Measuring User Satisfaction on Smart Speaker Intelligent Assistants Using Intent Sensitive Query Embeddings

Seyyed Hadi Hashemi*
University of Amsterdam
Amsterdam, The Netherlands
hashemi@uva.nl

Kyle Williams
Microsoft
Redmond, USA
Kyle.Williams@microsoft.com

Ahmed El Kholy
Microsoft
Redmond, USA
Ahmed.ElKholy@microsoft.com

Imed Zitouni
Microsoft
Redmond, USA
izitouni@microsoft.com

Paul A. Crook†
Facebook
Seattle, USA
pacrook@fb.com

ABSTRACT

Intelligent assistants are increasingly being used on smart speaker devices, such as Amazon Echo, Google Home, Apple Homepod, and Harmon Kardon Invoke with Cortana. Typically, user satisfaction measurement relies on user interaction signals, such as clicks and scroll movements, in order to determine if a user was satisfied. However, these signals do not exist for smart speakers, which creates a challenge for user satisfaction evaluation on these devices. In this paper, we propose a new signal, user intent, as a means to measure user satisfaction. We propose to use this signal to model user satisfaction in two ways: 1) by developing intent sensitive word embeddings and then using sequences of these intent sensitive query representations to measure user satisfaction; 2) by representing a user's interactions with a smart speaker as a sequence of user intents and thus using this sequence to identify user satisfaction. Our experimental results indicate that our proposed user satisfaction models based on the intent-sensitive query representations have statistically significant improvements over several baselines in terms of common classification evaluation metrics. In particular, our proposed task satisfaction prediction model based on intent-sensitive word embeddings has a 11.81% improvement over a generative model baseline and 6.63% improvement over a user satisfaction prediction model based on Skip-gram word embeddings in terms of the F1 metric. Our findings have implications for the evaluation of Intelligent Assistant systems.

ACM Reference Format:
Seyyed Hadi Hashemi, Kyle Williams, Ahmed El Kholy, Imed Zitouni, and Paul A. Crook. 2018. Measuring User Satisfaction on Smart Speaker Intelligent, Assistants Using Intent Sensitive Query Embeddings. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271802

Keywords: personal intelligent assistants; user satisfaction; query embeddings; query intent

1 INTRODUCTION

There is a growing interests in integrating intelligent assistants (IAs) such as Apple Siri, Google Now, Microsoft Cortana and Amazon Alexa in different devices [6]. This has led to the creation of smart devices, such as smart phones and smart speakers. Each of these smart devices provides device specific means of interaction. For example, smart speakers do not include any screen interface and therefore users interact with them via voice. This is in contrast to IAs on Desktop computers and mobile phones, which have clicks and gestures as user interactions [38, 48]. Since user behavior differs on these different platforms and in different contexts [16, 17, 26, 47] there may be a need to develop different means of evaluation for different platforms.

Smart speaker devices with integrated IAs such as Amazon Echo, Google Home, Apple Homepod, and Harmon Kardon Invoke with Cortana have become increasingly popular in recent years. For instance, one study found that there was a 128.9% increase in the number of smart speaker users in the United States in 2017 compared to 2016[1]. Therefore, measuring the effectiveness of IAs on these popular smart devices is becoming increasingly important.

An emerging metric for evaluating Information Retrieval (IR) and IA systems is user satisfaction, which is often based on user interaction data [1, 12, 19, 20, 25, 29, 31, 33, 34]. User satisfaction is a subjective measure of a user's experience with an information system, which indicates to some extent if the user's desire or goal is fulfilled [27]. User satisfaction evaluation in IAs on mobile phones and Desktop computers has previously been studied [24, 28, 33, 34, 38, 48, 49]; however, to our knowledge, there have been no studies investigating user satisfaction and IA effectiveness for smart speakers. In this paper, we use the phrase smart speaker to refer to a wireless speaker device that integrates an IA. For the purpose of this study, we focus on devices that have no screen and where the only method of communicating with the device is via voice.

Many implicit signals have been studied for measuring user satisfaction in Web search or for IAs on desktops and mobile devices. Some examples of signals include: clicks followed by a long dwell-time [13, 22, 25, 30, 31], mouse movements [38], touch gestures [33, 48], and browser view-port interactivity [35]. However, besides user queries, none of these implicit user satisfaction signals are available for smart speakers due to voice being the only method of interaction. Therefore, evaluating user satisfaction with IAs on smart speakers presents a new challenge, which is finding an effective implicit user satisfaction feedback signal.

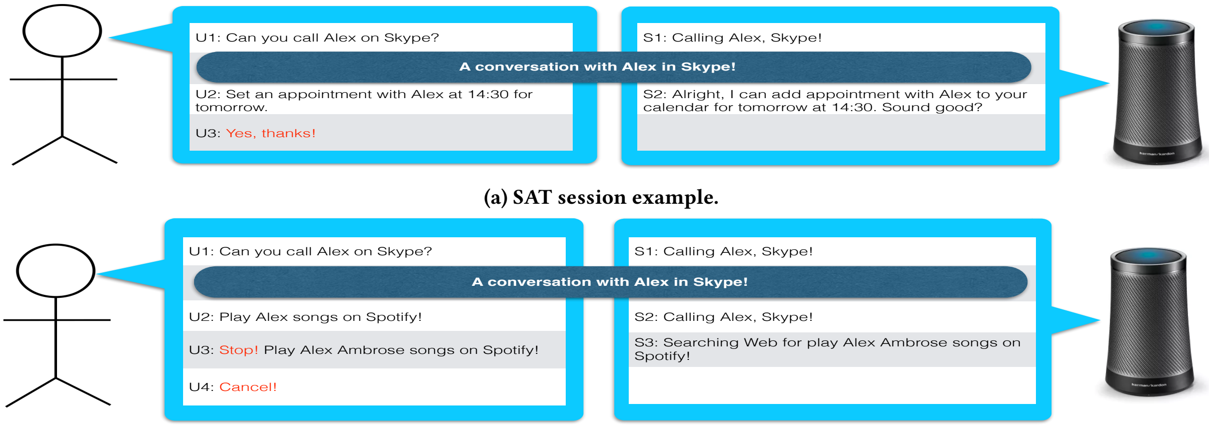
[*] Work done while interning at Microsoft.

[†] Work done while at Microsoft.

[1] https://www.emarketer.com/Article/Alexa-Say-What-Voice-Enabled-Speaker-Usage-Grow-Nearly-130-This-Year/1015812

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271802



U1: Can you call Alex on Skype?

S1: Calling Alex, Skype!

A conversation with Alex in Skype!

U2: Set an appointment with Alex at 14:30 for tomorrow.

S2: Alright, I can add appointment with Alex to your calendar for tomorrow at 14:30. Sound good?

U3: Yes, thanks!

(a) SAT session example.

U1: Can you call Alex on Skype?

S1: Calling Alex, Skype!

A conversation with Alex in Skype!

U2: Play Alex songs on Spotify!

S2: Calling Alex, Skype!

U3: Stop! Play Alex Ambrose songs on Spotify!

S3: Searching Web for play Alex Ambrose songs on Spotify!

U4: Cancel!

(b) DSAT session example.

Figure 1: SAT/DSAT user session example in smart speaker IA.

