*	0.820
ASR Quality	**Avg F_1**	**Pos F_1**	**Neg F_1**	**Accuracy**
Human Annotated Action	0.713	0.873	0.553	0.804
Predicted Action	0.699*	0.877	0.522**	0.805
Intent Classification Quality	**Avg F_1**	**Pos F_1**	**Neg F_1**	**Accuracy**
Human Annotated Action	0.772	0.886	0.658	0.831
Predicted Action	0.764	0.883	0.645	0.825

* and ** means p<0.05 and p<0.01 comparing to "Human Annotated Action".

sessions, the chances of ending with "Execute" is still much higher than other actions (not shown in the table). Therefore, this pattern can only be applied to a limited number of DSAT sessions (high precision but low recall).

Since it relies on correct "Execute" to achieve the task goal, we further analyze action sequence prior to and next to "Execute", i.e., (*, *, Execute) and (Execute, *, *). Results show that it is a strong indicator of SAT sessions if the user successfully commands the system to execute an operation at the first round (the ratio is as high as 3.84). "Execute" next to "Yes" or "Select" are weaker, but still indicative of SAT sessions. In comparison, if (Command, Execute) does not happen at the beginning of a session, it is an indicator of DSAT sessions, e.g., (Option, Command, Execute) and (Execute, Command, Execute) all have lower than 1 ratio. These sequences indicate errors in the previous rounds. For example, when intelligent assistant provides a list of options ("Option"), normally user should select one option ("Select"). If the user issues a new command instead, it means the options are not useful at all. This also supports using time_first_execute as a feature (Section 5.5.3).

We further analyze action sequence patterns next to "Execute". It shows that if the session does not terminate after an "Execute", it is nearly always an indicator of DSAT sessions. This is because that any further interactions after an "Execute" means that the executed operation is incorrect (such that the user needs to correct it or retry).

The last pattern is actions next to user commands. Results show that if the user confirms to execute an operation after "Command", it is a very strong indicator of SAT sessions (the ratio is 4.47). (Command, Execute, END) is also a strong indicator. In contrast, it indicates DSAT sessions if user continuously issues two commands in adjacent requests, e.g., (Command, Execute, Command). This pattern suggests system did not correctly understand the previous command so that user needs to retry a new command. Besides, providing an option list or asking users for questions after a user command probably means negative user experience. On the one hand, this is probably because "Option" and "Question" indicate partial understanding of previous user commands in Cortana. For example, in Table 4, user asks for the closest pharmacy, but system does not understand "closest" and just returns a list of pharmacies nearby. On the other hand, this may suggest limited user experience for such interaction style, because (Command, Option, Select) and (Command, Question, Answer) increases the time to "Execute" and the effort to achieve the task goal by involving more interactions.

However, comparing to Table 8, some action patterns suggest distinct information in predicting of ASR and intent classification quality. Table 9 shows the probability ratios for action patterns with large differences in the three evaluation scenarios. Some patterns indicating SAT sessions do not suggest positive ASR quality, e.g., (Confirm, Yes, Execute) and (Yes, Execute, END). Similarly, some frequent patterns in DSAT sessions do not indicate negative ASR quality. This probably explains why action sequence model is less effective in evaluating ASR quality. Similar differences were found between satisfaction and intent classification quality. However, we

Table 8. Action sequence (using human annotated action types) probability ratio in SAT and DSAT sessions for four patterns.

(*, *, END)			Ratio
Command	Execute	END	3.43
Yes	Execute	END	1.76
Select	Execute	END	1.34
Answer	Confirm	END	0.21
Command	WebSearch	END	0.08
Command	Error	END	0.00

(*, *, Execute)			Ratio
START	Command	Execute	3.84
Confirm	Yes	Execute	1.64
Option	Select	Execute	1.06
Option	Command	Execute	0.85
Execute	Command	Execute	0.70
Execute	Select	Execute	0.57
WebSearch	Command	Execute	0.34

(Execute, *, *)			Ratio
Execute	END	END	2.39
Execute	Command	Execute	0.70
Execute	Select	Execute	0.57
Execute	Command	WebSearch	0.28
Execute	Command	Confirm	0.00
Execute	Command	Option	0.00
Execute	Command	Question	0.00
Execute	Command	Error	0.00

(Command, *, *)			Ratio
Command	Confirm	Yes	4.47
Command	Execute	END	3.43
Command	Option	Select	0.85
Command	Execute	Command	0.51
Command	Question	Answer	0.40
Command	WebSearch	Command	0.23
Command	Option	Command	0.04

Table 9. Action sequence probability ratio difference in prediction of user satisfaction, ASR quality, and intent classification quality.

