Predicting User Satisfaction in Spoken Dialog System Evaluation With Collaborative Filtering

Zhaojun Yang, Gina-Anne Levow, and Helen Meng, *Senior Member, IEEE*

***Abstract*—We propose a collaborative filtering (CF) model to predict user satisfaction in SDS evaluation. Inspired by the use of CF in recommendation systems, where a user's preference for a new item is assume to resemble that for similar items rated previously, we adapt the idea to predict user evaluations of unrated dialogs based on the ratings received by similar dialogs. Ratings of dialogs are gathered by crowdsourcing through Amazon Mechanical Turk. A reference baseline is provided by a linear regression model (LRM) based on the PARADISE framework. We present two versions of the CF model. First, the item-based collaborative filtering model (ICFM) clusters rated dialogs and builds an LRM for each cluster. The rating of an unseen dialog is predicted by the LRM of its most similar cluster. Second, the extended ICFM (EICFM) separates dialog features into user-related and system-related groups, to build LRMs for these separately. Experimental results on dialogs from the Let's Go! system show both ICFM and EICFM can significantly improve the proportion of variability explained by the LRM. We also demonstrate the generalizability of the CF model to a new dialog corpus from the systems in the Spoken Dialog Challenge (SDC) 2010.**

***Index Terms*—Spoken dialog system evaluation, collaborative filtering, crowdsourcing user satisfaction.**

I. Introduction

A SPOKEN DIALOG SYSTEM (SDS) is a computer system which supports human-computer conversations in a restricted domain. The interaction between the human and computer is composed of many spoken dialog turns. A typical turn between a system and user is shown in Fig. 1: the first step for the system is to recognize the user's speech using automatic speech recognition, followed by the language understanding component which aims to interpret the user's intended semantics and actions. The dialog model maintains the history of the dialog, decides which action is appropriate based on language understanding and the discourse context, and controls the dialog flow. After the dialog model issues a proper action as the response, the natural language generator is responsible for translating the representation of the response semantics into text, which is then passed to the Text-to-Speech (TTS) synthesizer to generate audio output.

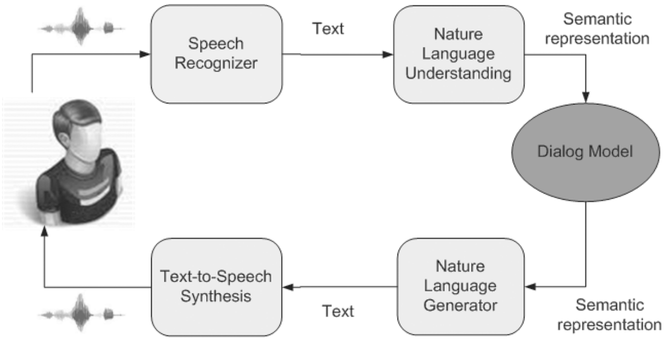

Fig. 1. A typical spoken dialog turn between the system and the user.

Advances in speech and language technologies have made SDSs an important research area and have brought about systems in a wide variety of application domains, such as bus schedule inquiries [1], flight information [2], stock market information delivery [3], tourist guides [4] and student tutoring [5]. As SDSs are becoming increasingly pervasive, their ultimate goal is to satisfy the users' needs with good performance yielding a good user experience. This calls for a sound strategy to evaluate, compare, and predict the performance of SDSs, which remains an open research problem. Generally, SDS evaluation can be categorized into *component-based* and *holistic* perspectives.

The component-based perspective covers the performance of individual components such as the correctness of speech recognition, the ability to understand natural language, the appropriateness of response generation, as well as the naturalness of the synthetic speech in conveying the responses. A thorough evaluation of an SDS needs to consider all relevant evaluation metrics covering the functionalities for all the system components [6].

In contrast, holistic evaluation assesses not only individual components but also the integrated performance of an SDS. It involves the perceived level of system usability, system intelligence and abilities in error recovery by considering the system in its entirety [7]. Holistic evaluation also covers the wide variety of users' impressions (user judgments) relating to all dimensions of the quality of an SDS [8]. The ultimate objective of an SDS is to satisfy the demands of real users. Therefore, user satisfaction is considered the most important criterion for system evaluation [9].

Many evaluation methods have been developed in recent years. One popular method of measuring user judgments is

Manuscript received July 24, 2012; revised October 19, 2012; accepted November 12, 2012. Date of publication November 27, 2012; date of current version January 03, 2013. This work was supported in part by a grant from the HKSAR Government Research Grants Council (Project No. 415609). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pascal Poupart.

Z. Yang is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: zhaojuny@usc.edu).

G. Levow is with the Department of Linguistics, University of Washington, Seattle, WA 98195 USA (e-mail: levow@u.washington.edu).

H. Meng is with the Human–Computer Communications Laboratory, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: hmmeng@se.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2012.2229965

through questionnaires completed after users interact with an SDS. The questionnaire often involves a range of perceptions of the system such as task completion and user satisfaction. This traditional approach has some disadvantages. First, it is a costly and time-consuming process. Moreover, this approach is often limited to a small number of evaluators whose feedback may not be statistically representative of the larger user population that can access the SDS. Furthermore, when a system has already been deployed, real users are often unwilling to patiently complete an evaluation survey.

Another popular evaluation method, the PARADISE framework, has been proposed for automatic inference of overall user satisfaction of unrated dialogs [10]. It assumes that the overall performance (user satisfaction) of an SDS can be described in terms of a linear regression model of a set of dialog metrics [11]. The trained model can explicitly identify which factors contribute most to user satisfaction. Its predictive power is measured in terms of the proportion of variability explained by the regression, denoted by R-squared (R^2). Low R^2 may be caused by the lack of inter-rater agreement on user satisfaction ratings [12], or the linear model may not sufficiently capture the relations between user satisfaction and dialog features.

The primary emphasis of this paper is on the development of an efficient and effective paradigm for SDS evaluation. Recently, crowdsourcing technologies have been widely applied to collect, transcribe and annotate speech and language data [13]–[15]. Crowdsourcing refers to outsourcing a task to a crowd of people. Unlike the traditional method in which data is manually labeled by experts or trained people, tasks can be completed with crowdsourcing in a cost-effective, efficient and flexible manner. Novotney *et al.* collected high quality transcriptions of conversational speech with only one thirtieth the cost of professional transcription [15]. Snow *et al.* conducted varieties of NLP tasks in the crowd [14]. We believe that user judgments for SDS evaluation can also be collected by using crowdsourcing instead of user experiments. Our previous work [16] has developed a crowdsourcing methodology for this purpose through Amazon Mechanical Turk (MTurk) (http://www.mturk.com). In addition, collaborative filtering (CF) has been successfully applied to the development of recommendation systems [17]. It assumes that the preference of a user for a new item may resemble that for the similar items rated previously by users. Collaborative filtering appoaches can be categorized as user-based [18], [19] and item-based [17], [20]. Given an unrated item of a target user, user-based collaborative filtering searches for the most similar users to the target user [19]. In contrast, item-based collaborative filtering searches for items most similar to the target one in a data set which has been rated. Prediction of the target item is then computed based on other similar items [21]. Item-based CF is computationally efficient and can guarantee recommendation quality [22]. We adapt this approach to predict user evaluations of unrated dialogs assuming that the rating of a (previously unrated) target dialog should be similar to the ratings received by similar dialogs. Therefore, this work aims to extend the PARADISE framework by incorporating CF to improve its prediction performance, based on user judgments collected through MTurk. We believe that the information from the most similar neighbors of an unrated dialog can better predict its performance than the information from the whole corpus. In addition to the work in [23] showing that the CF model can significantly improve the performance of predicting user satisfaction, this paper will also demonstrate the generalization ability of the CF model across multiple systems.