Since queries are the only means by which users interact with smart speakers, they represent a natural starting point for measuring user satisfaction. In fact, the reformulation of a query is a well known signal for user dissatisfaction [21]. In this sense, one can think of queries as not only representing a user's information need, but also as providing an implicit feedback signal similar to the case of a click in Web search or a touch on mobile phones. In this paper, a query is considered more generic than the typical query used in IR literature. Specifically, in this study, queries could be typical Web queries or command-like queries (e.g., device control commands). Figure 1a shows an example of a SAT session, in which a user tries to complete a "setting an appointment" task after making a call to Alex. We hypothesize that the sequence of user queries can be a beneficial implicit user satisfaction (SAT) signal. Specifically, issuing the query "Yes, thanks" after the query "Set an appointment with Alex at 14:30 for tomorrow." might be an indicator of user satisfaction. However, the user saying "Cancel" after the query might be an indicator of user dissatisfaction (DSAT).

Furthermore, Figure 1b indicates an example of a DSAT session. In this session, a user asks the smart speaker to play a song by Alex on Spotify after making a call to Alex (i.e., U2). However, the IA does not detect the user intent properly as it wrongly determines that the context is still making a call, which leads to an undesirable system response. As the system response is not satisfactory, the user tries to stop the IA from giving a wrong response and issues a similar query again (i.e., U3). The system responds by searching the Web for the user query, which is not satisfactory for the user as she has stopped the smart speaker by saying "Cancel". As can be seen by these examples, a sequence of queries can lead to an implicit feedback signal of user satisfaction. The question then arises on how one should use these queries to measure user satisfaction and, as the specific focus of this paper, how one can find effective query representations for measuring user satisfaction.

In this paper, we hypothesize that the intent of a user query can function as an implicit signal for measuring user satisfaction, where intent refers to the meaning of the query [43]. For instance, the query "Set an appointment with Alex at 14:30 for tomorrow." has "create_calendar_entry" intent. If this is followed by a "confirm" intent, then we may conclude the user was satisfied. In contrast, if it was followed by a "reject" intent then we may conclude the user was dissatisfied. Based on this intuition, we propose to measure user satisfaction based on representations of user queries that are intent sensitive. We propose to do this in two ways. In the first way, we define Intent Sensitive Word Embeddings (ISWEs), which are word embeddings that not only represent the semantics of words, but also semantics of the intents associated with words. For example, although the queries "Yes, thanks" and "Cancel" occur in similar word contexts, (e.g., "Set an appointment with Alex at 14:30" → "Yes, thanks!" and "Set an appointment with Alex at 14:30" → "Cancel"), they have very different intents, i.e., "confirm" for the former and "reject" for the latter. Our proposed methods for producing ISWEs scatters these words with different intents in the representation space. We use this sequence of ISWEs to measure user satisfaction based on a series of user queries.

In the second approach, we consider each query as a single unit having a single intent and train query representations based on a sequence of query intents. For example, "play some jazz music" is a single query having a single intent of "play_music", and "Set an appointment with Alex at 14:30" → "Yes" is an example of a task containing two queries with intents "create_calendar_entry" and "confirm". Therefore, these would represent sequences of length 1 and 2, respectively, and we use these sequences of intents as input to our proposed user satisfaction prediction model. This approach differs from our other proposed approach since in ISWE we use intents to derive intent sensitive word embeddings. In this approach we forgo the words and only focus on the intent of the entire query.

In this paper, our main aim is to study the question: *How to evaluate user satisfaction in Intelligent Assistants based on user queries?* Specifically, we answer the following research questions:

(1) *How to model intent-sensitive query representations for user satisfaction prediction?*

(2) *How effective is the proposed intent-sensitive user satisfaction model in evaluating intelligent assistants on smart speakers?*

Our contributions include: (1) a user satisfaction prediction model that predicts user satisfaction based on just user query sequences; (2) proposing a novel intent-sensitive word embedding (ISWE) that can capture query term intents by learning word representations based on both word neighbor context and query intent; and (3) an unsupervised intent embedding approach based on the Skip-gram model that learns intent representations for each query.

In making these contributions, the rest of this paper is organized as follows. In Section 2, we review related work . Section 3 is devoted to task and user satisfaction definitions. The proposed user satisfaction prediction model and intent-sensitive query representation learning methods are described in Section 4. Section 5 presents experimental results of our study. Finally, we present our conclusions and discuss future work in Section 6.

2 RELATED WORK

Related work falls into two categories: we first review user satisfaction in search and intelligent assistants, and then we discuss related work on word embeddings.

2.1 User Satisfaction

Online evaluation have been widely used to control and improve IR system effectiveness [4, 8, 10, 46]. An emerging metric for evaluating Web search engines is user satisfaction based on implicit signals from user interactions [1, 12, 19, 20, 25, 29, 31, 33, 34]. User satisfaction in search is a subjective measure of a user's search experience, which is addressed by the extent to which a user's specified desire or goal is fulfilled [27]. User satisfaction is different from traditional relevance measures in IR such as MAP and Precision, which are based on relevance of the retrieved results for a given query. In user satisfaction, user experience and their success in fulfilling a goal plays a major role, which has been addressed based on user interaction signals in web search [19, 21, 29, 31] and intelligent assistants [33, 34, 38, 48, 49].

One of the common signals that has been used for user satisfaction prediction is a click followed by a long dwell-time [13, 22, 25, 30, 31]. Hassan et al. [21] propose query reformulation as a signal of user dissatisfaction and they show that incorporating query features and query reformulation in user satisfaction prediction outperforms an approach based on click features alone.

Session and SERP features such as time to the first click and average number of clicks per query have been also studied in personalized and customized search satisfaction prediction [22]. Furthermore, gesture features, such as reading time and touch actions, have been used in search satisfaction prediction [33] and good abandonment detection [48] in mobile web search. In addition, tracking the browser viewport on mobile devices has been also studied as an implicit signal for user search satisfaction [35].

To model user satisfaction, Hassan et al. [20] model the search process as a sequence of actions, such as queries and clicks, and built two Markov models for identifying satisfactory and dissatisfactory search sequences. They further shows that using a semi-supervised search success prediction approach based on sequence of actions can lead to an improvement over the supervised approach [19]. More recently, Mehrotra et al. [38] proposed user satisfaction prediction in Desktop intelligent assistants based on fine-grained actions, such as mouse movements.

Our work is different from all the above user satisfaction prediction models in two aspects. First, we focus on user satisfaction prediction with IA on smart speakers, in which the only means of interaction is via voice queries. In fact, none of the past works are applicable in user satisfaction on smart speakers, except the query reformulation proposed in [21]. Our proposed approach is different from the query reformulation as we use a sequence of query terms to predict user satisfaction. Furthermore, we propose user intent as a new signal to measure user satisfaction.

2.2 Word Embeddings

Recently, continuous word embeddings have gained popularity in different IR tasks, such as query and document language models [14, 51], neural ranking models [7, 53, 54], and query expansion [3, 9, 52]. In particular, Zamani and Croft [52] propose a theoretical framework for query embedding vectors representation based on individual vocabulary term embeddings. Furthermore, they propose using word embeddings to weight terms that do not occur in the query, yet are semantically related to the query terms in the query language model. More recently, Ai et al. [2] proposed a hierarchical embedding model that jointly learns distributed representations for query, product and users in a personalized product search.

Zamani and Croft [53] recently showed that the linear context is not sufficient for learning an effective word embeddings for IR tasks, and they propose learning word representations based on query-document relevance information. In addition, Rekabsaz et al. [44] propose post filtering of related terms by global context relatedness measures to avoid topic shifting in retrieval models. Furthermore, Mehrotra and Yilmaz [37] propose learning query representations based on task context in search logs.