Action Sequence			Ratio		
			SAT	ASR	Intent
Yes	Execute	END	1.76	0.83	1.68
Confirm	Yes	Execute	1.64	0.78	1.54
Command	Option	Select	0.85	1.48	0.96
Execute	Command	Execute	0.70	1.10	0.77
Answer	Confirm	Yes	0.66	0.38	1.27
Question	Answer	Question	0.47	0.27	1.44
START	Command	WebSearch	0.39	1.26	0.19
Question	Answer	Confirm	0.37	0.28	1.33
Answer	Question	Answer	0.34	0.27	1.44
No	Option	Select	0.21	0.12	2.31
Confirm	No	Option	0.17	0.16	1.73
Answer	Confirm	No	0.09	0.12	1.35

Table 10. Effectiveness of using action sequence to predict user satisfaction and the quality of ASR and intent classification.

Features	User Satisfaction				ASR Quality				Intent Classification Quality			
	Avg F_1	Pos F_1	Neg F_1	Accuracy	Avg F_1	Pos F_1	Neg F_1	Accuracy	Avg F_1	Pos F_1	Neg F_1	Accuracy
Click	0.524^{**}	0.825^{**}	0.222^{**}	0.718^{**}	0.561^{**}	0.880^{**}	0.242^{**}	0.795^{**}	0.533^{**}	0.844^{**}	0.222^{**}	0.743^{**}
Request	0.815	0.901	0.729	0.856	0.743^{*}	0.903	0.583^{*}	0.846	0.796	0.904	0.688^{*}	0.858
Response	0.758^{**}	0.850^{**}	0.665^{**}	0.796^{**}	0.673^{**}	0.884^{**}	0.463^{**}	0.810^{**}	0.695^{**}	0.880	0.511^{**}	0.812^{*}
Acoustic	0.743^{**}	0.849^{**}	0.638^{**}	0.790^{**}	0.703^{**}	0.882^{**}	0.525^{**}	0.815^{**}	0.745^{*}	0.875^{*}	0.616^{*}	0.813^{*}
Action Sequence	0.819	0.892	0.746	0.850	0.763	0.903	0.623	0.848	0.776	0.892	0.660	0.839
Best Feature Set	All Features				Action + Request + Acoustic				Action + Request + Acoustic			
	0.852^{**}	0.920^{**}	0.783^{*}	0.886^{**}	0.786	0.909	0.664^{*}	0.859	0.825^{**}	0.913^{*}	0.736^{**}	0.874^{**}

* and ** means p<0.05 and p<0.01 comparing to "Action Sequence"; shaded cells indicate the best results among those using individual feature set.

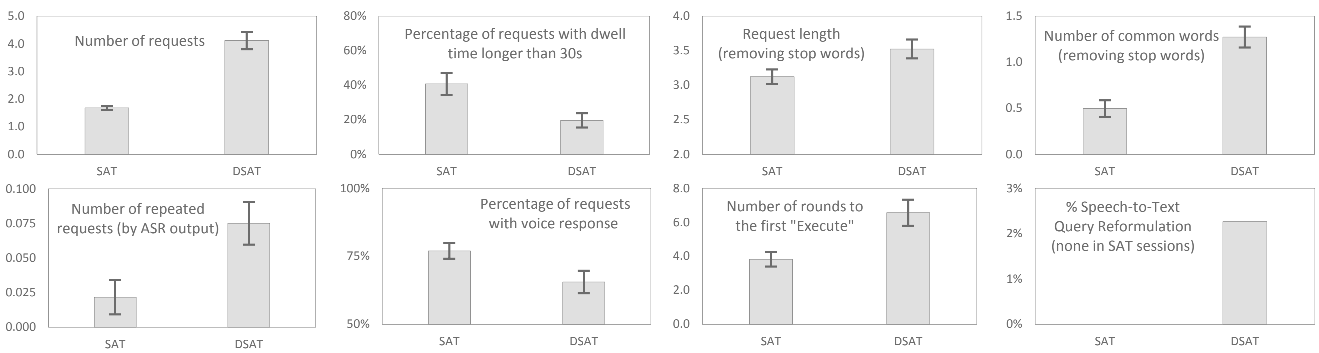

Figure 2. User behavior statistics (with standard error) in SAT and DSAT sessions. All differences are significant at 0.05 level.

still found many patterns indicating ASR and intent classification quality. This is why action sequence model can effectively evaluate ASR and intent classification quality. The differences suggest it is necessary to train different models for each evaluation scenario.