The rest of this paper is organized as follows: in Section 2, we present a review of previous work in SDS evaluation. Section 3 describes our experimental corpus and user annotations obtained through crowdsourcing. Section 4 details our collaborative filtering (CF) model in prediction of user satisfaction. Section 5 gives the experimental results and analysis and shows that the CF approach can distinctly improve prediction accuracy over a baseline obtained from the PARADISE framework. Finally, Section 6 concludes the paper and points out some directions for future research.

II. Prior Work on SDS Evaluation

The performance of an SDS can be measured with many types of metrics, such as task success, number of utterances in the dialog, speech recognition accuracy, system response delay, naturalness of output speech, users' expectations and cooperativeness of the system [6]. These metrics are used for both component-based and holistic evaluation and are usually categorized into *subjective* and *objective* metrics. Subjective metrics, which reflect users' perceptions of the quality of an SDS, are often obtained from real or test users. Objective metrics, which quantify the system's behavior during interactions and the performance of various components of an SDS, can be extracted automatically or labeled manually by experts based on user-system interactions. Objective metrics are also called interaction metrics in [6].

A. Subjective User Judgments

Since subjective metrics mostly rely on user judgments of system quality, distributing questionnaires to users before or after interaction with an SDS is an effective way to collect quantifiable user judgments. Developing a reliable and valid questionnaire for subjective judgment collection has attracted much attention in the research community. The SASSI questionnaire (Subjective Assessment of Speech System Interfaces) is designed for subjective assessment of speech-based systems [24]. SASSI consists of 50 items (statements), and each item is rated by users on a 7-point scale of agreement: strongly agree, agree, slightly agree, neutral, slightly disagree, disagree and strongly disagree. A factor analysis of the collected data from 226 completed questionnaires suggests that there are six main factors that contribute to a user's subjective perception of speech-based systems, i.e., perceived system response accuracy, likeability, cognitive demand, annoyance, habitability and speed.

The ITU recommendation proposed another list of questions for the evaluation of SDSs in telephone services [25]. Three types of questionnaires are distinguished in the recommendation. Type 1 questionnaires are intended to collect the user's background information and are distributed at the beginning of an evaluation experiment. Type 2 questionnaires are related to user-system interactions. Type 3 questionnaires are about the users' overall impressions of the system quality. A list of topics are proposed for each type of questionnaire and exemplar statements are rated on a 5-point scale.

B. Interaction Metrics

In contrast to subjective judgments of system performance, interaction metrics can easily quantify the ability of the system or its components to perform the designed functions. Such information is obtained from the log files which record the interactions between the system and users. Surface metrics that are based on utterances from the user or the system, such as dialog duration or recognition confidence, can often be automatically extracted from the log files. Other metrics that are related to the content of the interactions, such as language understanding accuracies or task success rates, are usually manually labeled by experts or trained annotators.

In recent decades, many metrics have been identified for measuring the functions of the system and its components. Early metrics were for individual components, such as the speech recognizer and language understanding components. Commonly used metrics are *Word Accuracy (WA), Sentence Accuracy (SA), Concept Accuracy (CA), Query Density (QD), Concept Efficiency (CE)* [26], *etc*. Later, metrics for whole systems were developed, including *Task Success (TS)* to measure the extent to which the system achieves the task, *number of dialog turns* for measuring the dialog cost, or *Contextual Appropriateness* for measuring the degree to which the system provides an appropriate response [27].

Based on the literature about interaction metrics, Möller *et al.* summarized a set of metrics for SDSs evaluation and classified them into five categories [6]:

- The dialog- and communication-related category: Metrics about the overall dialog, such as the overall dialog duration, number of dialog turns, average number of words per system turn, *etc.*
- The meta-communication-related category: Metrics describing speech recognition and language understanding capabilities, such as the number of help requests, number of barge-in attempts from the user, *etc.*
- The cooperativity-related category: Metrics about the cooperativity of system actions (responses). The contextual appropriateness of system responses directly measures cooperativity, which is often evaluated by several human experts based on Grice's maxims.
- The task-related category: Task success is a key aspect of successful task-oriented systems. Möller defined task success in seven aspects, i.e., success in providing a completely right answer; success with relaxation of constraints from the user, the system, or from both the user and the system; success in spotting that no solution exists; failure that results from the user's non-cooperative behavior or the system's inappropriate response.
- The speech-input-related category: Metrics about the capability of the system to recognize the input speech and to understand the meaning of the input. Commonly used metrics are *WA, SA, or CA* as introduced above.

This categorization and the metrics in each category have been incorporated in the ITU recommendation [28].

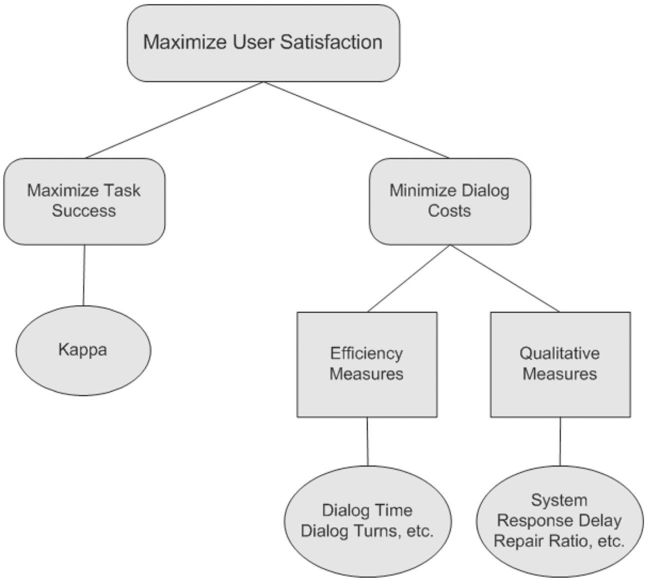

Fig. 2. The PARADISE structure of objectives for dialog performance [10].

C. The PARADISE Framework

PARADISE (PARAdigm for DIalogue System Evaluation) is a general framework for evaluating and comparing the performance of spoken dialog systems [10]. It identifies which system properties have a large impact on system usability and supports the development of predictive models of system performance.