Although, incorporating additional signals to improve word embeddings in IR is very new, there have been plenty of research in NLP to improve word representations by using metadata [55], semantic lexicons [42], syntactic word relations [36] and document topics [11]. Our work is different from the above as we propose a novel Intent Sensitive Word Embedding (ISWE) method that can leverage information from a query's intent to improve query term representations. We are the first who modify the Skip-gram model [39] to capture query term intents and use them as input to a user satisfaction prediction model.

3 TASK SATISFACTION

In this section, we first define tasks and sessions as they apply to IAs, and then we define task satisfaction in IAs. In IAs, users usually take a sequence of steps to achieve a goal to solve one or more tasks [33]. Since IAs can keep context from previous queries, this allows for task chaining where the context of one task can be used as input to the next. Considering the multi-task nature of the users' behaviors in IA, we follow the task and session definitions as proposed in [18]:

- **A Task** is a single information need that can be satisfied by at least one query and one IA generated response.
- **A Session** is a short period of contiguous time spent to fulfill one or multiple tasks.

Given the definitions of tasks and session, we define task satisfaction as follows:

- **Task satisfaction** is how successful a user is in completing a single information need using at least one query and receiving at least one IA generated response.

IA generated responses are not always in the form of replying to a user query in a dialogue manner. For some queries such as “Stop”, a proper IA response could be simply stopping whatever the IA was doing. Table 1 shows an example of a user's task satisfaction in an IA session. In this example, the user is performing four tasks, including: reviewing her calendar; sending a text; calling on Skype; and setting an appointment. These tasks are part of a session, in which the user is organizing a meeting with Alex. Tasks 1 and 2 in Table 1 show examples of satisfactory (SAT) and a dissatisfactory (DSAT) tasks, respectively.

To summarize, in this section, we have defined task and sessions in IAs, and then we have defined task satisfactions in IAs. In the next section, we detail the task satisfaction prediction problem in smart speaker IAs and describe our proposed task satisfaction prediction model.

Table 1: An example of a user's task satisfaction in an IA on a smart speaker.

User Utterance and System Response	User Satisfaction
Task 1: Calendar review	
U1: What does my day look like tomorrow? S1: You don't have anything scheduled for tomorrow.	SAT
Task 2: Sending a text	
U2: Text Alex and ask if he is available for a short meeting tomorrow S2: Sorry, I can't send messages here. Try the app on your phone or PC.	DSAT
Task 3: Calling on Skype	
U3: Can you call Alex on Skype? S3: Calling Alex, skype	SAT
Task 4: setting an appointment	
U4: Set an appointment with Alex at 14:30 for tomorrow. S4: Alright, I can add appointment with Alex to your calendar for tomorrow at 14:30. Sound good? U5: Yes, thanks!	SAT

4 TASK SATISFACTION PREDICTION

This section first presents user satisfaction prediction problem based on a sequence of user queries. We then detail our proposed user satisfaction prediction and query representation learning models .

4.1 Satisfaction Classification Model

The task of user satisfaction prediction based on a sequence of user queries can be regarded as a sequence classification problem. To be more specific, a user starts querying the IA at time stamp t_0 and can keep querying the IA up to time stamp t_n in a task or a session. Therefore, we can represent a user's set of interactions with an IA on a smart speaker as a sequence of queries $q_{t_0}, q_{t_1}, q_{t_2}, ..., q_{t_{n-1}}, q_{t_n}$, where q_t is a query q at time stamp t. Given a sequence s of queries $q_t \in Q$, the task is to predict whether the sequence of queries leads to a satisfactory (SAT) or a dissatisfactory (DSAT) experience in a task. In particular, using a variable $c \in \{0, 1\}$, the goal is to find the most likely class c, given a sequence s.

Considering the sequential nature of user queries and its variable length in accomplishing tasks in IA, we propose to use a Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) to model user satisfaction because of the following reasons: 1) LSTMs have been shown to be effective in different sequence classification problems such as text classification [15], sentence similarity [40] and satisfaction prediction in the case of good abandonment [50]. 2) LSTMs are more effective than standard RNNs in their ability to model long time dependencies.

The LSTM updates a hidden layer representation sequentially using time steps relying on four components: 1) a memory state c_t, 2) an input gate i_t, 3) a forget gate f_t, and 4) an output gate o_t. The input and forget gates control what gets stored in memory based on each input and the current state. The output gate controls how the memory state impacts other units. In an LSTM, updates at each time step t are as follows:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ \widetilde{c_t} &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= i_t \odot \widetilde{c_t} + f_t \odot c_{t-1} \\ h_t &= o_t \odot tanh(c_t), \end{aligned}$$

where x_t is an input at time step t, $\widetilde{c_t}$ is the candidate value for the state of the memory cell, and h_t is the output of the unit. The W_i, W_f, W_o, and W_c are the weight matrices for the current input. The U_i, U_f, U_o, and U_c are the weight matrices for the previous output, and b_i, b_f, b_o and b_c are bias vectors.

In this study, we use the LSTM model defined above to model the sequence of queries issued by a user to accomplish a task. The input x_t in our model is an intent-embedding of a user's queries. In Section 4.2, we describe in detail how we acquire these embeddings.

In our model, the embedding layer is connected to a block of LSTM units. To prevent over-fitting problem, we have used a dropout at the LSTM layer, which randomly drops units and their connections to avoid unit co-adapting [23, 45]. Following previous work, we have used $p = 0.5$ in our dropout network as this value has been reported as a close to optimal value for a wide range of networks in different applications [45]. The output of the last time step in the LSTM feed to a standard feed-forward neural network that contains a single output neuron that uses the sigmoid activation function.

In the learning phase, the derivatives of the loss function are backpropagated through the neural network. Our neural network is trained using the stochastic gradient descent (SGD) algorithm with mini batches, which is widely used algorithm for training neural networks. In order to do hyper-parameter optimization in the learning phase, we have used random search [5], which has been reported to be as good as or better than the grid search in hyper-parameter optimization of neural networks [5].

The random search has been done using a continuous parameter space in range of $[0.0001, 0.1]$ for the learning rate. The chosen learning rates by the random search are adjusted based on the Adam optimization algorithm [32].

4.2 Query Representation Learning

We presented our LSTM-based model for user satisfaction prediction in the previous section. We mentioned that the inputs to our model were embeddings that represented the query. In this section, we describe two different representations. One representation is based on an Intent Sensitive Word Embedding (ISWE) while the other is based on unsupervised intent embeddings. Specifically, we answer the research question: *How to model intent-sensitive query representations for user satisfaction prediction?*

4.2.1 Intent-Sensitive Word Embeddings. To learn query representations based on query terms, we explored different word representation models. The word2vec Skip-gram model is one of the state of the art approaches to learn vector representations of words. Word embeddings trained using the Skip-gram model have been shown to be very useful in many tasks [9, 11]. However, in many of the previous efforts, embeddings were generated without taking into consideration the targeted task leading to generic embeddings that might not serve the task well. For example, Skip-gram model leads to word representations considering "Stop" and "Start" words as being similar. However, although "Stop" and "Start" might be similar based on linear neighbor word context in sessions (e.g., "**start** my

