Predicting User Engagement Status for Online Evaluation of Intelligent Assistants

Rui Meng[1(✉)], Zhen Yue[2], and Alyssa Glass[3]

[1] University of Pittsburgh, Pittsburgh, PA 15213, USA
rui.meng@pitt.edu
[2] Disney Streaming Service, CA, USA
zhen.yue@disney.com
[3] Apple Inc., CA, USA

Abstract. Evaluation of intelligent assistants in large-scale and online settings remains an open challenge. User behavior based online evaluation metrics have demonstrated great effectiveness for monitoring large-scale web search and recommender systems. Therefore, we consider predicting user engagement status as the very first and critical step to online evaluation for intelligent assistants. In this work, we first propose a novel framework for classifying user engagement status into four categories – fulfillment, continuation, reformulation and abandonment. We then demonstrate how to design simple but indicative metrics based on the framework to quantify user engagement. We also aim for automating user engagement prediction with machine learning methods. We compare various models and features for predicting engagement status using four real-world datasets. We conduct detailed analyses on features and failure cases to discuss the performance of current models as well as potential challenges.([1]Resources used in this study can be found at https://github.com/memray/dialog-engagement-prediction.)

Keywords: Intelligent assistant · User engagement · Online evaluation

1 Introduction

The increasing popularity of intelligent assistants such as Alexa, Siri and Google Home has attracted broad attention to human-machine dialogue systems, but also brought challenges for evaluating the performance of dialogue systems in online environments. Previous research demonstrated that the most effective way to improve any online system is to optimize it for end-user engagement [6]. For example, recommender systems can be optimized for user click and dwell time [43] and web search systems can be optimized for click-through rate [9]

R. Meng, Z. Yue and A. Glass—This work was done when the authors were at Yahoo Research.

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12656, pp. 433–450, 2021.
https://doi.org/10.1007/978-3-030-72113-8_29

and reformulation rate [14]. Nevertheless, designing proper metrics to optimize online intelligent assistant systems remains a big challenge.

Previous studies seeking to evaluate dialogues systems mainly focus on the performance of individual system component rather than overall user engagement. The common practice in system-oriented evaluation is breaking down the dialogue system into parts, such as dialogue act classification and state tracking, and evaluating the performance of each component respectively. However, we cannot assess the performance of the whole system by simply aggregating the performance of each component. There were several methods developed to evaluate the overall system performance. For example, one can evaluate the quality of system responses by measuring their similarities to ground-truth responses with metrics like BLEU [39,40]. However, users' requests in online environment are very diverse and it is very expensive to build ground-truth datasets, which make the evaluation hard to scale up for online scenarios.

Research in web search has a long history of conducting large-scale online evaluation utilizing user engagement and behavior signals [8,13–15]. The idea was to regard possible user interaction outcomes as different engagement types, such as long-dwell click, query reformulation and abandonment. These engagement types can then be used to gauge search success and cost, thus making these measurements scalable for online evaluation. We think that the same idea can be adopted to the evaluation of intelligent assistants as well. For example, we can classify each user utterance in a dialogue system into success and failure requests. Previous research proposed a conceptual framework PARADISE [41] for evaluating dialogue systems. It pointed out that a successful dialogue system should maximize task success and minimize cost. Same for the online evaluation of intelligent assistant, we should not only focus on whether or not users' requests have been fulfilled but also measure how much effort it takes. We cannot simply use the conversation length as measurement for cost, since it might take multiple necessary turns to finish a complex user request. Instead, we should focus on whether or not the interaction is necessary for the intelligent assistant to fulfill the request. In order to solve the problem, we proposed a novel scheme categorizing users' utterances into different types of engagement status, with which we can design metrics to measure task success and cost for online evaluation of intelligent assistant.

Furthermore, we aim for a more challenging task, delivering an automatic method for predicting the user engagement status. In recommendation and search, researchers utilize behavior signals such as dwell time and query content features to predict user engagement. Similarly, we utilize interaction signals between users and intelligent assistants to predict users' engagement status. Comparing to the short queries in web search, the interaction between users and intelligent assistants contains rich contents, which can be used for creating sophisticated automatic methods. We investigate various machine learning models and feature settings for the engagement prediction task with four newly annotated datasets.

2 Related Work

2.1 Evaluation of Intelligent Assistants

There are several major methods being widely used for evaluating intelligent assistants: (1) Evaluation on specific components [10,22,33]. People have established several tasks to examine certain aspects of the systems, such as dialog state tracking and dialogue act classification, and evaluate them by metrics like precision and recall. While these evaluations are useful to identify problems in each component, the outcomes cannot reflect the overall performance of the dialogue system. (2) Evaluation by comparing system responses with ground-truth responses [28,36,39]. This type of approaches is broadly adopted for response generation. The basic idea is to measure the similarity between generated responses and ground-truth responses with metrics like BLEU [34]. However, a high degree of token matching may imply its readability, but does not mean it is a logical response, and such methods have been proved correlated poorly with the human judgment [29]. (3) There are a few tasks aiming to detect problematic system responses which share a similar motivation to our study, such as error detection [26,31] and breakdown detection [18]. But in these tasks, the cost of communication is not considered and task boundaries are presumably given. In the real world, both task success and cost affect users' experience considerably and users can move to a new task anytime, therefore our specially designed framework, detecting both system failures and user request boundaries, are more suitable for evaluating real-world systems.

