COLLABORATIVE FILTERING MODEL FOR USER SATISFACTION PREDICTION IN SPOKEN DIALOG SYSTEM EVALUATION

Zhaojun Yang[1], *Baichuan Li*[2], *Yi Zhu*[2], *Irwin King*[2], *Gina Levow*[3‡], *Helen Meng*[1†]

[1]Department of System Engineering and Engineering Management
[2]Department of Computer Science and Engineering
[1,2]The Chinese University of Hong Kong, Shatin.N.T., Hong Kong
[3]Department of Linguistics, University of Washington, Seattle, WA 98195 USA

ABSTRACT

Developing accurate models to automatically predict user satisfaction about the overall quality of a Spoken Dialog System (SDS) is highly desirable for SDS evaluation. In the original PARADISE framework, a linear regression model is trained using measures drawn from rated dialogs as predictors with user satisfaction as the target. In this paper, we extend PARADISE by introducing a collaborative filtering (CF) model for user satisfaction prediction and its corresponding extension. This prediction model is drawn from the idea of CF in recommendation systems, which uses information from near neighbors of an unrated dialog to predict its user satisfaction. We also present the methodology of collecting user judgments on SDS quality with crowdsourcing through Amazon Mechanical Turk. Experimental results show that the CF approaches could distinctly improve the prediction accuracy of user satisfaction.

Index Terms— spoken dialog system, user satisfaction prediction, collaborative filtering, item-based, Let's Go

1. INTRODUCTION

Spoken dialog systems (SDSs) have been widely used in many different domains, like bus schedule inquiries, financial information delivery, restaurant guides, etc. Accordingly, the diversity calls for sound strategies in evaluating, comparing, and predicting the performance of SDSs.

Initially, SDS evaluation primarily focuses on the performance of the individual SDS components, including the speech recognition accuracy, ability to understand natural language, or the naturalness of synthetic speech. Such performance metrics of individual components are well-developed [1]. However, as systems become more complex and their components are integrated in more intricate ways, it becomes difficult to use component-based evaluation, especially for comparing two systems with different components. In contrast, holistic evaluation, which assesses the overall quality of an SDS taking into account the performance of individual components, is more appropriate. The overall quality of an SDS is usually measured by asking users to fill out a questionnaire after they interact with the system. The questionnaires often involve user perceptions of the system, such as task completion, system usability, or system intelligence.

However, inviting human users in SDS evaluation is a costly process. This motivates the design of automatic evaluation strategies for holistic evaluation [2]. The PARADISE framework proposed in [3] attempts to automatically predict user satisfaction for unrated dialogs, assuming that user satisfaction can describe the overall quality of a system. A linear regression model is trained using measures of rated dialogs as predictors with user satisfaction as the target. PARADISE has been widely applied in evaluating many SDSs, such as the IT-SPOKE tutoring system [4], DARPA Communicator [5], etc. Despite its popularity, the predictive power is limited, especially on test data, with R^2 around 0.22 [6]. When we apply the original PARADISE framework on the Let's Go dialog corpus [7], R^2 also stays around 0.27 test data prediction. Low R^2 may be caused by the lack of inter-rater agreement on user satisfaction ratings [8] or the linear model may be insufficient in capturing the relations between user satisfaction and dialog features.

In this paper, we extend the PARADISE framework by introducing collaborative filtering (CF). CF has been successfully applied to the development of recommendation systems [9]. It assumes that a user's preference for a new item may resemble that for the similar items rated previously, which also holds for automatic evaluation of SDSs. In our work, we develop a basic CF model for user satisfaction prediction, which uses information from near neighbors of an unrated dialog to predict its user satisfaction. Then the basic model is extended by considering user judgments with respect of *user style* and *system quality*. Experiments demonstrate that our CF models both outperform the linear regression by a large margin. To the best of our knowledge, this is the first

†H. Meng is the corresponding author.