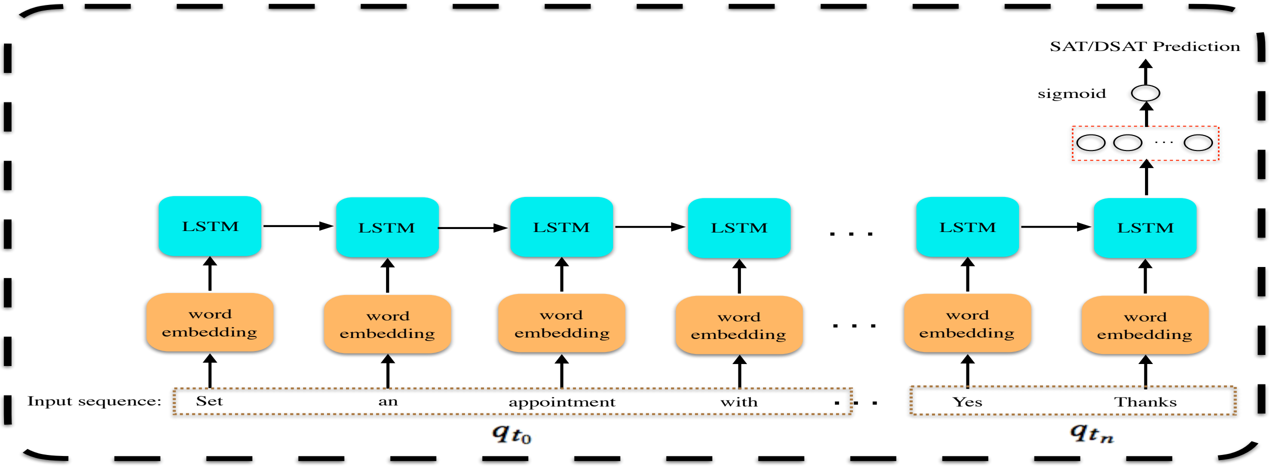


Figure 2: User satisfaction prediction model based on sequence of query terms as input.

jazz playlist on Spotify!" and "**stop** my jazz playlist on Spotify!"), they lead to different queries having completely different intents.

In the rest of this section, we explain our proposed intent sensitive word embedding approach. In our approach, we augment the standard Skip-gram model word embeddings with the query intent information to avoid learning similar representations for words who have similar linear context but different intents. This way, we generate more effective and task oriented word representations compared to the original Skip-gram word embeddings. The main idea is to add more information when the immediate linear context of the word is not very informative. We train word embeddings based on our proposed Intent-Sensitive Skip-gram model for a query $q \in Q$, having intent $\imath \in I$, and containing a sequence of words $w_1, w_2, ..., w_T$. The objective of the Intent-Sensitive word embedding Skip-gram model is to maximize the log-likelihood of context word-intent pair $w^{\imath}_{t+j}$ given the target word w_t:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-b\leq j\leq b, j\neq 0} \log p(w^{\imath}_{t+j}|w_t),$$

in which, b is the size of training context ($b = 10$ in all of our experiments), and $\imath$ is intent of the query in which the word occurred. In order to train the Intent Sensitive Word Embeddings (ISWEs), we have collected intent labels of about 900K IA queries. To collect queries intent labels, we use an in-house micro-tasking platform that outsources crowdwork to judges who regularly perform intent labeling tasks. We presented one query to the judges at a time and they select intent of the given query from a predefined set of intents. The annotations include queries having 266 unique intents like "create_calendar_entry" and "make_call" from 30 different usage domains in IA such as "calendar" and "communication".

We also use negative sampling as discussed in [39]. In negative sampling, having a dataset D of observed $(w, c^{\imath})$ pairs of word w and intent-sensitive context $c^{\imath}$, we generate the set D' including random $(w, c^{\imath})$ pairs assuming they are incorrect. The probability of a $(w, c^{\imath})$ coming from the data is denoted by $p(D = 1|w, c^{\imath})$ and $p(D = 0|w, c^{\imath}) = 1 - p(D = 1|w, c^{\imath})$ is the probability of $(w, c^{\imath})$ coming from the negative examples. Ideally, the $p(D = 1|w, c^{\imath})$ must be high for the word and context pairs observed in the data and low for the random negative samples. The negative sampling training objective is as follows:

$$argmax_{v_w, v_{c^{\imath}}}\left(\sum_{(w,c^{\imath})\in D}\log\sigma(v_{c^{\imath}}\cdot v_w) + \sum_{(w,c^{\imath})\in D'}\log\sigma(-v_{c^{\imath}}\cdot v_w)\right),$$

where $\sigma(x) = 1/(1 + e^x)$, v_w and $v_{c^{\imath}}$ are d-dimensional vectors which are model parameters to be learned using stochastic-gradient updates over the whole corpus including both observed and negative sampled word and context pairs (i.e., $D \cup D'$). In all of our experiments in this paper, we set $d = 100$. To create the negative sample, we follow Mikolov et al. [39] in creating n negative samples $(w, c^{\imath}_1)$, $(w, c^{\imath}_2)$, ..., $(w, c^{\imath}_n)$ where each $c^{\imath}_j$ sampled based on its unigram distribution raised to the 3/4 power.

According to experiments detailed in [39], n in the range of 5-20 are useful for small training datasets. Thus, following other successful Skip-gram with negative sampling experiments [36], we have chosen 15 as the negative sample size for each positive observed sample in our dataset.

The above objective optimization leads to word and intent-sensitive context pairs having similar embeddings for the pairs observed in the data while scattering negative sampled pairs. In ISWE, words appearing in a similar intent-sensitive context (i.e., context-word and query-intent) should have similar embeddings. We feed a sequence of queries, in which each query contains a sequence of ISWE query terms, to the proposed task satisfaction prediction model discussed in Section 4.1. Figure 2 shows an example of how we feed ISWE query term representations to the satisfaction prediction network. Effectiveness of the learned query representation in predicting user satisfaction is discussed in Section 5.

4.2.2 Unsupervised Intent Embedding. In this model, we choose to forgo the individual words of a query and instead choose to represent the entire query by its intent. To model intent embeddings, we propose using the Skip-gram model [39], which is common for learning word embeddings in NLP [11, 36, 41]. In our proposed Intent2Vec Skip-gram model, each intent $\imath \in I$ is associated with a vector $v_i \in R^d$, where I is the intent vocabulary, and d is the embedding dimension. In all of our experiments, we set $d = 100$. The training objective of the Intent2Vec model is to find intent representation, which are effective to predict the intent of surrounding queries in a task or a session. Formally, given a sequence of intents $\imath_1, \imath_2, ..., \imath_T$ in a session, the objective of the Intent2Vec Skip-gram

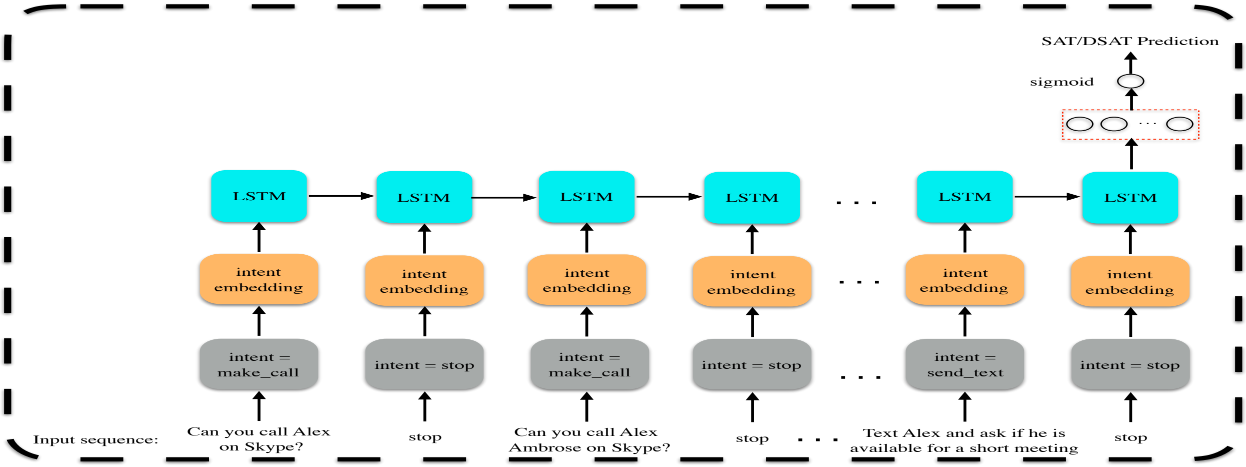


Figure 3: User satisfaction prediction model based on sequence of query intents as input.

model is to maximize the following function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-b\leq j\leq b, j\neq 0}\log p(\imath_{t+j}|\imath_t),$$

in which, b is the size of training context, which is set to $b = 10$ in all of the experiments in this paper. To train the intent embedding, we have used negative sampling as presented in Section 4.2.1 and chose 15 as the number of negative samples for each positive sample observed in our dataset.