2.2 User Engagement Prediction

User satisfaction rating in dialogue systems has been discussed for a long time [21,35,38]. A wide variety of techniques and features have been studied [4,11,42], as well as some recent efforts on the basis of deep neural networks [30]. Most of these studies output a holistic satisfaction rating for the entire dialogue, but it cannot offer any further information about how the system fails to satisfy users. Therefore it is not a reliable optimization target that can be used for improving the dialogue system.

PARADISE [41] framework tackles this problem by breaking down the measurement of user satisfaction into two parts: task success and dialogue cost. However, PARADISE was proposed more than twenty years ago and is more a conceptual framework than a practical solution: the two factors are too general to implement and they did not provide any automatic method practicable in a nowadays large-scale setting. For example, to measure task success it requires conversations to be represented as task-specific attribute-value pairs which are not flexible nor transferable. The measurements of dialogue cost using utterance length and number of dialogue turns are also arguable, because for many user requests, say the restaurant inquiry in DSTC2 (see Table 1), it can take several necessary turns to finish. Our work is also similar to [1,12,20,25] in the sense of using user behavior signals to predict short-term user engagement/satisfaction.

3 Classifying and Measuring User Engagement

3.1 Framework for Classifying User Engagement

Before introducing the classification framework, we would like to clarify several concepts that play important roles in it. We consider a continuous interaction between user and system within a small range of time as a *session*. And a *session* is comprised of a number of *utterances*, each of which is issued by either the user or the system. We can further group *utterances* into *tasks* basing upon user's potential information needs. For example, Table 1 shows a *session* sampled from the dataset DSTC2, in which a user (**User**) consults a dialogue system (**Bot**) about restaurants and the system asks the user to provide necessary information to narrow down the scope. It consists of 7 effective pairs of *user utterance* and *system utterance*, and 4 basic *tasks* (or *user requests*): requesting a restaurant (#1-#3), requesting its address (#4), and requesting its phone number (#5) and requesting its postcode (#6-#7).

With the goal of measuring both success and cost of user interaction with intelligent assistant, we propose a four-class utterance classification scheme. Each class represents the engagement status of a user after issuing an utterance:

Table 1. An annotated session from ***DSTC2***.

# Turn	Utterance	Label	Task
	Bot: Hello, welcome to the Cambridge restaurant system? You can ask for restaurants by area, price range or food type. How may I help you?		
1	**User:** Moderately priced	R	
	Bot: What part of town do you have in mind?		
2	**User:** North	C	**Success**
	Bot: Would you like something in the cheap, moderate, or expensive price range?		
3	**User:** Moderate	F	
	Bot: Golden wok is a nice restaurant in the north of town in the moderate price range		
4	**User:** Address	F	**Success**
	Bot: Sure , golden wok is on 191 Histon Road Chesterton		
5	**User:** Phone number	F	**Success**
	Bot: The phone number of golden wok is 01223 350688		
6	**User:** Postcode	R	
	Bot: Golden wok is a nice restaurant in the north of town in the moderate price range		
7	**User:** Postcode	A	**Failure**
	Bot: Golden wok is a nice restaurant in the north of town in the moderate price range		
8	**User:** Thank you good bye		

- ***Fulfillment* (F)**: current user request is understood and fulfilled by the system.
- ***Continuation* (C)**: current user request is understood by the system but more interactions are needed.
- ***Reformulation* (R)**: current user request is wrongly or incompletely understood by the system, and user repeats/rephrases this request in the next turn.
- ***Abandonment* (A)**: current user request is not understood or fulfilled by the system, so the user abandons this request by closing the conversation or starting a new request.

In Table 1, user utterances are annotated with the proposed classification scheme, as shown in the rightmost column. Specifically, the user told the system her desired price range and location (*Turn #1* and *#2*), but the system failed to catch the first price information. After the user repeated it (*Turn #3*), the system returned a restaurant that the user might be interested in. It is worth noting that the annotation of an utterance utt_i has to be one-turn delayed, determined after knowing the future responses (utt_{i+}) from both the system and the user side. Therefore the *Turn #1* utterance is annotated as '**R**'. The system replied correctly in both *Turn #4* and *#5*. The user requested the postcode in *Turn #6* and repeated it in the *Turn #7*, and in the end she terminated the conversation after an incorrect response. Thus #6 is labeled as '**R**' and #7 is '**A**'.

Table 2. Two dimensions along which the proposed classification scheme can be binarized.

	Ongoing	Ending
Correctly responded	Continuation	Fulfillment
Wrongly responded	Reformulation	Abandonment

From the definition of each type and the examples, we can see that the proposed classification scheme is clearly defined and highly explainable, because the four classes of user utterance are mutually exclusive and each depicts an explicit user behavior. As shown in Table 2, our scheme can be thought as two orthogonal binary classifications by checking (1) if the user continues or terminates the current task/request and (2) if the system gives a correct or wrong response. Based on the two conditions, one can assign labels much easier than giving a subjective score [30,42] or a sentiment class [4]. For example, we can split the session in Table 1 into four tasks and classify them in into **Success** or **Failure** using **F** or **A** as task boundary and satisfaction indicator.