‡Dr. Gina Levow was involved in this work while she was a Visiting Scholar at The Chinese University of Hong Kong.

time to introduce CF into SDS evaluation problems. Building a general prediction model requires a rich and statistically representative training set. To solve this problem, we collect user judgments of SDS quality with crowdsourcing on Amazon Mechanical Turk (MTurk) [1]. Crowdsourcing has recently gained popularity in speech and language data collection/annotation/evaluation [2] [10]. This work attempts to use crowdsourcing for *dialog evaluation*.

The rest of the paper is organized as follows. Section 2 gives a brief introduction about CF and details our approach. We describe the corpus and feature extraction procedure in Section 3, which involves collection of user judgments with crowdsourcing through MTurk. Experimental results are shown and analyzed in Section 4. Finally, we end with conclusions and an outlook on future work.

2. CF MODEL FOR USER SATISFACTION PREDICTION

CF uses a database of users' preferences for items to predict the utility of a certain item for a particular user. In this section, we firstly introduce an implementation of CF from which we develop our approaches, i.e., item-based CF, followed by the description of our basic and extended SDS evaluation models.

2.1. Item-Based CF

Item-based techniques are one main category among CF's implementations. These methods search for items most similar to the target one in a data set which has been rated by users. Prediction of the target item is then computed based on similar ones. Item-based CF is computationally efficient and can guarantee recommendation quality [11].

Suppose that the k most similar items of the target i are selected for the active user u, and their ratings by u are denoted as $\{r_{u,j}\}_{j=1}^{k}$. 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 k similar items,

$$P_{u,i} = \frac{\sum_{j\in\{k \text{ similar items}\}} s_{i,j} * r_{u,j}}{\sum_{j\in\{k \text{ similar items}\}} s_{i,j}}, \quad (1)$$

where the weights $\{s_{i,j}\}_{j=1}^{k}$ are similarities between i and the k items. For some more elaborate algorithms for item-based CF we refer readers to [9].

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 items by some common features (see Section 3.1), 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 dialog for a particular user u, but rather for a general population of users.

2.2. 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 $\mathbf{f}_i$ which has been normalized to its z score, and the similarity between two dialogs d_i and d_j is measured as the cosine similarity of their feature vectors,

$$s_{i,j} \doteq s(d_i, d_j) = \frac{\mathbf{f}_i^T \mathbf{f}_j}{|\mathbf{f}_i| * |\mathbf{f}_j|}. \quad (2)$$

To save computation time, we cluster dialog corpus using k-means in advance. Let $C = \{C_i\}_{i=1}^{M}$ be the clusters created from D such that $\cap_i C_i = \phi$ & $\cup_i C_i = D$. Therefore, the retrieval process of k similar dialogs for the target dialog d relates to its assignment to a cluster C^*,

$$C^* = \arg\max_{C_i} s(d, c_i), \quad (3)$$

where c_i is the centroid of C_i.

Sarwar et al. pointed out that two items with high similarity may be distant in Euclidean distance [11], therefore they proposed to map the known rating $r_{u,j}$ in Eq. 1 to $g(r_{u,j})$. When $g(\cdot)$ is a linear mapping, it reduces to the linear regression problem. Hence we use linear regression trained on the selected cluster C^* to predict the rating for the target dialog d, rather than use the weighted sum (see Eq. 1). 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 below,

1. Extract feature vector $\mathbf{f}_i$ for each dialog $d_i \in D$.
2. Use k-means to create dialog clusters C for the dialog corpus D based on the feature representations $\mathbf{f}$ and the similarity measure in Eq. 2.
3. Build linear regression models $L = \{L_i | r = L_i(\mathbf{f})\}_{i=1}^{M}$ for the created clusters, which means model L_i is trained from dialogs in cluster C_i.
4. Given an unseen dialog d (unevaluated dialog here), we first extract a feature vector $\mathbf{f}_d$ and then assign d into cluster C^* with Eq. 3.
5. Use L^* which is trained on C^* to predict user satisfaction for d.

[1] http://www.mturk.com

[2] See NAACL-HLT 2010 Workshop on the use of MTurk for speech and language collection/annotation/evaluation.