PARADISE uses decision theoretic methods to relate a series of dialog metrics to the system's overall performance and determine the significant contributors. The PARADISE performance model is shown in Fig. 2. In this model, the overall performance is correlated with user satisfaction. Hence, the primary objective of a system is to maximize user satisfaction. This objective can be further decoupled into two sub-objectives: maximizing task success and minimizing dialog cost, assumed to be the two main contributors to user satisfaction. In the original PARADISE framework, task success is measured with the use of the Kappa coefficient derived from an attribute value matrix (AVM). Dialog costs can be categorized into two types: dialog efficiency and quality. Dialog efficiency is represented by the number of dialog turns or the dialog duration, while dialog quality is measured in terms of the appropriateness of system response, or system repair ratio, *etc.*

The PARADISE framework posits that the objective structure in Fig. 2 can be realized by building a performance model through multivariate linear regression with user satisfaction as the target and the dialog metrics of task success, dialog efficiency and quality as predictors. Building the performance model requires a dialog corpus to be collected through controlled user experiments during which users subjectively rate their satisfaction. Moreover, the predictors of the model, i.e., the dialog metrics, can either be automatically extracted from dialog log files or manually labeled by experts. Based on these illustrations, the performance model of an SDS is:

$$S_u = (\alpha * N(\kappa)) - \sum_{i=1}^{n} w_i * N(c_i), \quad (1)$$

where S_u is system performance correlated with user satisfaction here, κ is a measure for task success, c_i is a measure for dialog cost, n is the number of dialog measures, α is a weight on κ, w_i is a weight on c_i, and $N(\cdot)$ is a z-score normalization function [29]. Both κ and c_i can be represented as dialog measures m, so (1) can be simplified to be:

$$S_u = \sum_{i=1}^{n+1} w_i * N(m_i). \tag{2}$$

Since the dialog measures have been normalized into the same scale by $N(\cdot)$, the weight w_i reflects the relative contribution of the corresponding measure m_i to user satisfaction. We will use this linear regression model as our baseline model.

By applying the performance model, values of user satisfaction of SDSs are directly predicted from a suite of dialog metrics which are simply extracted from dialogs, without the need to conduct user experiments to assess user satisfaction. In addition, system developers can directly figure out which system components have greater impact on user satisfaction by inspecting the coefficients of dialog metrics in the performance model, so that they can focus on improving the performance of those "important" components.

PARADISE has been widely applied in evaluating many SDSs, such as the ITSPOKE tutoring system [30] and DARPA Communicator [31]. It has been applied to test dialog models with different initiative and information presentation policies in the two systems ELVIS and TOOT, respectively [11]. Kamm *et al.* discussed the generalization ability of PARADISE across three different systems [32]. Walker *et al.* studied its generalization across different user populations and found that the it did not generalize well from novice users to expert users [33]. Researchers have also extended PARADISE to evaluate multimodal systems [34], [35].

III. Experimental Corpus and Annotation

We first describe the dialog data that forms the basis of our experiments. We then present the crowdsourced annotation of spoken dialog system quality that forms the training data and ground truth for automatic prediction of user satisfaction.

A. Spoken Dialog Corpus

All dialog data employed in the current study are drawn from materials associated with the Spoken Dialog Challenge (SDC) 2010 [36] and the CMU Let's Go! system [1]. SDC 2010 was coordinated by Maxine Eskenazi and Alan Black from Carnegie Mellon University (CMU) and was designed to focus effort on tasks including spoken dialog systems (SDS) development, user simulation, and SDS evaluation. The application domain was information-seeking, specifically to access Pittsburgh bus information. Dialog examples from the four SDC systems are shown in Tables I–IV.

TABLE I
An Example Dialog From System 1 (the Let's Go! System)

S: *Welcome to the CMU Let's Go bus information system*

S: *What can I do for you?*

U: *61B*

S: *The 61B. Is this correct?*

U: *YES*

S: *Where do you wanna leave from?*

U: *BRADDOCK AND TALBOT*

S: *Leaving from BRADDOCK AND TALBOT?*

U: *YES*

S: *Where do you wanna go?*

U: *SWISSVALE*

S: *Going to SWISSVALE?*

U: *YES*

S: *When are you going to take that bus?*

U: *NOW*

S: *You want the next bus. Is this correct?*

U: *YES*

S: *Hold on. Let me check that for you.*

S: *The next 61B leaves TALBOT AVENUE AT FIFTH BRADDOCK HOSPITAL at 10:26 p.m. and arrives at WASHINGTON at 10:43 p.m.*

TABLE II
An Example Dialog From System 2

S: *Welcome to the Pittsburgh bus information line. Where are you going to take your bus from?*

U: *um oh great are there any market*

S: *did you mean ”fifth avenue at market”?*

U: *no*

S: *Okay*

S: *Where are you going to take your bus from?*

U: *null*

S: *I didn't quite catch that.*

S: *Where are you going to take your bus from?*

U: *null*

S: *I didn't quite catch that.*

S: *Where are you going to take your bus from?*

U: *null*

The SDC reference system implementation was drawn from the Let's Go! system, i.e., System 1 in Table I. Let's Go! has been deployed to regular users for several years to provide after-hours access to Pittsburgh bus information, when phone lines are not staffed. Data from over 50,000 dialogs is available. In addition to the training dialogs, we also utilize data collected from the test deployments of the systems developed for SDC 2010, from CMU and three other participating sites. In the final

TABLE III
AN EXAMPLE DIALOG FROM SYSTEM 3

S: *Welcome to the bus information system for Pittsburgh. Where would you like to leave from?*

U: *EAST OHIO STREET OPPOSITE CHESTNUT STREET*

S: *Where do you wanna go?*

U: *SOUTH SIDE EAST CARSON STREET*

S: *When do you want to travel?*

U: *TONIGHT AT NINE THIRTY*

S: *There is a 54C leaving from east ohio street at chestnut at 9:33 p.m. It arrives at east carson street at twentieth in the south side at 10:15p.m. You may now say. next bus, previous bus, restart, or goodbye.*

TABLE IV
AN EXAMPLE DIALOG FROM SYSTEM 4

S: *East Pittsburg Bus Schedules. Say a bus route, like X28, or say I'm not sure.*

U: *fifty four c*

S: *Okay, 54C. To change, say go back. Where are you leaving from?*

U: *bloomfield*

S: *Okay, bloomfield. Where are you going to?*

U: *corner of bausman and brownsville road*

S: *I heard corner of bausman and brownsville road, but I don't have any routes that run from bloomfield to corner of bausman and brownsville. Say the place you're going to, or you can just hang up.*

U: *mount oliver*

S: *Okay, mount oliver. Do you want times for the next few buses?*

U: *yeah*

S: *To get the next few buses, say YES. Otherwise, say NO.*

U: *yeah*

S: *At 10:20am today there is a 54 C from PENN AVENUE, arriving BROWNSVILLE ROAD and BAUSMAN at 11:00am. Say repeat, next, previous, or start over.*

stage of the shared task, these systems were made accessible to real users for a limited period of time. All dialog interactions, transcripts, data and features are drawn from automatically generated system log files. There are 310 SDC dialogs in total which will be used as testing data when verifying the generalization ability of our evaluation model.

B. Corpus Annotation for User Satisfaction

While the SDC dataset described above constitutes a large and important dataset, it lacks a key requirement for use in spoken dialog system evaluation: a measure of user satisfaction. This measure is crucial both for developing and assessing models to predict user satisfaction as a holistic measure of system quality. In laboratory settings, user satisfaction has typically been assessed through user questionnaires. However, in real-world deployment it is impossible to expect users to complete surveys, and the corpus is far too large to allow complete assessment by trained, and presumably expensive, experts. To overcome these problems, we developed a methodology to collect judgments of system quality through crowdsourcing.

Crowdsourcing refers to outsourcing a task to a large and typically non-expert group. In this work, we employ the Amazon Mechanical Turk (MTurk) crowdsourcing platform. We cast SDS evaluation as a "Human Intelligence Task" (HIT) presented by "Requesters" to be performed by online "Workers." Each HIT consisted of the automatic text transcript of one of the corpus dialogs, along with a questionnaire, motivated by prior work in SDS evaluation [24], [25]. The questions cover *user's confidence*, *perceived task completion*, *expected behavior*, *overall performance* and *categorization of task success*. The questionnaire, along with a brief description of the goal of each question, appears in Table V; the aims in Table V did not appear in the actual HIT, but are presented here for explication.