3.2 Online Evaluation Metrics Based on User Engagement Status

In the context of industrial web services, ahead of optimizing any system to improve its performance for end users, it is common to first determine how to measure the user engagement with a system, i.e. creating engagement metrics that accurately reflect the user-end performance of a product. With the proposed

classification scheme, not only are we able to understand the engagement status of a user after each request immediately, it also enables us to define a series of evaluation metrics to monitor the system performance in an online manner. Similar to PARADISE [41], we define two metrics to measure the user engagement, from the aspect of success and cost respectively.

Since *Fulfillment* or *Abandonment* indicates the boundary of a task as well as a good/bad user experience, we can split a session to several tasks, and then group them into successful/unsuccessful tasks. We define the **Success Rate** of a session $\mathbb{S}$ as the percentage of success tasks as in Eq. (1), where $\#(TASK_{success \in \mathbb{S}})/\#(TASK_{\in \mathbb{S}})$ denotes the number of successful/all tasks in the session $\mathbb{S}$:

$$SuccessRate = \frac{\#(TASK_{success \in \mathbb{S}})}{\#(TASK_{\in \mathbb{S}})} \tag{1}$$

Similarly, we would like a metric to represent how efficiently a system can respond to requests. Firstly, we can use a statistic of *Reformation* to represent the degree to which a user repeats in a task. We define **Reformulation Rate** of a session as the average percentage of reformulated utterances in each task as in Eq. (2), where $\#(UTT_{reform \in T})/\#(UTT_{\in T})$ denotes the number of *Reformulation*/all user utterances in the task **T**. Furthermore, we hope the final metric can also reflect the degree of user fatigue in the interaction. Though *Continuation* utterances are considered necessary in most cases, we think long dialogues should be avoided and better interaction models can be designed to shorten the length. To this end, we define **Fatigue Value** as the average thresholded length of a task as in Eq. (3) – if a task is longer than α turns (α is a preset parameter), we count its fatigue value as $\#(UTT_{\in T}) - \alpha$ otherwise as 1. Then we define **Efficiency Rate** as in Eq. (4), which means the less reformulation or the shorter dialogue in each task, the more efficient we consider a session is.

$$ReformRate = \sum_{T \in \mathbb{S}} \frac{\#(UTT_{reform \in T})}{\#(UTT_{\in T})} \tag{2}$$

$$FatigueValue = \frac{\sum_{T \in \mathbb{S}} max(1, \#(UTT_{\in T}) - \alpha)}{\#(TASK_{\in \mathbb{S}})} \tag{3}$$

$$EfficiencyRate = \frac{1 - ReformRate}{FatigueValue} \tag{4}$$

Lastly, we can define a unified **User Engagement Score** representing the overall user experience of a session. Here we define it as a plain arithmetic mean of both Success Rate and Efficiency Rate (Eq. (5)), but it can be extended to more sophisticated forms to fit specific cases and applications.

$$UE_{SCORE} = \frac{SuccessRate + EfficiencyRate}{2} \tag{5}$$

Overall, this classification scheme and metrics are conducted at the utterance level, which is easy-to-run for real-time systems. Furthermore, as the proposed

user engagement status can indicate a positive/negative experience explicitly, the corresponding metrics are explainable and instructive for troubleshooting potential system problems.

3.3 Datasets

Since there does not exist dataset available for our study, we collect data from four intelligent assistants – **DSTC2, DSTC3, Yahoo Captain (YCap), Google Home (GHome)** – and annotate them. All dialogues take place between a human and a real system , which fit our goal of evaluating real intelligent assistants. **DSTC2** [16] and **DSTC3** [17] are task-specific datasets, in which users call the system to inquire restaurant or tourist information. **YCap** is an SMS-based family assistant developed by Yahoo!. It supports functions like setting a reminder for family members, maintaining and sharing shopping list etc. **GHome** is collected from real users of Google Home, an intelligent home device powered by Google Assistant and responding to voice control with multiple functions. The **GHome** dataset is the most complicated among the four datasets. It not only covers a broad range of tasks including reminder, timer, search, in-house device control etc., but also supports open-domain chitchat. For **YCap** and **GHome**, as all the conversations are concatenated in a log file, we split dialogues by checking if the interval between two utterances is more than 10 min. Then we randomly select 1,000 anonymized dialogues from each dataset for annotation. We ask professional annotators to judge the engagement status of each user utterance. The first pass of annotation is done by two annotators independently and the conflicts are resolved by the third annotator. The inter-annotator agreement achieves a kappa of 0.790, indicating the proposed scheme is understandable and easy-to-annotate. Table 3 shows the statistics of each dataset. **#(user utt)** indicates the number of data examples used in the following study. Here we highlight several observations:

Table 3. Statistics of four annotated dialogue datasets

Dataset	#(task)	#(utt) per task	#(word) per utt	#(user utt)	C%/R%/F%/A%	Suc%/Effic%/Ref%/Fatigue	UE
DSTC2	2,825	4.36	3.87	5,700	28.6%/21.9%/47.1%/2.5%	93.8%/41.9%/17.0%/3.33	0.679
DSTC3	3,020	4.64	4.00	5,856	28.1%/20.4%/48.0%/3.6%	90.1%/45.1%/14.6%/4.01	0.676
YCap	2,733	2.37	4.49	3,530	7.6%/14.9%/70.8%/6.6%	91.8%/78.7%/12.4%/1.35	0.853
GHome	4,561	2.98	4.17	5,241	2.3%/10.6%/75.7%/11.4%	87.4%/73.3%/8.3%/1.80	0.804