To conclude, results in this section show that the action sequence model is an effective approach for evaluating intelligent assistants in device-dialog function tasks. Besides, analysis of action patterns is a helpful tool to diagnose effective and ineffective interactions.

7.4 Other Features

We further evaluate and compare the effectiveness of generic features. We first evaluate each individual feature set separately. Then, we evaluate combinations of feature sets. We use gradient boosted decision tree for classification [5]. Table 10 shows the results.

7.4.1 Request Features

Request features and action sequence are the two best performed feature sets. They have comparable effectiveness in predicting user satisfaction. Action sequence features are relatively more effective in predicting ASR quality, but less accurate in prediction of intent classification quality. They outperform other features significantly in both F-measure and accuracy. They are also more effective than the naive action sequence model approach, especially in predicting ASR quality (average F-measure improved by about 10%).

Figure 2 further shows comparison of user behavior statistics in SAT and DSAT sessions. It shows that significant differences exist in many request characteristics between SAT and DSAT sessions, including: the number of requests in a session (1.68 vs. 4.12), the percentage of requests with dwell time longer than 30 seconds (40.7% vs. 19.6%), request length after removing stop words (3.12 vs. 3.53), the number of common words between adjacent requests (0.50 vs. 1.27), and the number of repeated requests (simply count by whether two adjacent requests have same ASR outputs; 0.022 vs. 0.075). These statistics explain why request features have strong performance in predicting user satisfaction.

In prediction of ASR and intent classification quality, we observed similar differences. However, the distinction between sessions with SAT and DSAT ASR quality are less significant in terms of request length (3.21 vs. 3.53, p=0.09) and the number of common words between adjacent requests (0.91 vs. 1.14, p=0.08). This explains why request features are relatively less effective in predicting ASR quality (but still achieved 0.743 F-measure).

7.4.2 Response Features

Response features are useful for predicting user satisfaction (the average F-measure is 0.758), but relatively less indicative of the quality of ASR and intent classification (with average F-measure 0.673 and 0.695). As shown in Figure 2, we also found significant

Table 11. Effectiveness of features for predicting user satisfaction, ASR and intent classification quality in web search tasks.

Features	User Satisfaction				ASR Quality				Intent Classification Quality			
	Avg F_1	Pos F_1	Neg F_1	Accuracy	Avg F_1	Pos F_1	Neg F_1	Accuracy	Avg F_1	Pos F_1	Neg F_1	Accuracy
Click	0.536**	0.749**	0.324**	0.649**	0.493**	0.687**	0.299**	0.579**	0.533**	0.695**	0.370**	0.595**
Request	0.796	0.840	0.751	0.810	0.721**	0.754*	0.689**	0.730**	0.755	0.742	0.769	0.765
Response	0.614**	0.764**	0.465**	0.679**	0.509**	0.714**	0.305**	0.601**	0.587**	0.722**	0.452**	0.639**
Acoustic	0.721**	0.771**	0.671**	0.732**	0.752*	0.754*	0.749	0.758	0.715**	0.728**	0.702**	0.725**
Best Feature Set	**All Features**				**Click + Request + Acoustic**				**Request + Acoustic**			
	0.798	0.858	0.738	0.822	0.772	0.784	0.76	0.777	0.769	0.766	0.772	0.777

* and ** means p<0.05 and p<0.01 comparing to "Best Feature Set"; shaded cells indicate the best results among those using individual feature set.

differences between SAT and DSAT sessions in the percentage of requests with voice responses (76.9% vs. 65.5%) and the number of rounds to the first "Execute" action in the session (3.82 vs. 6.56). However, we did not find the Jaccard similarity of results useful, probably due to the fact that only about 10% of the responses in device+dialog function tasks include a result list (counting both the web search results and answer items).

We found similar and significant differences in response features and related user behavior in predicting intent classification quality. However, we only observed a slightly significant difference in the number of rounds to the first "Execute" action between sessions with and without satisfactory ASR quality (4.38 vs. 5.75, p=0.07). Other response features are not very indicative of the ASR quality. This is probably why response features perform relatively worse in predicting ASR quality comparing evaluating user satisfaction and intent classification quality.