11,000 dialogs were uploaded to the MTurk platform and were rated by around 700 online workers in 45 days at a cost of a few hundred dollars. We designed tasks specifically to assess interrater agreement, with 16 Workers rating each dialog. Since the questions elicited responses that ranged in value from 1 to 5 (Q1–Q4) and from 1 to 7 (Q5), we employed Cohen's weighted Kappa [37]. We found a moderate level of agreement ($\kappa \approx 0.5$) for the more objective task success and task completion related measures captured by Q2 and Q5. This result provides validation for the crowdsourcing methodology and evidence of the reliability and utility of MTurk Workers. The levels of interrater agreement on the more subjective measures relating to user expectations and overall system quality (Q3, Q4) were lower ($\kappa < 0.3$), which is not surprising.

IV. COLLABORATIVE FILTERING MODEL

Collaborative filtering (CF) uses a database of users' preferences for items to predict the utility of a certain item for a particular user. Item-based techniques are one main class of CF implementations. These methods search for items most similar to the target one in a data set which has been rated by users. Suppose that the n most similar items to the target i are selected for the active user u, and their ratings by u are denoted as $\{r_{u,j}\}_{j=1}^{n}$. A typical way to predict the rating $P_{u,i}$ of the target item i for the user u is to compute the weighted sum of ratings on the n similar items,

$$P_{u,i} = \frac{\sum_{j\in\{n \text{ similar items}\}} s_{i,j} * r_{u,j}}{\sum_{j\in\{n \text{ similar items}\} s_{i,j}}}, \tag{3}$$

where the weights $\{s_{i,j}\}_{j=1}^{n}$ are similarities between i and the n items. For some more elaborate algorithms for item-based CF we refer readers to [17].

While our proposed algorithms are inspired by item-based CF, we want to highlight some differences between the SDS evaluation problem and CF. First, items in our problem are more consistent than those in recommendation systems—they are all dialogs. This unique characteristic allows us to represent the

TABLE V
QUESTIONS CONSTITUTING THE TASK ON DIALOG EVALUATION (Q: QUESTION, OPT: OPTIONS). THE QUESTIONNAIRE COVERS *THE USER'S CONFIDENCE*, *THE PERCEIVED TASK COMPLETION*, *THE EXPECTED BEHAVIOR*, *THE OVERALL PERFORMANCE* AND *THE CATEGORIZATION OF TASK SUCCESS*

Q1	**Do you think you understand from the dialog what the user wanted?**
Opt	1) No clue 2) A little bit 3) Somewhat 4) Mostly 5) Entirely
Aim	*elicit the worker's confidence in his/her ratings.*
Q2	**Do you think the system is successful in providing the information that the user wanted?**
Opt	1) Entirely unsuccessful 2) Mostly unsuccessful 3) Half successful/unsuccessful 4) Mostly successful 5) Entirely successful
Aim	*elicit the worker's perception of whether the dialog has fulfilled the informational goal of the user.*
Q3	**Does the system work the way you expect it?**
Opt	1) Not at all 2) Barely 3) Somewhat 4) Almost 5) Completely
Aim	*elicit the worker's impression of whether the dialog flow suits general expectations.*
Q4	**Overall, do you think that this is a good system?**
Opt	1) Very poor 2) Poor 3) Fair 4) Good 5) Very good
Aim	*elicit the worker's overall impression of the SDS.*
Q5	**What category do you think the dialog belongs to?**
Opt	1) TS:Fu – Failed because of the user behavior, due to non-cooperative user behavior 2) TS:Fs – Failed because of the system behavior, due to system inadequacies 3) TS:SN – Succeeded in spotting that no solution exists 4) TS:CsCu – Succeeded with constraint relaxation both from the system and from the user 5) TS:Cu – Succeeded with constraint relaxation by the user 6) TS:Cs – Succeeded with constraint relaxation by system 7) TS:S – Succeeded
Aim	*elicit the worker's impression of whether the dialog reflects task completion.*

items by common features (see Section IV.C), such as the total number of system turns or average recognition score, and the similarity between two dialogs is hence computed from their feature vectors. Secondly, the dialogs similar to the target may be rated by different users, so we do not intend to predict the rating of the target for a particular user u, but rather for a general population of users.

A. ICFM for User Satisfaction Prediction

We detail our item-based CF model (ICFM) for user satisfaction prediction in the following. Let $D = \{(d_i, r_i)\}_{i=1}^{N}$ be a large dialog corpus where each dialog d_i is rated as r_i. As pointed out in the previous section, we represent each dialog d_i with a feature vector $\mathbf{f}_i$, e.g., the total number of user help requests and average recognition score (see Section IV.C). The similarity between two dialogs d_i and d_j is measured as the cosine similarity of their feature vectors,

$$s_{i,j} \doteq s(d_i, d_j) = \frac{\mathbf{f}_i^T \mathbf{f}_j}{|\mathbf{f}_i| * |\mathbf{f}_j|}. \tag{4}$$

We could also choose other similarity measures, such as Euclidean distance, which we have tried and obtained very similar results.

To save computation time, we cluster the dialog corpus in advance. Any efficient clustering algorithm is acceptable and the basic k-means clustering is used in our work. Let $C = \{C_i\}_{i=1}^{M}$ be the set of clusters created from D such that $\cap_i C_i = \phi \& \cup_i C_i = D$. Therefore, the process of retrieving similar dialogs for the target dialog d is reduced to its assignment to a cluster C^*,

$$C^* = \arg\max_{C_i} s(d, c_i), \tag{5}$$

where c_i is the centroid of C_i. All the dialogs in cluster C^* are similar dialogs to the target dialog d.

Sarwar *et al.* pointed out that two items with high similarity may be distant in Euclidean distance [20], where the weighted combination of the raw ratings of such similar items may lead to poor prediction. Therefore, in our problem, instead of using the weighted sum in (3), we use linear regression built on dialogs in the selected cluster C^* to predict the rating for the target dialog d. Note that since we have partitioned the dialog corpus into M clusters, the linear regression can be trained on each cluster beforehand. With such modifications, ICFM is formulated as follows:

1) Extract feature vector $\mathbf{f}_i$ for each dialog $d_i \in D$.
2) Use k-means to create dialog clusters $C = \{C_i\}_{i=1}^{M}$ for the dialog corpus D based on the feature representations $\mathbf{f}$ and the similarity measure s in (4).
3) Build the linear regression model $r = L_i(\mathbf{f}) = \alpha_i + A_i\mathbf{f}$ for each cluster C_i, where A_i comprises the regression coefficients and α is the constant term.
4) Given an unseen dialog d (unevaluated dialog here), we first extract a feature vector $\mathbf{f}_d$ and then assign d into its nearest cluster C^* with (5).
5) Use L^* which is trained on C^* to predict user satisfaction for d, i.e., $r_d = L^*(\mathbf{f_d}) = \alpha^* + A^*\mathbf{f_d}$.