1. By checking the average number of user utterances (#(utt) per task), dialogues of the text-based system (**YCap**) are averagely shorter than the ones of spoken systems. Also, since **YCap** takes the user typed input directly, though the data is intact from the error-prone ASR, it suffers from the typo errors of user inputs.
2. *Continuation* accounts for a large part in **DSTC2/3**. This is because, in order to inquire restaurants of interest, users have to interact with the system for

many turns. But in **YCap** and **GHome**, most user requests can be solved in one turn, such as "set up a reminder at 8pm" or "turn on the light".
3. Utterances of *Fulfillment* and *Continuation* take the major part across all four datasets. By summing up these two types, we can see a basic success rate of each system at the utterance level (75.6%:75.9%:78.4%:78.0%).
4. *Abandonment* on task-specific systems is notably fewer than on more complicated systems such as **GHome**, which can be attributed to the fact that tasks in **GHome** are more diverse and difficult.
5. Overall, the class distribution is very skewed, and models may severely suffer from the data scarcity on minor classes.

3.4 Case Study of User Engagement Metrics

We compute the user engagement scores of each dataset as shown in Table 3. We also visualize the distribution of session scores in a 2-D scatter plot in Fig. 1. Here we set α to 2 for all datasets to discount tasks longer than 3 turns. From the table we can see that **YCap** and **GHome** perform overall better than the other two. All four assistants are able to achieve a satisfactory success rate, but **DSTC2** and **DSTC3** perform badly on efficiency. Specifically, among all the successful sessions ($SuccessRate = 1.0$), the ratio of tasks whose efficiency is less than 0.5 is more than 50% , but in **YCap** and **GHome** the percentage is less 20%. Since the system used in DSTC datasets is considerably dated, we think the high *Reformulation Rate* can be attributed to the poor ASR quality. What's more, we can also use the metric to quickly identify problematic dialogues, i.e. the ones have low engagement scores. There are 35/63/9/13 sessions whose overall score is less than 0.2. After manually examining those sessions, we find the most prominent issues affecting **DSTC2** and **DSTC3** are poor ASR and language understanding ability. A user may repeat 5 times to make the system understand what the request is about. **YCap** only takes user commands matching particular templates and oftentimes users reform their request several times to make it accepted. In **GHome**, problems are more diverse since it supports various functions and users can ask open-domain questions to which the system cannot handle well yet.

The goal here is to demonstrate how metrics based on the proposed user engagement status could be used to evaluate system performance and troubleshoot failures, and these metrics can be easily adopted for online A/B testing.

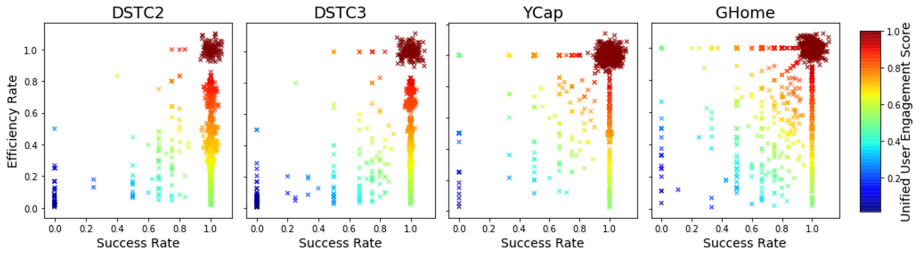

Fig. 1. 2-D scatter plot of user engagement metrics. A jitter is applied to show the size of clusters.

4 Automatic Prediction of User Engagement Status

Now we have defined a series of user engagement metrics for intelligent assistants, the next step is to automate the prediction of user engagement status so that the proposed metrics can be used in large-scale and online applications. We formalize this task as a four-class classification problem at the utterance-level.

4.1 Model Setting

We mainly examine two groups of models. The first group is classic classifiers, working together with hand-crafted features. We consider three models which are broadly used for text classification: Logistic Regression (**LR**), Support Vector Machine (**SVM**) and Random Forest (**RF**). The second group is convolutional neural networks (**CNN**), which learn continuous representations without manual feature engineering and allow us to leverage word vectors pretrained on a large corpus, with which a significant performance boost has been observed in various NLP studies. We use two variants of CNNs proposed by [23]: **CNN.Rand** and **CNN.MultiCh** (multi-channel). We have also tested a group of models based on recurrent neural networks, however they cannot converge well (may be due to the size of datasets). Thus their scores are not discussed.

4.2 Feature Setting

We think the status of user engagement is system-independent and identifiable by analyzing the dialogue contents. Therefore we only use features that can be extracted from transcriptions and ignore the other types of system-specific outputs (e.g., dialogue state, ASR output). From each utterance, we define six groups of features and use them to predict user engagement status. Besides, we notice that *Reformulation* implies a high semantic similarity between two user requests, thus we also define a set of *similarity features* for each feature group. We use '#feature_x'to denote the count of the feature.