7.4.3 Acoustic Features

Results suggest that acoustic features can also effectively predict user satisfaction as well as the quality of ASR and intent classification in device function tasks. We found significant differences in speech recognition confidence scores between SAT and DSAT sessions, as well as between those with SAT and DSAT quality of ASR and intent classification. This suggests that Huang et al.'s approach [12] is not only effective for predicting ASR quality of utterances, but also correlate well with user ratings of ASR quality at session level for intelligent assistant tasks. This is also why it is effective for predicting user satisfaction and intent classification considering the correlations between them.

In addition, we found that detecting requests with slower speaking rates are predictive of sessions with dissatisfactory ASR quality. Figure 3 shows the percentage of requests with slower speaking rates at different slower ratios (with standard error). The percentage of slower speaking requests ("% request" in Figure 3) are consistently higher in session with dissatisfactory ASR quality when we set the slower ratio r below 70%. The differences are significant at 0.05 level when we set r to 0.15, 0.6 or 0.65. Similar differences exist between the SAT and DSAT sessions, but are less significant. However, we did not observe such differences between sessions with satisfactory and dissatisfactory intent classification quality. This suggests that slowing-down is probably only a strategy of the user to deal with speech recognition error, but not necessarily for intent classification errors.

7.4.4 Click Features

Click features have the worst performance comparing to others, despite they are important in predicting satisfaction in web search. We found this is mainly because users do not "click" results very often in device+dialog function tasks. Here click means to tap a result on the screen. The result can be a search result or an answer item returned in device+dialog function tasks (e.g., it can be a place answer when looking for locations). One possible reason is that intelligent assistants allow users to interact with result items through voice interaction, e.g., the "Option" and "Select" action. In such

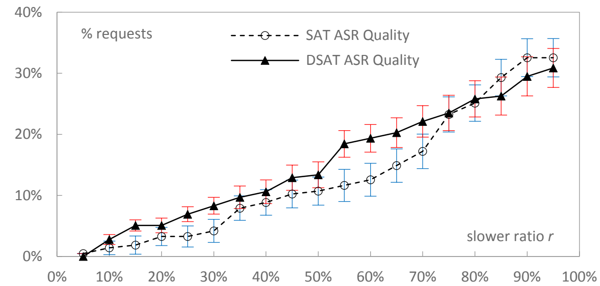

Figure 3. Percentage of requests (y-axis) whose speaking rate comparing to the previous requests falls below ratio *r* (x-axis).

case, users may rely less on tapping or other touch based interaction. Also, we did not observe significantly different user behavior related to click features. Thus, it is not surprising that click features are not as effective as they are in web search.

7.4.5 Using Multiple Features

After combining multiple features, we significantly improved the prediction of user satisfaction and intent classification quality by about 4%-6% in F-measure and accuracy (comparing to using the action sequence features). We finally achieved about 0.8 F-measure and 0.85 accuracy in all three evaluation scenarios. This confirms that our approach can effectively evaluate overall quality of intelligent assistants as well as its components.

8. WEB SEARCH AND CHAT TASKS

We further evaluate the effectiveness of features in web search and chat tasks. Table 11 shows the results for web search tasks.

Similar to the findings in device+dialog function tasks, results in Table 11 show that click features have very weak predictive powers for overall satisfaction and ASR and intent classification quality in web search tasks. The chances of clicking results remains low in web search tasks on intelligent assistants. Still, we did not observe significant differences in click features between SAT and DSAT sessions. However, different from the device+dialog function tasks, we cannot simply explain this by the availability of voice interaction to make selections, because Cortana did not provide users such function in web search scenario (e.g., at the time of the study, Cortana will not understand requests such as "open the first webpage").

One possible reason is that user becomes less willing to click and read details of a result page on mobile devices. Instead, they may prefer knowing answers directly, either displayed on the result webpage, or responded by Cortana in voice. Some participants' responses in the interview support this explanation: "For example with the scores it would have been nice if I just said San Francisco Giants and it just came up with a list of scores ... Rather than a website where I have to check it or a sports blog." [sic] Another evidence supporting this claim is that 14.8% requests in SAT sessions have voice responses (an indicator that Cortana provides extracted instant answers to the query in web search tasks), which is significantly higher than only 6.6% in the DSAT sessions.