B. Extended ICFM for User Satisfaction Prediction

The features used to represent dialogs (see Section IV.C) can be separated into *user-related* and *system-related* types. For example, #BargeIn (i.e., the total number of the user's barge-in attempts) reflects the characteristics of user behavior and can be classified as a user-related feature, while #SystemQuestion (i.e., the total number of the system's questions in the dialog) is a system-related feature. The intuition for this separation is that the judgement rating for a dialog can be influenced by two types of features, i.e., user style and system quality. On one hand, users with different user styles may have different preferences for the dialog, which can result in different evaluations for the same dialog. On the other hand, a high-quality dialog coming from the system is more likely to get a high rating statistically. Ratings determined by user style can be obtained from user-related features and those due to system quality can be drawn from system-related ones. Hence, we can predict judgment ratings based on the two types of features *separately*, rather than on the basis of the entire feature set. Thus, we extend ICFM to EICFM, as follows:

1) Create system-related clusters $C^s = \{C^s_i\}_{i=1}^{M^s}$ for dialog corpus D based on system-related features $\mathbf{f}^s$, e.g., average recognition score.
2) Create user-related clusters $C^u = \{C^u_j\}_{j=1}^{M^u}$ for D based on user-related features $\mathbf{f}^u$, e.g., the total number of user help requests.
3) Build linear regression models $r^s = L^s_i(\mathbf{f}^s) = \alpha^s_i + A^s_i\mathbf{f}^s$ and $r^u = L^u_j(\mathbf{f}^u) = \alpha^u_j + A^u_j\mathbf{f}^u$ for C^s_i and C^u_j respectively.
4) Given an unseen dialog d, choose C^{s*} and C^{u*} which are the most similar to d with respect to $\mathbf{f}^s$ and $\mathbf{f}^u$, respectively.
5) Use the regression model L^{s*} of C^{s*} to predict system-related judgement r^s for d, i.e., $r^s = L^{s*}(\mathbf{f}^{\mathbf{s}}_{\mathbf{d}}) = \alpha^{s*} + A^{s*}\mathbf{f}^{\mathbf{s}}_{\mathbf{d}}$, and use $r^u = L^{u*}$ of C^{u*} to predict user-related judgement r^u, i.e., $r^u = L^{u*}(\mathbf{f}^{\mathbf{u}}_{\mathbf{d}}) = \alpha^{u*} + A^{u*}\mathbf{f}^{\mathbf{u}}_{\mathbf{d}}$.
6) The final rating r is obtained by linearly combining the two kinds of ratings, $r = r^u * w + r^s * (1 - w)$, where w is a weight varying from 0 to 1.

Compared with ICFM, EICFM can have a better balance between user judgments based on user style and system quality. As will be seen, experiments demonstrate that this extension distinctly improves evaluation performance.

C. *Extraction of Interaction Features*

This subsection describes how we extract the feature vector. The corpus consists of 5000 Let's Go! dialogs for which the user judgments have been collected with crowdsourcing through MTurk (see Section III.B).

Based on the ITU Recommendation [28], we extract 10 interaction features, defined in Table VI, from the log files for each dialog. These features are chosen since they can be automatically extracted from the log files. The features **#Help Requests** and **#User Questions** are obtained based on cue phrases such as "help",“ what", "where", *etc*. The features of **#System Turns**, **#User Turns**, **AveRecogScore**, **#Barge In** and **#Help Requests** were used in the original version of the PARADISE model [10]. The feature **#DTMF** is specific to the Let's Go! system since it provides touch tone functionality to users. Therefore, each dialog is represented by a vector $\mathbf{f}_i$ concatenating all ten features.

Among these features, **#System Turns**, **AveRecogScore** and **#System Questions** are considered *system-related* while the others are *user-related*. All features use z-norm scores in the following experiments, i.e., $z(f) = (f - \mu)/(\sigma)$, where μ and σ are the mean and standard deviation of feature f.

TABLE VI
FEATURES AUTOMATICALLY EXTRACTED FROM LOG FILES

Feature	**Definition**
#System Turns	Overall number of system turns
#User Turns	Overall number of user turns
WPUT	Average number of words per user turn
AveUserSpeakRate	Average speaking rate of the user
AveRecogScore	Average recognition score
#Barge In	Overall number of user barge in attempts
#Help Requests	Overall number of user help requests
#User Questions	Overall number of user questions
#System Questions	Overall number of system questions
#DTMF	Overall number of touch tone uses

V. EXPERIMENTAL RESULTS AND ANALYSIS

We conduct two experiments to investigate how ICFM and EICFM can improve user satisfaction prediction. The linear regression model (LRM) in PARADISE is used as the baseline model (see Section II.C). **Experiment I** compares the accuracy in predicting user satisfaction for ICFM, EICFM and LRM. **Experiment II** compares the mean values of true ratings and predictions of the test data over the number of system turns (**#System Turns**), because the LRM results show that it takes on the largest weight.

For convenience, we set the number of user-related clusters C^u to be equal to that of system-related clusters C^s in EICFM in all the experiments. As in [11], [38], we also use R^2 to evaluate the prediction accuracy of our models. R^2 is a popular measure of the predictive power of an evaluation model, which measures the proportion of variability explained by the regression model. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{\sum_{i=1}^{n}(r_i - \bar{r})^2}, \tag{6}$$

where n is the total number of ratings, r_i is the ground truth rating, $\hat{r}_i$ is the predicted rating from a prediction model, and $\bar{r}$ is the mean of $\{r_i\}_{i=1}^{n}$. R^2 varies from 0 to 1, and a higher value indicates a higher prediction accuracy.

A. *Prediction of User Satisfaction*

In **Experiment I**, we use 10-fold cross validation on the data corpus (5,000 rated Let's Go! dialogs, see Section III.B) to measure R^2 in predicting user satisfaction of test data for ICFM, EICFM and LRM. Recall that Q3 and Q4 in the questionnaire (see Table V) covering the user's expectation and overall impression are both related to user satisfaction. Based of the method of computing user satisfaction in [11], we obtain a user satisfaction score by averaging the responses to Q3 and Q4 for each dialog.

Recall that in EICFM, the final rating of a target dialog is a linear combination of user-related and system-related ratings, i.e., $r = r^u * w + r^s * (1 - w)$. Fig. 3 shows R^2 of EICFM in relation to the weight w for different numbers of clusters. As can be seen, EICFM achieves the best performance when $w = 0.1$ for different numbers of clusters, and the prediction performance goes down for values of w above 0.1. The result suggests that system-related features are more helpful than user-

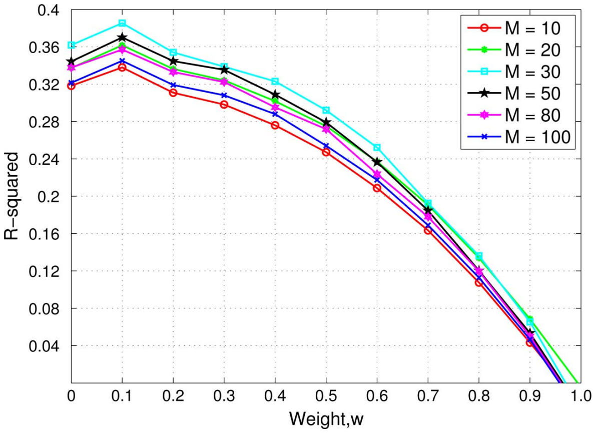

Fig. 3. R^2 of EICFM for user satisfaction prediction in terms of w.