In order to train query representations based on the Intent2Vec model, we sampled about 500K IA queries issued in a 3 month period of a commercial smart speaker usage. This dataset does not have any relation with the 900K queries dataset used for training ISWE. We use a high-quality production-level offline intent classifier to assign intents to each query. Our query sample has 181 unique intents like "check_weather" and "volume_down" from 23 different domains such as "weather" and "media-control". As previously mentioned, Intent2Vec trains intent embeddings based on neighboring query intents. Thus, to train intent embeddings, Intent2Vec requires a sequence of queries issued in each session as an input. In order to create a sequence of query intents for each session, we need to identify sessions from the raw IA log data. To do this we follow the approach of [18] to identify session boundaries, which leads to creation of about 69K sessions based on the one hour session boundary identified using the approach presented in [18]. Using sessions as input of I2V leads to training intent embeddings based on a larger context size compared to training the intent embeddings on tasks. Training the intent embeddings on sessions provides representations that considers both within task and cross-task contexts in IA sessions.

To predict task satisfaction, we feed a sequence of query intent representations to the model discussed in Section 4.1. Figure 3 shows an example of how we feed a sequence of queries to the user satisfaction prediction network. We first assign an intent to each user query using a high-quality production-level offline intent classifier. We then feed a sequence of intent embeddings query representations to the user satisfaction prediction network. In our proposed user satisfaction prediction models, we just consider users' queries as input and we do not use system responses, which are directly controlled by the IA, as input and feature to the user satisfaction prediction model. By excluding system responses, we avoid using endogenous features (i.e., features that the search engine has control over [50]) in our online user satisfaction prediction model. Evaluation of the model is available in Section 5.

In this section, we have defined the task satisfaction problem and our proposed model to address the problem. We have also described the intent-sensitive word representation learning and Intent2Vec query-intent representation learning models with an aim of learning effective query representations for task satisfaction prediction. In the next section, we present a set of experiments evaluating these models.

5 EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed models by answering the research question: *How effective is the proposed intent-sensitive user satisfaction model in evaluating intelligent assistants on smart speakers?*

5.1 User Satisfaction Judgment Crowdsourcing

This study is based on a random sample of users interaction logs with a commercial IA being used on a smart speaker during August, September and October, 2017. Our random sample includes user sessions having one to ten queries. As it is not easy to collect explicit feedback about actual users' satisfaction, crowdsourced judgments, which have been widely used to obtain labeled data for different problems including user satisfaction in IAs [38, 48], is used. To collect user satisfaction judgments, we use an in-house micro-tasking platform that outsources crowdwork to judges who regularly perform relevance judgment tasks. We removed all the personal identifiable information (PII) from the sessions before sending them for judgment.

A detailed guideline including a video explaining how to judge user satisfaction was shown to the judges. We presented a whole session to judges, and asked them to assess query-level satisfaction and session level satisfaction of a user in the given session. We also collected task identification labels for a user session, in which crowdsource workers judged whether the user was trying to fulfill the same information need as the previous query by issuing the current query. To judge user satisfaction and task identification, judges could read or listen to the user's query, read system response, look at the original timestamp of queries, and read or listen to the user's previous or future queries.

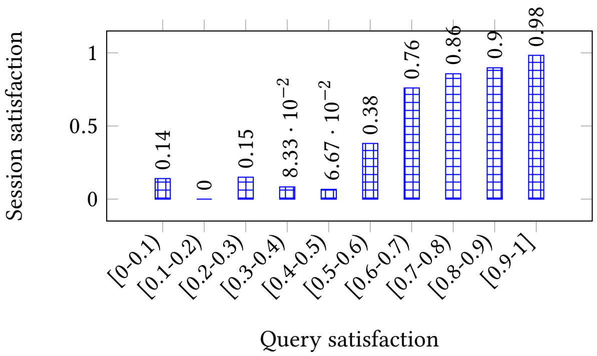


Figure 4: Impact of percentage of satisfactory queries in a session on session satisfaction. This chart is based on a crowdsourced judgments collected in Section 5.1.

In order to obtain a high-quality user satisfaction labels, at least 3 and at most 5 crowdsource workers judged each session. Qualifying tests and spam detection were used to filter out low-quality judgments. The final label used for satisfaction is based on a majority vote. We randomly sampled over 1700 user sessions in the smart speaker IA and collected user satisfaction labels for them. We measure inter-rater agreement using Fleiss' Kappa [12].

The goal of this study is to measure task satisfaction. To create tasks, we used the majority vote as task boundary labels. Specifically, we create tasks based on sequence of queries issued by a user up to a point that the task is ended using the task boundary collected labels. In our collected dataset, the crowdsourced task boundary labels led to 3105 tasks including 6920 query-level satisfaction labels. The Kappa value is 0.42 for the collected query-level user satisfaction labels and 0.72 for the collected task boundary labels.

After creating the tasks, we need to assign a label to each task. In Figure 4, we have shown the correlation between percentage of satisfactory queries in a session and the session satisfaction probability based on the crowdsourced session satisfaction collected labels. According to our observation on session-level user satisfaction in Figure 4, having from 60% to 70% user query-level satisfaction in a session leads to session-level satisfaction in 76% of the cases. Overall, having at least 60% user query-level satisfaction leads to 93% session-level satisfaction compared to just 19% user satisfaction for tasks having less than 60% user query-level satisfaction. Therefore, we have intuitively used the 60% threshold for task-level satisfaction labels, which means that if 60% or more of queries issued in a task are labeled SAT, then we label the task as SAT.

5.2 Baselines

We consider three baselines when evaluating our models.

(1) Query reformulation (QR) [21] is one of the defined baselines, which classifies a task as DSAT if the last query of a user in the task is a reformulation of the second to last query with no user interaction after the last query. Otherwise, it classifies the task as a SAT experience. We use the method of [21] to determine if a query was reformulated.

(2) The second baseline is a variant of the generative model (GM) [19], which uses a sequence of query terms to predict satisfaction. The GM is a mixture model composed of SAT and DSAT components. Give a sequence of interactions, the goal during classification is to identify whether the sequence was generated by the SAT or DSAT components of the mixture model [19]. The model was originally used to predict search success based on a sequence of actions in a session including clicks, query reformulation and queries. We use the query terms as sequence of actions in the GM as they are the only means of user interactions in smart speaker IAs.

Table 2: Nearest neighbors of three examples in different representation spaces based on cosine similarity. Both word2vec skip-gram baseline and the intent-sensitive word embeddings are trained on a same dataset.