Request features and acoustic features are also very effective for evaluating web search tasks. Request features are the strongest

Table 12. Effectiveness of evaluating chat tasks.

	Avg F_1	Pos F_1	Neg F_1	Accuracy
User Satisfaction	0.673	0.563	0.783	0.715
ASR Quality	0.702	0.767	0.637	0.724
Intent Classification Quality	0.590	0.329	0.851	0.761

feature set in predicting user satisfaction and intent classification quality. In comparison, acoustic features outperforms others in predicting ASR quality. Further analysis found differences of related user behavior similar to those in device+dialog function tasks. Comparing to DSAT sessions, SAT sessions of web search tasks have significantly less (1.83 vs. 3.68) and shorter queries (4.34 vs. 4.72 words), as well as longer query dwell time (43% queries' dwell time is longer than 30s vs. 23% in DSAT sessions). ASR confidence score and queries with slower speaking rates also differ significantly in SAT and DSAT sessions. However, we did not find significant differences in query similarity (i.e., the number of common words) between SAT and DSAT sessions in web search tasks.

Besides, our results also confirm Shokouhi et al.'s findings [26] that switching from voice to text input is a signal of dissatisfactory user experience. As shown in Figure 2, among all the SAT sessions of web search tasks, we observed no voice-to-text switches. In comparison, 2.3% of the query reformulations in DSAT sessions are from voice to text. This suggests that voice-to-text is a highly precise indicator for DSAT sessions (but the coverage is limited).

After combining multiple features, we achieved better prediction results. The best approach has 0.798 average F-measure predicting user satisfaction, which is comparable to those reported for state-of-the-art approaches in normal web search scenario [32].

We further apply request, response, and acoustic features to chat tasks (because it seems meaningless to apply click features to chat tasks). Table 12 shows that these features are still predictive of user satisfaction and ASR quality in chat tasks, although less accurate than they are device+dialog function and web search tasks. They have limited effectiveness in evaluating intent classification quality in chat tasks. This suggests that automatic evaluation of chat quality is more challenging, just as it is probably also more challenging to solve the chat task comparing to other two types of tasks due to its open-ended nature.

To conclude, results suggest that we can apply features adopted in previous web search satisfaction studies (e.g., most of the click, request, and response features), as well as those newly designed for intelligent assistants (e.g., acoustic features, voice-to-text, etc.) to evaluate web search tasks and chat tasks in intelligent assistants. However, some of them are less effective in intelligent assistant, e.g., click features. Besides, our results show that using the same features, we can train different models to effectively predict ASR and intent classification quality as well. In addition, the effectiveness of acoustic features suggests that we can use voice interaction and related user behavior to improve modeling of applications in voice enabled scenario, e.g., intelligent assistant.

9. DISCUSSION AND CONCLUSION

This paper explored evaluation of intelligent assistants. This is a challenging problem due to the variability of tasks and topics. We developed task-independent approaches to address the challenge and proved their effectiveness in experiments. Results demonstrate that using the same approach, we can evaluate not only intelligent assistant itself, but also its two components: speech recognition and intent classification. Results of a user study show the quality of both components affect user experience in intelligent assistants.

Admittedly, our study is limited in the coverage of t asks and the way data is collected. Due to the sensitivity and difficulty of data collection, we did not study a popular function of many intelligent assistants, i.e., proactive suggestions. This function requires lots of personal information, while in our lab study it is difficult to prepare such data for participants to perform tasks. Besides, we develop our tasks based on a very early sample of Cortana logs. These tasks are not necessarily representative of other products, nor Cortana today. In addition, using lab study to collect data is a double-bladed sword. An alternative approach usually adopted is to ask third-party judger to assess session quality. In comparison, lab study can collect real user's judgments, but cannot simulate the real environment of using intelligent assistants, which may affect user interaction pattern. For example, we may observe a longer dwell time when user is driving, and user may not repeat requests but switch to text more often when it is noisy. Although many user behavior signals may change, this should not affect the effectiveness of our approach, as long as we can detect and train models for the environment.