- **Basic Features** Three subgroups of features indicating basic information of each utterance: (1) Utterance length: **utt_length**; (2) Time: **if_dialogue_start**, **if_dialogue_end**, **#utt_from_end**, **#utt_to_end**, **time_percent**; (3) Three features based on common user commands (e.g. "remind", "alarm", "add item"): **command_word** (one-hot vectors), **#command_word**, **jaccard_sim** (jaccard similarity between two adjacent user utterances).
- **Phrasal Features** We apply Stanford CoreNLP to extract 1) noun phrases (**noun_phrase**) and 2) entities (**entity**) from each utterance and represent them as one-hot vectors. We define three similarity features: 3) **repetition**: if any noun phrase/entity is repeated in two adjacent user utterances; 4) **#repetition**: number of repeated noun phrases/entities; 5) **jaccard_sim**.
- **Syntactic Features** The syntactic dependencies can help us understand the core components of utterances. From the dependency tree of each utterance, we can extract three types of syntactic features and represent them as one-hot vectors: 1) root word (**root_word**), 2) topmost subject word (**subject_word**) and 3) topmost object word (**object_word**). For similarity we only check if there is any repetition of these words between two user utterances: 4) **repeat_root_word**, 5) **repeat_subject_word** and 6) **repeat_object_word**.

- **N-grams Features** The n-grams is considered one of the most robust features for text classification. We extract 1-/2-/3-grams and represent them as one-hot vectors weighted by TfIdf. Two similarity features: 1) **edit_distance** (Levenshtein edit distance) and 2) **jaccard_sim**.
- **Topic Features** We apply the Latent Dirichlet Allocation (LDA) to capture the topical information in utterances (**lda_feature**). We train separate LDA model for each dataset and set its dimension to 50. We use the cosine similarity of LDA vectors between two user utterances (**lda_cosine**) as its similarity feature.
- **Distributed Representations** Previous studies [5,19,24,27] have demonstrated the efficacy of transferring language knowledge learned from rich resources to new tasks. Since we have only a limited amount of dialogues for training, we would like to know if we could utilize large text representation models to alleviate data shortage. Here we present three models to represent utterances: **Word2Vec** [32] (averaging word vectors in the utterance, dimension = 300), **Doc2Vec** [27] (treating each utterance as a document, dimension = 300) and **Skip-thought** [24] (using bi-skip model, dimension=2400).

4.3 Context Setting

The user engagement status greatly depends on the response from the system as well as the corresponding feedback of user. Previous studies have demonstrated the effects of contextual information in facilitating identification [3,22]. By comparing different settings of context, we are able to know which utterances are most effective for predicting user engagement. We denote five utterances in time order and define five settings of context as follows, covering different range of utterances in the dialogue:

- $user\_utt_{-1}$: previous user utterance,
- sys_utt_{-1}: previous system utterance,
- $user\_utt_0$: current user utterance,
- $user\_utt_{+1}$: next user utterance,
- sys_utt_{+1}: next system utterance.

- **CUR_UTT=**$\{user\_utt_0\}$,
- **CUR=**$\{user\_utt_0, sys\_utt_{+1}\}$,
- **NEXT=**$\{user\_utt_0, sys\_utt_{+1}, user\_utt_{+1}\}$,
- **PREV=**$\{user\_utt_{-1}, sys\_utt_{-1}, user\_utt_0\}$,
- **ALL=**$\{user\_utt_{-1}, sys\_utt_{-1}, user\_utt_0, sys\_utt_{+1}, user\_utt_{+1}\}$.

5 Results of Automatic Prediction

We conduct comprehensive experiments on four datasets to study the effects of different machine learning models, context ranges and feature settings. Specifically, we train and evaluate all models on each dataset using 10-fold cross-validation: 80%/10%/10% for training/validation/testing respectively. In order to perform significance tests on the relatively small datasets, we repeat the cross-validation five times, yielding 50 random splits and corresponding results. Unless otherwise stated, we report unweighted macro-average scores of 50 experiments on the testset. We apply two-sided paired T-test to examine the significance of changes. Besides, we also utilize the Bonferroni correction for T-test [2,37] to counteract the risk of using overlapping data partitions

5.1 Comparison of Models

We compare the performance of different models to get a general idea. We run experiments with the context range of **ALL** to include as many features as we can. All three classic classifiers are trained with *N-grams* features as well as similarity features. We report accuracy and F1-score, common metrics for classification tasks, of each model with optimal hyperparameters after a simple grid search, in Table 4.

Two simple baseline models are compared here, outputting the major class in the training set (Majority) or a random class uniformly (Random). Both simple baselines work poorly, and the F1-score of Majority is even lower due to the very skewed class distribution. The primary models perform fairly well. The two CNN models, without any human-designed feature, outperform all the other models in the current setting. The benefit of adopting pre-trained word vector is slight but significant ($p_{value} < 0.01$).

It shows comparative performance among three classic models. The SVM performs the best, but its advantage over LR is marginal ($p_{value} > 0.05$). Thus for the rest of this study, we only discuss the results of Logistic Regression, due to its advantage on interpretability of feature importance. Specifically, we use the LR with the L1 regularization ($\lambda = 1.0$), which performs robustly across different features and datasets.

Table 4. The averaged performance of user engagement classification with different models on four datasets (context=**All**, with similarity features). The underline indicates the maximum value in each column.

Model	Accuracy	F1-score
Majority	0.6020	0.1858
Random	0.2503	0.2029
SVM	0.8410	0.6440
LR	0.8398	0.6413
RF	<u>0.8415</u>	0.6192
CNN.Rand	0.8287	0.6549
CNN.MultiCh	0.8367	<u>0.6674</u>

Table 5. The comparison of user engagement classification without and with similarity features (context=**NEXT**). †/‡ indicates a significant change at p<0.05/p<0.01 between results with and without similarity features. The bold font/underline indicates the maximum value in the respective row/column.