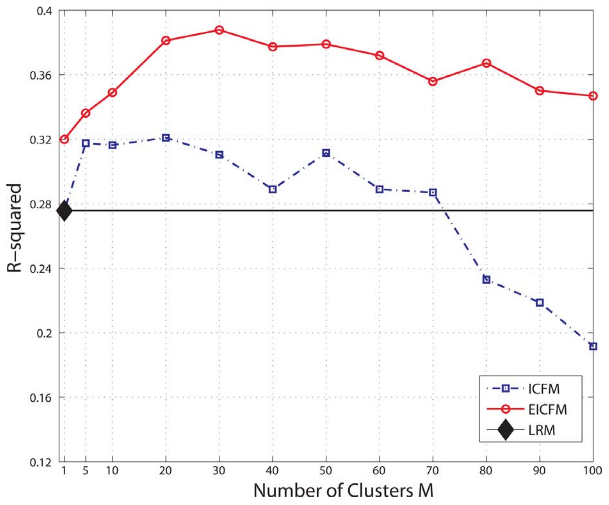

Fig. 4. R^2 for user satisfaction prediction, in relation to the number of clusters M for the three prediction models.

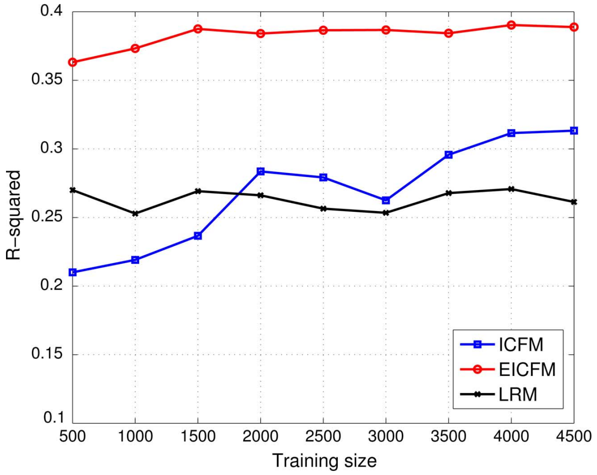

Fig. 5. R^2 for user satisfaction prediction, in relation to the number of data points in the training set for the three prediction models ($M = 30$).

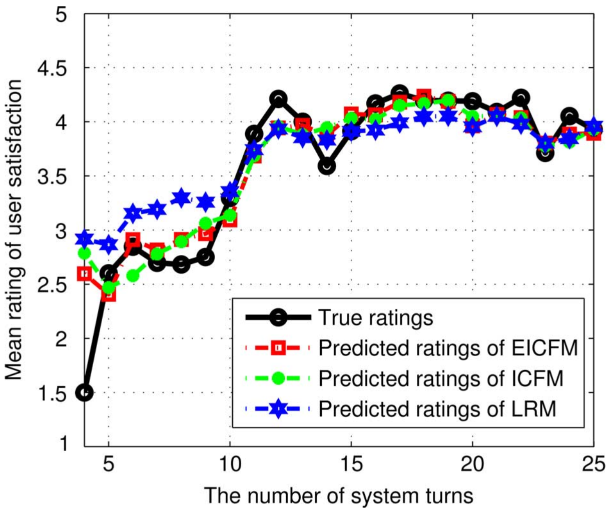

Fig. 6. Average ratings of user satisfaction for dialogs over #System Turns.

related ones in determining user satisfaction. Results of EICFM presented below are all with $w = 0.1$.

Fig. 4 shows R^2 of predicting user satisfaction varying with the number of clusters M for the three models. Since LRM is unrelated to the number of clusters, we represent the LRM result with a single diamond at $M = 1$. We observe that ICFM outperforms LRM for most values of M, and EICFM has the best performance throughout. In particular, when $M = 30$, R^2 values for EICFM, ICFM and LRM are 0.39, 0.31 and 0.27 respectively. Based on t-test, we find EICFM significantly outperforms LRM throughout, and ICFM significantly improves performance when $M >= 5 (p < 0.01)$. Fig. 5 presents the performance of three models with varying training size, when $M = 30$ for the CF models. EICFM achieves the best performance, indicating its robustness to data size. In contrast, ICFM requires more data, outperforming the baseline with 2,000 or more training examples.

The improved performance from our CF models may result from the fact that local information is used to predict the ratings, rather than information from the entire database, which may introduce noise. EICFM outperforms ICFM, possibly due to EICFM's better balance between user and system-related influences on overall ratings. In addition, EICFM is more robust to the number of clusters than ICFM. The R^2 for ICFM drops below that of LRM when $M > 70$, while the performance of EICFM remains stable. This drop is reasonable - since we need sufficient samples to train a good regression model, the error increases as the number of clusters increases and the number of samples in each cluster decreases. The number of feature dimensions in ICFM is larger than that in EICFM, implying that ICFM needs more samples per cluster.

In **Experiment II**, the three prediction models ($M = 30$ for both ICFM and EICFM) are trained on 4,000 dialogs, and are tested on the remaining 1,000 ones. We compare the average values of predicted and true ratings over **#System Turns**. In other words, the ratings are averaged over dialogs sharing the same **#System Turns**. This method compares ratings for groups of dialogs rather than single ones [38].

Fig. 6 shows that both ICFM and EICFM can better reproduce the relations between ratings of user satisfaction and **#System Turns** than LRM. However, all the three models show a larger divergence between true ratings and predicted ones when **#System Turns** ≤ 4. This divergence may be caused by the fact that there are fewer training dialogs (around 10) of this length, yielding poor models.

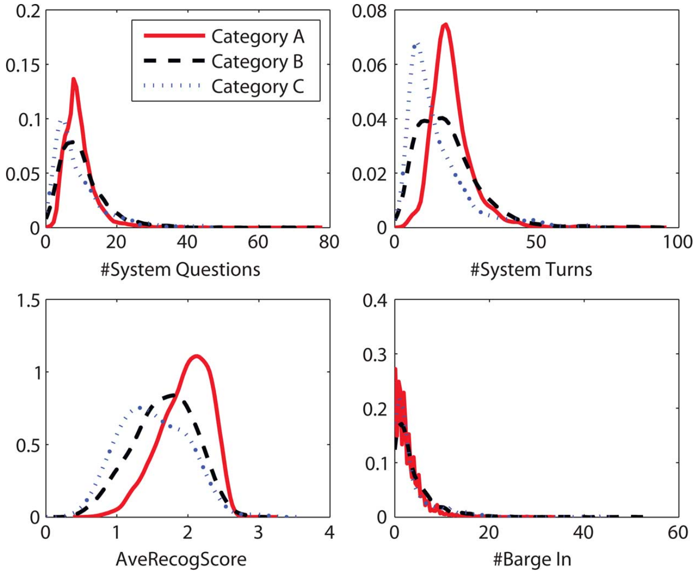

Fig. 7. The probability density plots of **#System Questions**, **#System Turns**, **AveRecogScore** and **#Barge In** for dialogs rated high (A), medium (B) and low (C). The plots of other features are similar to that of **#Barge In**.

Moreover, the plots of true ratings and predicted ones from ICFM and EICFM all show that the ratings of user satisfaction are at a low level (less than 3) and decrease when **#System Turns** <10. This is a reasonable result, if we consider the characteristics of the Let's Go! system. As Table I shows, the system has to obtain enough information from the user (including the bus number, origin, destination and departure time) before it can retrieve information from the database and provide the appropriate response. After the user provides the requested information, the system also has to confirm each piece of information according to an explicit confirmation strategy. Hence, due to the design of the dialog model, dialogs with fewer system turns (less than 10) are prone to failure and consequently get low ratings of user satisfaction.