Another issue we did not consider in this paper is the variability of interface and interaction. Comparing to well-studied applications (e.g., web search and the "10 blue links"), there is no "standard" interface or interaction mode in intelligent assistants, especially in device+dialog function tasks. On the one hand, there are different ways of implementing the same action, e.g., the intelligent assistant can ask user to confirm and wait for user response, or simply think the user agrees if there is no explicit response after a few seconds. When asking users for more information, it can ask by voice and wait for voice response, or just show input box on the screen and wait for text input. On the other hand, the set of actions may evolve in the future, although we found it is enough to characterize interaction in existing products. Depending on the availability of actions and their implementation, we may observe different action patterns, and we may need to retrain models to evaluate new interface designs. This suggests we should be cautious to use implicit feedback-based models to evaluate systems with different interface designs.

Comparing the three types of tasks, our approach works the best in device function tasks, and the worst in chat tasks. We suspect it is due to the number of features applicable to each task. As analyzed in Section 7.3, many action patterns indicate positive or negative user experience in device function tasks, but they are not available in web search and chat. Comparing to web search, it is meaningless to use click features as well as many other features in chat tasks, such as request type, voice-to-text, result similarity, etc. In addition, this may also be caused by the subjectivity of tasks. That device+dialog function tasks are the least subjective one, because the goal is clear and straight-forward, i.e., commanding intelligent assistant to do something. In contrast, some of the web search tasks are more open-ended and complex, e.g., finding flight schedule. Chat is the most open-ended task, since different users may judge chat quality differently, e.g., some may prefer funny responses, while some others may expect the conversation to be as natural as talking to real person. We suspect the more open-ended a task is, the more difficult we can train a unified model to evaluate the task, because the subjectivity of the task may result in distinct user behavior.

Our study also assumes the existence of an accurate classifier for the three task types. Although this is reasonable, it is unclear how actual accuracy of the classifier would affect the effectiveness of the overall evaluation process. In addition, we only adopt satisfaction as the user experience measure in this study. It is unclear how well our approach can predict other user experience measures such as task success and efforts. These are left for future work. Despite these limitations, we push forward studies of intelligent assistant as well as modeling of user satisfaction and evaluation.

First, we are the first study to address the challenge of evaluating intelligent assistants. Although similar studies exist in web search, the diversity of tasks and interaction are not comparable to those in intelligent assistants. Our approach is task-independent. Although some features (e.g., action sequence) only apply to device+dialog function tasks, the majority is shared among all the tasks, making it a generalizable approach for other products and future changes.

Second, our results demonstrate that, using the same approach, we can train models to evaluate not only overall user satisfaction in intelligent assistant, but also the quality of speech recognition and intent classification. This contributes to more detailed diagnosis of intelligent assistants in evaluation, as well as a user-centric way of evaluating related techniques (e.g., ASR) in specific applications.

Third, through user behavior analysis, we found lots of patterns and behavioral signals indicating user experience in intelligent assistants. Although some of them may be affected by system design, most of the patterns are potentially generalizable to similar applications, e.g., the session does not terminate with an Execute can be a strong signal of user abandon. These patterns are useful for related studies and evaluation of other systems.

As we discussed, it requires further work to refine a few places in our study, e.g., to consider uncovered tasks, environments, and variability of interface designs, to examine the effects of task type classification accuracy on the effectiveness of the approach, to correlate with other user experience measures, and to compare native and non-native speakers. Solutions to these issues will further improve the performance and reliability of our approach.

ACKNOWLEDGMENT

Jiepu Jiang was an intern at Microsoft Research while conducting the study, and he was supported in part by the Center for Intelligent Information Retrieval at the time of writing this article. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Ageev, M., Guo, Q., Lagun, D. and Agichtein, E. (2011). Find it if you can: a game for modeling different types of web search success using interaction data. *Proc. SIGIR '11*, 345–354.

[2] Smith R.W. and Hipp, D.R. (1995). *Spoken Natural Language Dialog Systems: A Practical Approach*. Oxford University Press.

[3] Feild, H.A., Allan, J. and Jones, R. (2010). Predicting searcher frustration. *Proc. SIGIR '10*, 34–41.

[4] Fox, S., Karnawat, K., Mydland, M., Dumais, S. and White, T. 2005. Evaluating implicit measures to improve web search. *ACM TOIS*, 23(2), 147–168.

[5] Friedman, J., Hastie, T., Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*.

[6] Hassan, A. (2012). A semi-supervised approach to modeling web search satisfaction. *Proc. SIGIR '12*, 275–284.