Model	w\o Similarity	w\Similarity
Basic	0.3836	**0.4105** (+2.69%)
Phrasal	0.5913	**0.6316** (+4.03%)†‡
Syntactic	0.6078	**0.6280** (+2.02%)†‡
N-grams	0.6113	**<u>0.6573</u>** (+4.60%)†‡
Topic Model	0.5803	**0.6346** (+5.43%)†‡
Word2Vec	<u>0.6162</u>	**0.6521** (+3.59%)†‡
Doc2Vec	0.5858	**0.5968** (+1.10%)
Skip-thought	0.6063	**0.6216** (+1.53%)

Table 6. The performance (F1-score) comparison of user engagement classification with different context settings (without similarity features). †/‡ indicates a statistical significant difference at p<0.05/p<0.01 between **CUR_UTT** and **PREV** or between **NEXT** and **ALL**.

Model	CUR_UTT	CUR	NEXT	PREV	ALL
Basic	0.3425	0.3503	0.3836	0.3501†‡	**0.3963**†‡
Phrasal	0.3679	0.5521	**0.5913**	0.3709	0.5661†‡
Syntactic	0.3485	0.5530	**0.6078**	0.3671†‡	0.5867†‡
N-grams	0.3839	0.5694	**0.6113**	0.3788	0.5984†‡
Topic model	0.2982	0.5255	0.5803	0.3464†‡	**0.5829**
Word2Vec	0.3704	0.5723	**0.6162**	0.3827†‡	0.6032†‡
Doc2Vec	0.3427	0.5379	**0.5858**	0.3722†‡	0.5740†‡
Skip-thought	0.3648	0.5545	**0.6063**	0.3692	0.6008†
CNN.Rand	0.4252	0.5862	**0.6647**	0.4153	0.6549†
CNN.MultiCh	0.4207	0.5829	**0.6685**	0.4288	0.6674

5.2 Comparison of Context Settings

In this subsection, we investigate what context are most important for detecting user engagement status. We list the performance comparison with five context settings in Table 6. Note that, since there is no similarity feature for **CUR_UTT** and **CUR**, we exclude all similarity features for these experiments for a fair comparison.

Firstly, we see that, the score difference is consistent across different context settings, indicating that the context is a significant factor in engagement status prediction. **CUR_UTT** performs the worst among the five settings, since it includes only the content of the current user utterance and it provides very limited information. As for **CUR**, with one system utterance, the performance is remarkably better than the **CUR_UTT**. Furthermore, with the evident feedback from user ($user\_utt_{+1}$), **NEXT** performs generally the best among all context settings. This result conveys a clear message that, the following utterances from both system and user are critical in determining whether the next system response is relevant or not and whether the user is satisfied or not. As for **PREV** and **ALL**, which include the historical information of user requests, the performances are generally no better than the **CUR_UTT** and **NEXT** respectively. But this gap is smaller on distributed representations and models, especially for CNN. We speculate this is because most user requests can be satisfied within a few turns and do not require much historical information, thus the features from previous utterances rarely have an effect and even become detrimental.

5.3 Effects of Similarity Features

Based on the comparison of context settings, here we focus on analyzing the models with **NEXT** setting. We show the performances of Logistic Regression with

and without similarity features in Table 5. By adding similarity features, which are just one or two additional features, the scores on different feature groups increase significantly. The similarity features are devised to facilitate detecting the reformulated utterances, and we observe that the average improvement on the *Reformulation* (8.06%) is much more salient than other three classes (2.99%, 1.74% and 0.51%). Feature importance analysis based on one-way ANOVA shows that similarities on *N-gram*, *LDA* and *Phrasal* features are most significant, which is consistent with the improvement in Table 5.

5.4 Analysis on Feature Groups

Furthermore, we apply another two techniques to explore better model performance: feature combination and feature selection. On one hand, the first four feature groups are discrete and capture various local linguistic information, while the rest four groups give continuous representations with regard to the whole utterance. Thus we consider combining these two sets of features and expect further improvement with the advantages of both. On the other hand, feature selection has been proved helpful in reducing noisy features. Here we apply *Chi-square statistic* to discrete feature groups and *Principal Component Analysis (PCA)* to continuous feature groups. We report the best performance of each setting in Table 7, after a simple grid search of hyperparameters.

Table 7. Scores of user engagement prediction with different features (context=**NEXT**, with similarity features). The right part presents F1-score of best models on each dataset. The underline indicates the best score in each column. The bold indicates the better score between models with and without feature selection. †/‡ indicates a statistical significant difference at $p < 0.05$/$p < 0.01$.