Word	good
Word2Vec Skip-gram	nice, great, wonderful, **bad**, lovely
Intent-Sensitive WE	nice, great, wonderful, fine, decent
Word	yes
Word2Vec Skip-gram	yeah, **not**, okay, **no**, ok
Intent-Sensitive WE	yeah, yep, sounds, correct, ok
Word	cancel
Word2Vec Skip-gram	delete, remove, erase, **edit**, **set**
Intent-Sensitive WE	remove, delete, disable, dismiss, clear

(3) In order to evaluate effectiveness of our proposed ISWE in user satisfaction predictions (ISWE-LSTM), we have defined another baseline based on our proposed user satisfaction prediction model detailed in Section 4, yet using original Skip-gram representations of query terms as input to the network (W2V-LSTM).

5.3 Experimental Result

We conduct experiments to evaluate the effectiveness of the proposed ISWE in capturing word-intent and the intent-sensitive user satisfaction models in differentiating between SAT and DSAT tasks. We also measure the impact of task-type on user satisfaction and the effectiveness of the proposed models in predicting satisfaction in tasks having different types. Specifically, we address the following research questions in this section:

(1) *How effective are the intent-sensitive word embeddings in estimating word similarity by capturing word-intent compared to the Skip-gram word2vec model?*

(2) *How effective is the intent-sensitive user satisfaction model compared to the user satisfaction prediction baselines?*

(3) *How effective is the intent-sensitive user satisfaction model in different task types?*

5.3.1 Intent-Sensitive Word Embeddings in Word Similarity. The word representations learned by our proposed ISWE are expected to capture word intent. Table 2 shows three different examples of the nearest neighbors of words based on cosine similarity. As it is shown in Table 2, in contrast to the Skip-gram word2vec embeddings that is not capable of capturing word-intent by putting words like "yes" and "no" very close in the representation space, ISWE can effectively capture word-intent by scattering words having different intents. For example, Table 2 indicates that the top-5 similar words to word "yes" based on cosine similarity of word2vec Skip-gram word representations are "yeah", "not", "okay", "no" and "ok", which includes words with very different query-intents. On the other hand,

Table 3: Task satisfaction prediction result. * indicates statistical significant improvements based on Student's paired t-test and Wilcoxon signed-rank test ($\rho < 0.05$).

Classifier	P	R	Acc.	F1
QR	0.5117	0.5059	0.6248	0.5086
GM	0.6056	0.6068	0.6345	0.6062
W2V-LSTM	0.6513	0.6209	0.6910	0.6356
I2V-LSTM	0.6254	0.5528	0.6566	0.5862
ISWE-LSTM	**0.6891***	**0.6669***	**0.7174***	**0.6778***
Impr. over GM (%)	13.80%	9.90%	13.07%	11.81%
Impr. over W2V-LSTM (%)	5.80%	7.41%	3.82%	6.63%

Table 4: Satisfactory session distribution over different task type.

Task Type	Distribution	Sat Task Percentage
Single-Query	54.33%	60.05 %
Two-Turn	18.20%	63.00 %
Multi-Turn	27.47%	77.02 %

the top-5 similar words to word "yes" based on cosine similarity of ISWE word representations are "yeah", "yep", "sounds", "correct" and "ok", which is an example of how ISWE captures word-intent in the word representations.

Capturing word-intent in the word representations can be very beneficial for user satisfaction prediction, as understanding user query intent incorrectly can lead to a DSAT experience. For example IA might proactively ask a user "You have a reminder. Should I read it?". Then, if the user says "no" and the IA starts reading the reminder, the user would have a DSAT experience. However, if a user satisfaction prediction model cannot capture the user's intent, then it might classify the task to the SAT category as it considers words "yes" and "no" as very similar words.

ISWE performs better than the standard skip-gram word embeddings because the objective function of the ISWE does not just depend on words linear context in sentences. In fact, query-intent plays a major role in the objective function, which scatters words having different intents. In the next part, we discuss effectiveness of our proposed intent-sensitive user satisfaction model compared to the baselines.

5.3.2 Impact of Intent Sensitive Word Embeddings on User Satisfaction. We now answer our research question: *How effective is the intent-sensitive user satisfaction model compared to the user satisfaction prediction baselines?*

Table 3 shows the user satisfaction prediction results of the proposed intent-sensitive user satisfaction prediction models compared to the baselines. In these experiments, the classification threshold is 0.5. The experiment is based on a 5-fold cross validation, where three folds were used for training, one for validation, and one for testing. We repeat the process for all the five folds and report the average of the evaluation metrics. According to our experimental results, the query reformulation baseline performs poorly in the prediction of user satisfaction. One possible explanation for this is the dialogue nature of queries, where users might refine their queries to give more details about their information needs. It also has a huge bias toward predicting the SAT class, which could be the second possible explanation of its poor performance. On the other hand, the generative model performs much better than the query reformulation baseline, which provides a good baseline for our proposed model evaluation.

As it is shown in Table 3, the intent-sensitive user satisfaction based on ISWE query representations leads to a significant improvement over all the baselines in terms of all the defined evaluation metrics. Specifically, it improves task satisfaction prediction over the generative model from 0.6062 to 0.6778 in terms of $F1$ metric. The proposed model also has a statistically significant improvement over the task prediction based on the Skip-gram query representation, in which the $F1$ metric is improved from 0.6356 to 0.6778.

One possible explanation of the improvements achieved by the ISWE-LSTM over the I2V-LSTM is that in contrast to the I2V-LSTM model getting sequence of queries as an input, the ISWE-LSTM gets sequence of intent-sensitive query terms as an input. Therefore, effectiveness of the ISWE-LSTM is less affected by single-query tasks. Furthermore, ISWE-LSTM has improvements over other baselines that also use query terms as input, because the ISWE model takes advantage of the query intents for satisfaction prediction.

Our experimental results shows that the user satisfaction prediction based on Intent2Vec query-intent representations (I2V-LSTM) does not lead to an improvement over the generative model in terms of the F1-measure. However, as it is shown in Table 4, 54.33% of tasks in our crowdsourced dataset contain a single query, which is not ideal for training the user satisfaction prediction based on a sequence of intents. We suspect that the I2V-LSTM would do better for long tasks having multiple queries compared to single query tasks. We investigate this in the next experiment.

5.3.3 Impact of Task Type on User Satisfaction. In our final experiment, we answer our research question: *How effective is the intent-sensitive user satisfaction model in different task types?*

The dialogue nature of user queries in IAs leads to tasks, in which users might issue multiple queries to accomplish a single information need. However, although users are capable of having long conversation with the IA, the majority of tasks are single query tasks in smart speakers [18]. In Table 4, we categorized tasks into three different types: 1) Single-Query tasks are tasks having a single query and a single system response, 2) Two-Turn tasks are those tasks where the user issues a second query after receiving a system response to their first query, and 3) Multi-turn tasks are tasks having more than two user queries (i.e., multiple turns).

According to Table 4, users seem more satisfied in tasks having multiple queries. One possible explanation of such a behavior is that the IA retains the context of the dialogue, and shorter sessions are more probable to be sessions that IA was not able to retain the dialogue context and consequently the user abandoned the task.