[7] Hassan, A., Jones, R. and Klinkner, K.L. (2010). Beyond DCG: user behavior as a predictor of a successful search. *Proc. WSDM '10*, 221–230.

[8] Hassan, A., Shi, X., Craswell, N. and Ramsey, B. (2013). Beyond clicks: query reformulation as a predictor of search satisfaction. *Proc. CIKM '13*, 2019–2028.

[9] Hassan, A., Song, Y. and He, L. (2011). A task level metric for measuring web search satisfaction and its application on improving relevance estimation. *Proc. CIKM '11*, 125–134.

[10] Hassan, A., White, R.W., Dumais, S.T. and Wang, Y.M. (2014). *Proc. WSDM '14*, 53–62.

[11] Heck, L.P., Hakkani-Tür, D., Chinthakunta, M., Tür, G., Iyer, R., Parthasarathy, P., Stifelman, L., Shriberg, E. and Fidler, A. (2013). Multi-modal conversational search and browse. *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia*, 96–101.

[12] Huang, P.S., Kumar, K., Liu, C., Gong, Y. and Deng, L. (2013). Predicting speech recognition confidence using deep learning with word identity and score features. *Proc. ICASSP*, 7413–7417.

[13] Huffman, S.B. and Hochster, M. (2007). How well does result relevance predict session satisfaction? *Proc. SIGIR '07*, 567–574.

[14] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM TOIS,* 20(4), 422–446.

[15] Järvelin, K., Price, S., Delcambre, L.L. and Nielsen, M. (2008). Discounted Cumulated Gain Based Evaluation of Multiple-Query IR Sessions. *Proc. ECIR '08*, 4–15.

[16] Jeng, W., Jiang, J. and He, D. (2013). Users' Perceived Difficulties and Corresponding Reformulation Strategies in Voice Search. *Proc. HCIR 2013*.

[17] Jiang, J., Hassan Awadallah, A., Shi, X. and White, R.W. (2015). Understanding and Predicting Graded Search Satisfaction. *Proc. WSDM '15*, 57–66.

[18] Jiang, J., He, D. and Allan, J. (2014). Searching, browsing, and clicking in a search session. *Proc. SIGIR '14*, 607–616.

[19] Jiang, J., Jeng, W. and He, D. (2013). How do users respond to voice input errors? lexical and phonetic query reformulation in voice search. *Proc. SIGIR '13*, 143–152.

[20] Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S. and Maloor, P. (2002). MATCH: An architecture for multimodal dialogue systems. *Proc. ACL '02*, 376–383.

[21] Kim, Y., Hassan, A., White, R.W. and Zitouni, I. (2014). Comparing client and server dwell time estimates for click-level satisfaction prediction. *Proc. SIGIR '14*, 895–898.

[22] Kim, Y., Hassan, A., White, R.W. and Zitouni, I. (2014). Modeling dwell time to predict click-level satisfaction. *Proc. WSDM '14*, 193–202.

[23] Kotov, A., Bennett, P.N., White, R.W., Dumais, S.T. and Teevan, J. (2011). Modeling and analysis of cross-session search tasks. *Proc. SIGIR '11*, 5–14.

[24] Niu, X. and Kelly, D. (2014). The use of query suggestions during information search. *IP&M*, 50(1), 218–234.

[25] Philips, L. (1990). Hanging on the Metaphone. *Computer Language*, 7(12), 39–44.

[26] Shokouhi, M., Jones, R., Ozertem, U., Raghunathan, K. and Diaz, F. 2014. Mobile query reformulations. *Proc. SIGIR '14*, 1011–1014.

[27] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C. and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3), 339–373.

[28] Traum, D.R. (2000). 20 questions on dialogue act taxonomies. *Journal of semantics*, 17(1), 7–30.

[29] Tur, G. and De Mori, R. (2011). *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

[30] Wahlster, W. (2006). *SmartKom: foundations of multimodal dialogue systems*. Springer.

[31] Walker, M.A., Litman, D.J., Kamm, C.A. and Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. *Proc. ACL '97*, 271–280.

[32] Wang, H., Song, Y., Chang, M.W., He, X., Hassan, A. and White, R.W. (2014). Modeling action-level satisfaction for search task satisfaction prediction. *SIGIR '14*, 123–132.

[33] Young, S., Gasic, M., Thomson, B. and Williams, J.D. (2013). POMDP-based statistical spoken dialog systems: A review. *Proc. IEEE*, 101(5), 1160–1179.