Model	w\o FeatSelect	w\FeatSelect		DSTC2	DSTC3	YCap	GHome
	w\Sim	w\o Sim	w\Sim				
(a) Basic	0.4105	–	0.4105	0.5411	0.5044	0.3079	0.2886
(b) Phrasal	0.6316	–	**0.6318**	0.6470	0.6703	0.6593	0.5508
(c) Syntactic	0.6280	–	**0.6402**†‡	0.6567	0.6469	0.7005	0.5566
(d) N-grams	0.6573	–	**0.6770**†‡	0.7078	0.6905	0.6851	0.6248
(e) Topic model	0.6346	–	**0.6358**	0.6774	0.6384	0.6397	0.5877
(f) Word2Vec	0.6521	–	**0.6523**	0.6919	0.6919	0.6209	0.6043
(g) Doc2Vec	0.5968	—	**0.5969**	0.6325	0.6335	0.5730	0.5486
(h) Skip-thought	0.6216	–	**0.6216**	0.6654	0.6414	0.6020	0.5775
(i) (a) + (b) + (c) + (d)	0.6694	0.6511	**0.7085**†‡	0.7360	0.7151	0.7218	0.6613
(j) + Topic Model	0.6720	0.6617	**0.7152**†‡	0.7438	0.7161	0.7314	0.6699
(k) + Word2Vec	0.6790	0.6617	**0.7135**†‡	0.7514	0.7194	0.7180	0.6651
(l) + Doc2Vec	0.6713	0.6631	**0.7100**†‡	0.7390	0.7149	0.7269	0.6592
(m) + Skip-thought	0.6747	0.6666	**0.7124**†‡	0.7412	0.7181	0.7209	0.6696
(n) All	0.6825	0.6589	**0.7140**†‡	0.7490	0.7213	0.7202	0.6655
(o) CNN.Rand	0.6647	–	–	0.6798	0.6669	0.6943	0.6176
(p) CNN.MultiCh	0.6685	–	–	0.6880	0.6612	0.7054	0.6196

Overall, we observe that most models with feature selection outperform the original ones significantly. The feature selection works more significantly on groups having a large number of features such as *N-grams*, *Syntactic* and combined feature groups, indicating that only a small proportion of discrete features is actually in effect. Also the performances on combined feature groups (row **i** to **n**) are much better than on any of individual groups. But we observe that the continuous representations (**j-n**) contribute marginally on the top of the combined discrete features (**i**).

With the help of feature combination and selection, the Logistic Regression outpaces the previous best model CNN by a large margin. But if we exclude the similarity features (3rd column), we find that CNN still works on a par with the best LR models. Since the CNN models do not take any explicit input about similarity, the best LR models with similarity features beat CNN soundly. In order to let the CNN be aware of the user reformulation, we think it might be helpful to leverage a submodule for similarity calculation: train the submodule separately in a way like paraphrase identification [44], and take the similarity vector as additional input for classification.

Table 7 also presents detailed scores on each dataset after feature selection. One trend emerging among most LR results is that, the scores decrease gradually from **DSTC2** to **GHome**, implying the difficulty of each dataset. *LR+Basic* works well on **DSTC2** and **DSTC3** but poorly on the other two datasets. As we know, the *command_word* in *Basic* covers the most common user commands, and therefore it performs adequately in simple dialogues. But in more complicated cases, general words or linguistic components from both user and system sides become necessary, such as confirmations (ok, sure, yeah, etc.), success and failure signals (discard, sorry, don't understand, etc.), function-related words, and they are captured in different feature groups.

5.5 Analysis on Failure Cases

To understand better what shortcomings our models suffer from, we manually examine 50 random wrongly-predicted examples from **GHome** dataset and try to understand the reasons behind: *Reformulation* - 22 (examples), *Abandonment* - 21, *Fulfillment* - 4, *Continuation* - 3. The highly skewed class distribution might be one major reason. The model is trained with very few examples of *Reformulation* and *Abandonment*, therefore it is more prone to make more mistakes on them. We also notice some issues that are general to all dialogue related tasks, which might be difficult to overcome with NLP techniques used in this study: (1) A common error (16 times) is that the model cannot distinguish whether a system response is relevant to a user's request or not. Our models can only determine the relevance by feature matching instead of understanding the actual semantics, particularly when the user request is long or task-general. (2) 15 examples that require considering contextual and historical information. For example, a user asks Google Home to "Turn the Christmas tree off" and "Turn it on", our model does not recognize "it" refers to the previous "Christmas tree". Another long-dependency case is, the system confirms a similar question after a few turns, which should be treated as *Reformulation*, but

this can be hardly addressed by current models. (3) The third common mistake is specific to *Reformulation*, which occurs 9 times. On one hand, a user may paraphrase an utterance in a different way to help the system understand, such as from "I want the stair lights" to "turn on the stair lights". On the other hand, a user can also issue two apparently similar but different requests, say "how skinny is my husband" and "how old is my husband". A more powerful semantic encoder [7] might be helpful in this case.

6 Conclusion and Future Work

In a preliminary effort to solve the challenging problem of online evaluation for large-scale intelligent assistants, we provide a practicable solution, by converting the problem into a more tractable classification task and automating it with various machine learning methods. We admit there is still a long way to improve our model to work well in real environments. Also, more research is in urgent need to bridge the gap between utterance-level user engagement status and task-level user experience. Thus, for future research, we will first apply online A/B testing to validate whether any of proposed utterance-level user engagement status and metrics correlates well with the real long-term success. We believe with insights from these studies, we can understand user experience with intelligent assistants better and design better evaluation methods.