B. *Analysis of Prediction Results*

To better understand the relations between user satisfaction and the dialog metrics, we analyze the prediction results from EICFM. Based on the predicted ratings of 1,000 dialogs from EICFM in **Experiment II** in Section V.A, we divide the evaluated dialogs into three categories: A (ratings in [3.5,5]), B (ratings in [2.5,3.5)) and C (ratings in [1,2.5)). Fig. 7 shows the probability density plots of 4 different features for dialogs in each category. We can see that the densities of A, B and C on **#System Questions #System Turns**, and **AveRecogScore** are different, which implies that the three features have relatively larger impact on user satisfaction. Dialogs with appropriately higher values for these features tend to get higher ratings.

In addition, we inspect the dialogs in each category and obtain some interesting observations for category A. We originally hypothesized that a dialog would be more efficient (i.e., a shorter dialog with a completed task) and get a higher rating of user satisfaction if the user were "aggressive" in trying to provide all related information at once, such as "I want to take bus $61C$ from the airport to murray at 10 p.m.". However, there are only a very few such dialogs in category A. Most of the dialogs in category A show that the users provide one piece of information per dialog turn, guided by the system. Table I shows a typical example. As can be seen, our original hypothesis seems invalid because there tend to be more recognition errors in longer utterances when the user includes more information in a single turn. These recognition errors reduce the users' overall impressions of system quality.

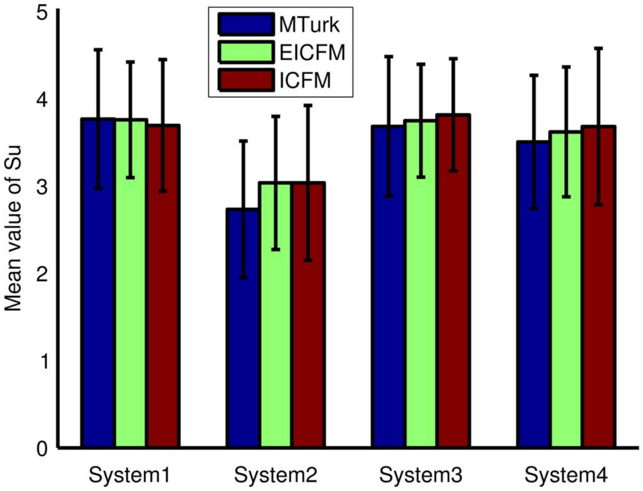

Fig. 8. The mean scores of Su of SDC systems from CF and MTurk.

C. *Verifying the Generalizability of the CF Model*

In Section V.A, we trained the CF model based on one set of Let's Go! dialogs and tested its prediction accuracy on another set of Let's Go! dialogs, training and testing on the same system. In this section, we assess the *generalizability* of CF model across multiple systems (within the same domain) by applying the CF model built on the Let's Go! dialogs to the SDC 2010 corpus [36] introduced in Section III.A.

We restrict our training set to the 5,000 Let's Go! dialogs mentioned in Section V.A, and test the trained ICFM and EICFM $(\mathrm{M} = 30)$ on the dialog corpus of the four systems in SDC 2010 separately. All 310 dialogs of the four SDC systems are rated by MTurk Workers as described in Section III.B, and these ratings will serve as ground truth.

The user satisfaction (S_u) rating for each dialog (both training and testing) is again obtained by averaging the Workers' answers to Q3 and Q4, and is on a 5-point scale. Dialog features include **TaskCompletion**, **#SystemTurns**, **#UserTurns**, **#SystemQuestions**, **#UserQuestions**, **#Help** and **WPUT** (see Table VI). These features are shared by the Let's Go! system and the SDC systems, so they are not exactly the same as those in Section IV.C. Except for **TaskCompletion** which is taken from Workers' annotations of Q2, the remaining features are automatically extracted from log files.

Fig. 8 shows the mean S_u of the four SDC systems predicted by ICFM and EICFM and rated by MTurk Workers, respectively. We observe that except for System 2, the ratings from our CF model and MTurk are quite close for the other three systems. The maximum difference between predicted and crowdsourced ratings is 0.4, and the minimum difference is 0. Moreover, the mean rating from EICFM is closer to the mean crowdsourced rating than that generated by ICFM.

TABLE VII
MSE BETWEEN PREDICTED AND MTURK RATINGS USING ICFM AND EICFM FOR ALL SDC 2010 SYSTEMS

	System1	System2	System3	System4
Total dialogs	91	61	75	83
LRM	1.25	1.32	0.96	1.23
ICFM	1.11	0.97	0.81	1.12
EICFM	0.96	1.00	0.79	0.89

In addition, we also investigate mean square error (MSE) of the CF model and LRM on predicting the ratings of the dialogs of the four SDC systems, where $\mathrm{MSE} = 1/N \sum_{i=1}^{N}(\hat{r}_i - r_i)^2$, N is the total number of dialogs, $\hat{r}$ is the predicted rating and r is the true rating from MTurk. As shown in Table VII, the MSE values of the CF model for the four systems are lower than those of LRM. EICFM achives the lowest MSE of 0.7989 for System 3. EICFM generally has better performance than ICFM. Though trained on a single system, our CF model generalizes well to multiple systems within the bus information domain.

VI. CONCLUSION

This paper proposes a collaborative filtering (CF) model to predict user satisfaction in SDS evaluation. This approach is inspired by the use of CF in recommendation systems, where the preference of a user for a new item is assumed to resemble that for similar items rated previously. In this case, we adapt the idea to predict user evaluations of unrated dialogs assuming that they should be similar to the ratings received by similar dialogs. A reference baseline is provided by a linear regression model (LRM) based on the PARADISE framework. We present two versions of the CF model for SDS evaluation. First, item-based collaborative filtering model (ICFM) clusters rated dialogs and builds a linear regression model for each cluster. A testing dialog then uses the LRM of its most similar cluster to predict user satisfaction. Second, the extended item-based collaborative filtering model (EICFM) separates dialog features into user-related and system-related classes, to build linear regression models for each feature class separately. These models are applied to the dialog corpus from the Let's Go! system and SDC 2010, for which the judgments are collected through crowdsourcing. Experimental results show both ICFM and EICFM can significantly improve the R^2 for prediction on test data when the number of clusters M is set appropriately. Moreover, EICFM exhibits the best performance and is less sensitive to M than ICFM. We also verify the generalization ability of the CF model across multiple systems by training it on the Let's Go! dialogs and testing on the SDC dialogs. Results show that the ratings of the SDC systems predicted by the CF model are closely related to those obtained from crowdsourcing, as measured by difference in mean ratings and mean squared error.

A possible future direction is to extend the current approach and utilize a unified model (e.g., a Bayesian network or a Markov decision process) to replace the linear models of the clusters. This extension will be help to uncover the latent factors that may enable us to gain insight into the influence of different dialog features on overall user satisfaction. Furthermore, we consider applying the CF model to evaluate not just speech-only dialogs, but also multimodal (e.g., speech with gestures) dialogs.

ACKNOWLEDGMENT

This work was done as part of a project related to the Spoken Dialog Challenge in 2010 (http://dialrc.org/sdc) which was organized by Professor Maxine Eskenazi and Professor Alan Black of the CMU Dialog Research Center. It was conducted while Zhaojun Yang was a Masters student and Dr. Gina-Anne Levow was a Visiting Scholar at The Chinese University of Hong Kong. The project team also included Professor Irwin King, Baichuan Li and Yi Zhu from CUHK.

REFERENCES

[1] A. Raux, B. Langner, D. Bohus, A. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Proc. Interspeech*, 2005.