As the user satisfaction level varies based on the defined task-types, and user behavior might be different in each of them, we investigate user satisfaction prediction for each task-type separately. Figure 5 details our proposed intent-sensitive user satisfaction prediction models effectiveness compared to the baselines in different task types. As expected, the query reformulation approach does not work well for the single query tasks as it assign SAT labels to all tasks. The query reformulation does not work well in two-turn and multi-turn tasks either, which shows that query reformulation is a poor approach for measuring user satisfaction in IAs on smart speakers. The generative model performs better for two-turn

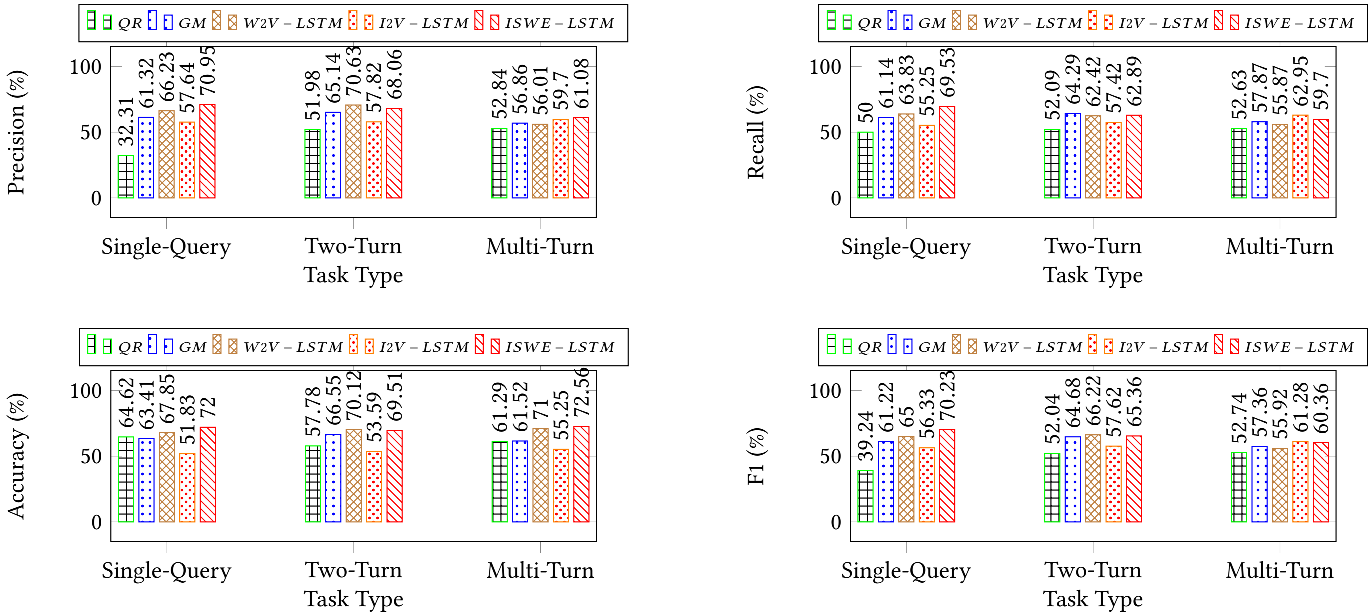


Figure 5: Effectiveness of intent-sensitive user satisfaction models compared to the baselines in different task types.

tasks compared to multi-turn tasks. We observed similar results for W2V-LSTM model in terms of precision and F1.

Figure 5 shows that our proposed intent-sensitive models perform better than all the baselines in single-query and multi-turn task-types in terms of all common metrics; however, the effectiveness of the generative model, W2V-LSTM and ISWE-LSTM in user satisfaction prediction in two-turn sessions are similar. In fact, the generative model and W2V-LSTM perform slightly better than ISWE-LSTM. The generative model performs better than W2V-LSTM and ISWE-LSTM in terms of recall of two-turn tasks, and W2V-LSTM is better than generative model and ISWE-LSTM in terms of precision. However, in terms of F1 as a more fair metric to evaluate classification, the genarative model, W2V-LSTM and ISWE-LSTM perform very similar.

Furthermore, as it is shown in Figure 5, the I2V-LSTM model improves as task length increases. As expected, the I2V-LSTM improves the task satisfaction prediction based on F1 score from 56.33 in single-query tasks to 61.28 in multi-turn tasks. Specifically, in single-query tasks, the I2V-LSTM performs poorly because of using a single query-intent as input to the sequence classification. In two-turn tasks, I2V-LSTM performance is improved as it uses a sequence of 2 intent representations in contrast to a single intent representation in single-query tasks. Using longer sequence of intents leads to a better understanding and prediction of user satisfaction in I2V-LSTM model, which makes the I2V-LSTM model the best performing user satisfaction model for multi-turn tasks in terms of F1.

To summarize, in this section, we have presented experimental results and shown effectiveness of ISWE compared to the original Skip-gram word2vec in capturing word-intents and learning query representation for user satisfaction prediction. Then, we discussed satisfaction in different task-types. In contrast to two-turn tasks, in which the generative model, W2V-LSTM and ISWE-LSTM models perform similarly, our experimental results indicates that the I2V-LSTM intent-sensitive user satisfaction prediction model is the best performing system in terms of F1, which is our main evaluation metric. Moreover, due to using intent-sensitive word-level representations of queries, ISWE-LSTM intent-sensitive user satisfaction prediction model is the best approach for single-query tasks.

6 CONCLUSION

In this paper, we investigated the user satisfaction prediction problem in intelligent assistants (IAs) on smart speakers, in which the only means of user interactions with the IA is a sequence of user queries. Our main research question was: *How to evaluate user satisfaction in Intelligent Assistants based on user queries?* To do this, we proposed a new implicit signal from user queries to predict user satisfaction, which is query intent. Using the query intent, we proposed an intent-sensitive user satisfaction prediction model. Learning effective query representations as input of the user satisfaction prediction model is one of the main contributions of this paper. To train intent-sensitive query representations, we proposed two intent-sensitive models. We first proposed an intent-sensitive word embeddings (ISWE) learning model, which is a modification of the popular word2vec Skip-gram model. According to our experimental results, the ISWE are very effective in capturing query term intents compared to the original Skip-gram model. Second, we proposed an unsupervised Intent2Vec Skip-gram model to capture linear context of query intents in user sessions.

According to our experiments, incorporating ISWE as the input to a user satisfaction prediction model based on a sequence of query terms leads to a statistically significant improvement over all the defined baselines. Furthermore, we further evaluated the effectiveness of the proposed intent-sensitive user satisfaction prediction models in different task types and showed that the proposed intent-sensitive user satisfaction model based on ISWE performs better than the baselines in single-query and multi-turn task-types, yet performs similar to baselines in two-turn task-type in terms of the common classification metrics. One possible explanation for this could be the nature of queries used in the training phase of ISWEs. The training set for learning the ISWEs was query and intent pairs without considering the whole task context. Therefore, the learned ISWEs are optimized for a single-query level context, and consequently the user satisfaction prediction model based on ISWE performs better in single-query task type. Furthermore, our experimental results indicate that compared to the two-turn task type, our proposed user satisfaction prediction based on ISWE performs better than baselines in multi-turn task type as it gets more inputs about the task-context by getting more queries as input. In addition, the experimental results indicate that the number of queries in a task (task length) has a positive impact on the user satisfaction prediction based on a sequence of Intent2Vec query intent representations, which leads to the best performing system in terms of F1 in multi-turn task-type. We have not studied more advance neural network architectures, as it is not the focus of this paper and we leave this to future work. Furthermore, other query features could be added to the intent-sensitive user satisfaction prediction model with the aim of improving the satisfaction measurement results. We also leave this for future work.

REFERENCES

[1] Mikhail Ageev, Qi Guo, Dmitry Lagun, and Eugene Agichtein. 2011. Find It if You Can: A Game for Modeling Different Types of Web Search Success Using Interaction Data. In *SIGIR*. 345–354.