References

1. Aliannejadi, M., Zamani, H., Crestani, F., Croft, W.B.: Asking clarifying questions in open-domain information-seeking conversations. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 475–484 (2019)
2. Armstrong, R.A.: When to use the bonferroni correction. Ophthalmic Physiol. Optics **34**(5), 502–508 (2014)
3. Bangalore, S., Di Fabbrizio, G., Stent, A.: Learning the structure of task-driven human-human dialogs. IEEE Trans. Audio, Speech Lang. Process. **16**(7), 1249–1259 (2008)
4. Chowdhury, S.A., Stepanov, E.A., Riccardi, G.: Predicting user satisfaction from turn-taking in spoken conversations. Interspeech **2016**, 2910–2914 (2016)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
6. Deng, A., Shi, X.: Data-driven metric development for online controlled experiments: seven lessons learned. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 77–86 (2016)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), pp. 4171–4186 (2019)

8. Diriye, A., White, R., Buscher, G., Dumais, S.: Leaving so soon?: understanding and predicting web search abandonment rationales. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1025–1034. ACM (2012)
9. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 13–20 (2010)
10. Griol, D., Callejas, Z.: A neural network approach to intention modeling for user-adapted conversational agents. Comput. Intell. Neurosci. **2016**, 44 (2016)
11. Hara, S., Kitaoka, N., Takeda, K.: Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010) (2010)
12. Hashemi, S.H., Williams, K., El Kholy, A., Zitouni, I., Crook, P.A.: Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1183–1192 (2018)
13. Hassan, A., Jones, R., Klinkner, K.L.: Beyond DCG: user behavior as a predictor of a successful search. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 221–230. ACM (2010)
14. Hassan, A., Shi, X., Craswell, N., Ramsey, B.: Beyond clicks: query reformulation as a predictor of search satisfaction. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 2019–2028. ACM (2013)
15. Hassan, A., Song, Y., He, L.W.: A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 125–134. ACM (2011)
16. Henderson, M., Thomson, B., Williams, J.D.: The second dialog state tracking challenge. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 263–272 (2014)
17. Henderson, M., Thomson, B., Williams, J.D.: The third dialog state tracking challenge. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 324–329. IEEE (2014)
18. Higashinaka, R., Funakoshi, K., Kobayashi, Y., Inaba, M.: The dialogue breakdown detection challenge: task description, datasets, and evaluation metrics. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 3146–3150 (2016)
19. Hill, F., Cho, K., Korhonen, A.: Learning distributed representations of sentences from unlabelled data. In: 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016, pp. 1367–1377. Association for Computational Linguistics (ACL) (2016)
20. Jiang, J.E.A.: Automatic online evaluation of intelligent assistants. In: Proceedings of the 24th WWW, pp. 506–516. International World Wide Web Conferences Steering Committee (2015)
21. Kamm, C.: User interfaces for voice applications. Proc. Natl. Acad. Sci. **92**(22), 10031–10037 (1995)

22. Kim, S.N., Cavedon, L., Baldwin, T.: Classifying dialogue acts in one-on-one live chats. In: Proceedings of the 2010 Conference on EMNLP, pp. 862–871. Association for Computational Linguistics (2010)
23. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
24. Kiros, R.E.A.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3294–3302 (2015)
25. Kiseleva, J., Williams, K., Hassan Awadallah, A., Crook, A.C., Zitouni, I., Anastasakos, T.: Predicting user satisfaction with intelligent assistants. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 45–54. ACM (2016)
26. Krahmer, E., Swerts, M., Theune, M., Weegels, M.: Error detection in spoken human-machine interaction. Int. J. Speech Technol. **4**(1), 19–30 (2001)
27. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1188–1196 (2014)
28. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 110–119 (2016)
29. Liu, C.W., Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J.: How not to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2122–2132 (2016)
30. Lowe, R., Noseworthy, M., Serban, I.V., Angelard-Gontier, N., Bengio, Y., Pineau, J.: Towards an automatic turing test: learning to evaluate dialogue responses. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1116–1126 (2017)
31. Meena, R., Lopes, J., Skantze, G., Gustafson, J.: Automatic detection of miscommunication in spoken dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 354–363 (2015)
32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
33. Ohtake, K.: Unsupervised approach for dialogue act classification. In: Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation, pp. 445–451 (2008)
34. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
35. Polifroni, J., Hirschman, L., Seneff, S., Zue, V.: Experiments in evaluating interactive spoken language systems. In: Proceedings of the workshop on Speech and Natural Language, pp. 28–33. Association for Computational Linguistics (1992)
36. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 583–593 (2011)
37. Salzberg, S.L.: On comparing classifiers: pitfalls to avoid and a recommended approach. Data Mining Knowl. Discov. **1**(3), 317–328 (1997)

38. Shriberg, E., Wade, E., Price, P.: Human-machine problem solving using spoken language systems (sls): factors affecting performance and user satisfaction. In: Proceedings of the Workshop on Speech and Natural Language, pp. 49–54. Association for Computational Linguistics (1992)
39. Sordoni, A., et al.: A neural network approach to context-sensitive generation of conversational responses. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 196–205 (2015)
40. Vinyals, O., Le, Q.V.: A neural conversational model. In: ICML Deep Learning Workshop (2015). http://arxiv.org/pdf/1506.05869v3.pdf
41. Walker, M.A., Litman, D.J., Kamm, C.A., Abella, A.: Paradise: a framework for evaluating spoken dialogue agents. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 271–280 (1997)
42. Yang, Z., Li, B., Zhu, Y., King, I., Levow, G., Meng, H.: Collaborative filtering model for user satisfaction prediction in spoken dialog system evaluation. In: Spoken Language Technology Workshop (SLT), 2010 IEEE, pp. 472–477. IEEE (2010)
43. Yi, X., Hong, L., Zhong, E., Liu, N.N., Rajan, S.: Beyond clicks: dwell time for personalization. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 113–120. ACM (2014)
44. Yin, W., Schütze, H.: Convolutional neural network for paraphrase identification. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 901–911 (2015)