[2] L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann, "Multi-site data collection and evaluation in spoken language understanding," in *Proc. Workshop Human Lang. Technol.*, 1993, pp. 19–24.

[3] H. Meng, P. Ching, S. Chan, Y. Wong, and C. Chan, "ISIS: An adaptive, trilingual conversational system with interleaving interaction and delegation dialogs," *ACM Trans. Computer-Human Interaction (TOCHI)*, vol. 11, no. 3, pp. 268–299, 2004.

[4] Z. Wu, H. Meng, H. Ning, and S. Tse, "A corpus-based approach for cooperative response generation in a dialog system," *Chinese Spoken Lang. Process.*, pp. 614–626, 2006.

[5] D. Litman, C. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman, "Spoken versus typed human and computer dialogue tutoring," *Int. J. Artif. Intell. in Educat.*, vol. 16, no. 2, pp. 145–170, 2006.

[6] S. Möller, "Parameters for quantifying the interaction with spoken dialogue telephone services," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2005, pp. 166–177.

[7] M. McTear, "Spoken dialogue technology: Enabling the conversational user interface," *ACM Comput. Surveys*, vol. 34, no. 1, p. 169, 2002.

[8] H. Bonneau-Maynard, L. Devillers, and S. Rosset, "Predictive performance of dialog systems," in *Proc. Lang. Resources Eval. Conf. (LREC)*, 2000.

[9] S. Moller and J. Skowronek, "Quantifying the impact of system characteristics on perceived quality dimensions of a spoken dialogue service," in *Proc. 8th Eur. Conf. Speech Commun. Technol.*, 2003.

[10] M. Walker, D. Litman, C. Kamm, and A. Abella, "PARADISE: A framework for evaluating spoken dialogue agents," in *Proc. Assoc. Comput. Linguist. (ACL)*, 1997.

[11] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "Evaluating spoken dialogue agents with PARADISE: Two case studies," *Comput. Speech Lang.*, vol. 12, no. 4, pp. 317–347, 1998.

[12] K. Engelbrech, F. Gödde, F. Hartard, H. Ketabdar, and S. Möller, "Modeling user satisfaction with hidden Markov model," in *Proc. SIGDIAL Workshop Discourse and Dialogue*, 2009, pp. 170–177.

[13] I. McGraw, C. Y. Lee, L. Hetherington, and J. Glass, "Collecting voices from the crowd," in *Proc. Lang. Resources Eval. Conf. (LREC)*, 2010.

[14] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, "Cheap and fast—But is it good?: Evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empir. Meth. in Nat. Lang. Process.*, 2008, pp. 254–263.

[15] S. Novotney and C. Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Proc. Assoc. Comput. Linguist. (ACL)*, 2010, pp. 207–215.

[16] Z. Yang, B. Li, Y. Zhu, I. King, G. Levow, and H. Meng, "Collection of user judgments on spoken dialog system with crowdsourcing," in *Proc. IEEE Workshop Spoken Lang. Technol. (SLT)*, 2010, pp. 277–282.

[17] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, 2009.

[18] J. Wang, A. De Vries, and M. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. SIGIR*, 2006, pp. 501–508.

[19] R. Jin, J. Y. Chai, and L. Si, "An automatic weighting an automatic weighting an automatic weighting scheme for collaborative fltering," in *Proc. SIGIR*, 2004.

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. World Wide Web (WWW)*, 2001, p. 295.

[21] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[22] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, pp. 734–749, 2005.

[23] Z. Yang, Y. Zhu, B. Li, I. King, G. Levow, and H. Meng, "Collaborative filtering model for user satisfaction prediction in spoken dialog evaluation," in *Proc. IEEE Workshop Spoken Language Technol. (SLT)*, 2010, pp. 472–477.

[24] K. Hone and R. Graham, "Subjective assessment of speech-system interface usability," in *Proc. 7th Eur. Conf. Speech Commun. Technol.*, 2001.

[25] I. Rec. "P. 851. Subjective quality evaluation of telephone services based on spoken dialogue systems," Int. Telecomm. Union, Geneva, 2003.

[26] J. Glass, J. Polifroni, S. Seneff, and V. Zue, "Data collection and performance evaluation of spoken dialogue systems: The MIT experience," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, 2000.

[27] M. McTear, "Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit," in *Proc. ICSLP*, 1998, pp. 1223–1226.

[28] "Parameters describing the interaction with spoken dialogue systems," ITU. Geneva, 2005, I. P. series Rec.

[29] P. Cohen, *Empirical Methods for Artificial Intelligence*. Cambridge, U.K.: MIT Press, 1995, vol. 55.

[30] K. Forbes-Riley and D. Litman, "Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters," in *Proc. Assoc. Comput. Linguist. (ACL)*, 2006, pp. 264–271.

[31] M. Walker, R. Passonneau, and J. Boland, "Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems," in *Proc. Assoc. Comput. Linguist. (ACL)*, 2001, p. 522.

[32] C. Kamm, M. Walker, and D. Litman, "Evaluating spoken language systems," in *Proc. AVIOS*, 1999.

[33] M. Walker, C. Kamm, and D. Litman, "Towards developing general models of usability with PARADISE," *Nat. Lang. Eng.*, vol. 6, no. 3&4, pp. 363–377, 2000.

[34] N. Beringer, U. Kartal, K. Louka, F. Schiel, and U. Türk, "Promise: A procedure for multimodal interactive system evaluation," in *Proc. LREC Workshop Multimodal Resources and Multimodal Syst. Eval.*, 2002, pp. 90–95.

[35] L. Dybkjaer, N. Bernsen, and W. Minker, "Evaluation and usability of multimodal spoken language dialogue systems," *Speech Commun.*, vol. 43, no. 1–2, pp. 33–54, 2004.

[36] A. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi, "Spoken dialog challenge 2010," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2010, pp. 448–453.

[37] M. Shoukri, *Measures of Interobserver Agreement*. Boca Raton, FL: CRC, 2004.

[38] K. Engelbrecht and S. Möller, "Pragmatic usage of linear regression models for the prediction of user judgments," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2007.

Zhaojun Yang is a Ph.D. student in electrical engineering at the University of Southern California. She received her B.E. Degree in electrical engineering from the University of Science and Technology of China 2009 and M.Phil. Degree in systems engineering and engineering management from the Chinese University of Hong Kong in 2011. Her research interests include spoken dialog system and multimodal signal processing.

Gina-Anne Levow earned her Ph.D. (1998) and Master's (1993) in computer science from the Massachusetts Institute of Technology and her B.A.S. in computer science and B.A. in oriental studies (1989) from the University of Pennsylvania. She is currently an Assistant Professor in the Department of Linguistics at the University of Washington. Her research interests include spoken language understanding and multimodal and spoken dialog systems.

Helen Meng (M'98-SM'09) received the S.B., S.M., and Ph.D. degrees, all in electrical engineering, from the Massachusetts Institute of Technology. She joined The Chinese University of Hong Kong in 1998, where she is currently Professor and Chairman in the Department of Systems Engineering and Engineering Management. She was also Associate Dean of Research of the Faculty of Engineering between 2005–2010. Her research interest is in the area of human-computer interaction via multimodal and multilingual spoken language systems, speech retrieval technologies and computer-aided pronunciation training. Prof. Meng served as Editor-in-Chief for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING (2009–2011). She is also an elected board member of the International Speech Communication Association.