[2] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Cro. 2017. Learning a hierarchical embedding model for personalized product search. In *SIGIR*. 645–654.

[3] Mohannad ALMasri, Catherine Berrut, and Jean-Pierre Chevallet. 2016. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *ECIR*. Springer, 709–715.

[4] Eytan Bakshy and Dean Eckles. 2013. Uncertainty in Online Experiments with Dependent Data: An Evaluation of Bootstrap Methods. In *KDD*. 1303–1311.

[5] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-parameter Optimization. *J. Mach. Learn. Res.* 13, 1 (2012), 281–305.

[6] F.A. Brown, M.G. Lawrence, and V.O.B. Morrison. 2017. Conversational virtual healthcare assistant. https://www.google.com/patents/US9536049 US Patent 9,536,049.

[7] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *SIGIR*. 65–74.

[8] Alex Deng, Tianxi Li, and Yu Guo. 2014. Statistical Inference in Two-stage Online Controlled Experiments with Treatment Selection and Validation. In *WWW*. 609–618.

[9] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *ACL* (2016).

[10] Alexey Drutsa, Gleb Gusev, and Pavel Serdyukov. 2015. Future User Engagement Prediction and Its Application to Improve the Sensitivity of Online Experiments. In *WWW*. 256–266.

[11] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Learning Topic-Sensitive Word Representations. *ACL* (2017).

[12] Henry A. Feild, James Allan, and Rosie Jones. 2010. Predicting Searcher Frustration. In *SIGIR*. 34–41.

[13] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating Implicit Measures to Improve Web Search. *ACM Trans. Inf. Syst.* 23, 2 (2005), 147–168.

[14] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. 2015. Word Embedding Based Generalized Language Model for Information Retrieval. In *SIGIR*. 795–798.

[15] Alex Graves et al. 2012. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer.

[16] Seyyed Hadi Hashemi, Charles L. A. Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M. Voorhees. 2016. Overview of the TREC 2016 Contextual Suggestion Track. In *TREC*.

[17] Seyyed Hadi Hashemi and Jaap Kamps. 2017. Where To Go Next?: Exploiting Behavioral User Models in Smart Environments. In *UMAP*. ACM, 50–58.

[18] Seyyed Hadi Hashemi, Kyle Williams, Ahmed El Kholy, Imed Zitouni, and Paul Crook. 2018. Impact of Domain and User's Learning Phase on Task and Session Identification in Smart Speaker Intelligent Assistants. In *CIKM*.

[19] Ahmed Hassan. 2012. A Semi-supervised Approach to Modeling Web Search Satisfaction. In *SIGIR*. 275–284.

[20] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. 2010. Beyond DCG: User Behavior As a Predictor of a Successful Search. In *WSDM*. 221–230.

[21] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. 2013. Beyond clicks: query reformulation as a predictor of search satisfaction. In *CIKM*. 2019–2028.

[22] Ahmed Hassan and Ryen W. White. 2013. Personalized Models of Search Satisfaction. In *CIKM*. 2009–2018.

[23] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580 (2012).

[24] Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. 2015. Automatic Online Evaluation of Intelligent Assistants. In *WWW*. 506–516.

[25] Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W. White. 2015. Understanding and Predicting Graded Search Satisfaction. In *WSDM*. 57–66.

[26] Maryam Kamvar, Melanie Kellar, Rajan Patel, and Ya Xu. 2009. Computers and Iphones and Mobile Phones, Oh My!: A Logs-based Comparison of Search Users on Different Devices. In *WWW*. 801–810.

[27] Diane Kelly. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval* 3, 1-2 (2009), 1–224.

[28] Madian Khabsa, Aidan Crook, Ahmed Hassan Awadallah, Imed Zitouni, Tasos Anastasakos, and Kyle Williams. 2016. Learning to Account for Good Abandonment in Search Success Metrics. In *CIKM*. 1893–1896.

[29] Youngho Kim, Ahmed Hassan, Ryen W. White, and Yi-Min Wang. 2013. Playing by the Rules: Mining Query Associations to Predict Search Performance. In *WSDM*. 133–142.

[30] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. 2014. Comparing Client and Server Dwell Time Estimates for Click-level Satisfaction Prediction. In *SIGIR*. 895–898.

[31] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. 2014. Modeling Dwell Time to Predict Click-level Satisfaction. In *WSDM*. 193–202.

[32] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[33] Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C Crook, Imed Zitouni, and Tasos Anastasakos. 2016. Predicting user satisfaction with intelligent assistants. In *SIGIR*. ACM, 45–54.

[34] Julia Kiseleva, Kyle Williams, Jiepu Jiang, Ahmed Hassan Awadallah, Aidan C Crook, Imed Zitouni, and Tasos Anastasakos. 2016. Understanding user satisfaction with intelligent assistants. In *CHIIR*. ACM, 121–130.

[35] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. 2014. Towards Better Measurement of Attention and Satisfaction in Mobile Search. In *SIGIR*. 113–122.

[36] Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings.. In *ACL*. 302–308.

[37] Rishabh Mehrotra and Emine Yilmaz. 2017. Task Embeddings: Learning Query Embeddings Using Task Context. In *CIKM (CIKM '17)*. 2199–2202.

[38] Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan Awadallah, Ahmed El Kholy, and Madian Khabsa. 2017. User Interaction Sequences for Search Satisfaction Prediction. In *SIGIR*. ACM, 165–174.

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

[40] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*. 2786–2792.

[41] Liqiang Niu, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. 2015. Topic2Vec: learning distributed representations of topics. In *IALP*. IEEE, 193–196.

[42] Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *ACL* (2014).

[43] Filip Radlinski, Martin Szummer, and Nick Craswell. 2010. Inferring Query Intent from Reformulations and Clicks. In *WWW*. 1171–1172.

[44] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Hamed Zamani. 2017. Word Embedding Causes Topic Shifting; Exploit Global Context!. In *SIGIR*. 1105–1108.

[45] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[46] Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. 2010. Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. In *KDD*. 17–26.

[47] Jaime Teevan, Amy Karlson, Shahriyar Amini, A. J. Bernheim Brush, and John Krumm. 2011. Understanding the Importance of Location, Time, and People in Mobile Local Search Behavior. In *MobileHCI*. 77–80.

[48] Kyle Williams, Julia Kiseleva, Aidan C Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. 2016. Detecting good abandonment in mobile search. In *WWW*. 495–505.

[49] Kyle Williams, Julia Kiseleva, Aidan C Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. 2016. Is This Your Final Answer?: Evaluating the Effect of Answers on Good Abandonment in Mobile Search. In *SIGIR*. ACM, 889–892.

[50] Kyle Williams and Imed Zitouni. 2017. Does That Mean You're Happy? RNN-based Modeling of User Interaction Sequences to Detect Good Abandonment. In *CIKM*. ACM.

[51] Hamed Zamani and W. Bruce Croft. 2016. Embedding-based Query Language Models. In *ICTIR*. 147–156.

[52] Hamed Zamani and W. Bruce Croft. 2016. Estimating Embedding Vectors for Queries. In *ICTIR*. 123–132.

[53] Hamed Zamani and W. Bruce Croft. 2017. Relevance-based Word Embedding. In *SIGIR*. 505–514.

[54] Guoqing Zheng and Jamie Callan. 2015. Learning to Reweight Terms with Distributed Representations. In *SIGIR*. 575–584.

[55] Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering.. In *ACL (1)*. 250–259.