2.3. Extended ICFM for User Satisfaction Prediction

By considering the characteristics of dialogs which record interactions between users and an SDS, we find that the features extracted (see Section 3.1) can be separated into *user-related* and *system-related* types. For example, #Barge In (overall number of user's barge in attempts) reflects the characteristics of user behavior and can be classified as a user-related feature, while #System Question (overall number of system's questions in the dialog) is a system-related feature. The intuition for this separation is that 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 tastes 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 the user style can be obtained from user-related features and those due to the system quality can be drawn from system-related ones. Hence, we can predict judgement ratings based on the two types of features *separately*, rather than on the basis of the entire feature set. Based on this idea, we extend ICFM to EICFM as follows,

1. Create system-related clusters C^s for dialog corpus D based on system-related features $\mathbf{f}^s$.
2. Create user-related clusters C^u for D based on user-related features $\mathbf{f}^u$.
3. Build linear regression models L^s and L^u for C^s and C^u respectively.
4. Given an unseen dialog d, choose C^{s*} and C^{u*} which are most similar to d with respect to system-related features $\mathbf{f}^s$ and user-related ones $\mathbf{f}^u$, respectively.
5. Use regression model L^{s*} to predict system-related judgement r^s for d, and use model L^{u*} to predict user-related judgement r^u.
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. This weight is determined by a validation set in our experiments.

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

3. EXPERIMENTAL CORPUS

In this section, we describe the methodology of collecting user judgments on the dialog corpus D used in our experiments, as well as how to extract the feature vector $\mathbf{f}_i$ for each dialog d_i in D.

The dialogs used in this paper are drawn from the log files of the Let's Go system developed by Carnegie Mellon University [7]. This system provides bus schedule information for Pittsburgh residents when the customer service hotline is not staffed. Input utterances from users are transcribed automatically by automatic speech recognition (ASR). Table 1 shows an example of the Let's Go dialogs.

Table 1. *An example of the Let's Go dialogs.*

S: *Welcome to the CMU Let's Go bus information system*
S: *What can I do for you?*
U: *When is the next 64A from Forbes and Murray to East Liberty?*
S: *The 64A. Did I get that right?*
U: *Yes.*
S: *Leaving from Forbes and Murray. Correct?*
U: *Yes.*
S: *Going to East Liberty. Am I right?*
U: *Yes*
S: *I think you want the next bus.*
U: *Yes*
S: *Hold on. I'll look that up.*
S: *The next 64A leaves Murray Avenue at Forbes at 10:32p.m. and arrives at Highland Avenue at Penn Avenue at 10:55p.m.*

3.1. Feature Extraction

According to ITU Recommendation [12], we extract some interaction features, whose meanings are illustrated in Table 2, from the log files for each dialog. The features of #System Turns, #User Turns, AveRecogScore, #Barge In, and #Help Requests were used in the original version of the PARADISE model [3], while #DTMF is specific to the Let's Go system since it provides touch tone functionality to users.

Table 2. *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 user's
AveRecogScore	Average recognition score
#Barge In	Overall number of user's barge in attempts
#Help Requests	Overall number of user's help requests
#User Questions	Overall number of user's questions
#System Questions	Overall number of system's questions
#DTMF	Overall number of touch tone uses

Among these features, #System Turns, AveRecogScore, and #System Questions are classified as *system-related* ones for they are mostly influenced by the characteristics of the system, while the others are determined by user behavior and are hence *user-related*. All features use z-norm scores in the following experiments.

3.2. User Judgment Collection with Crowdsourcing

In order to build a general prediction model, we need plenty of dialogs rated by as many people as possible to obtain a statistically representative set. MTurk provides an effective platform for such tasks. It is a crowdsourcing marketplace that utilizes human intelligence online to perform tasks which cannot be completed entirely by computer programs.

The MTurk platform organizes the work in the form of human intelligence tasks (HITs). The HITs in our work are designed to outsource the assessment of the SDS to MTurk Workers. To achieve this goal, we have authored a set of questions that constitute the HITs shown in Table 3. The questions cover user's confidence, perception of task completion, user's expectation, overall performance, and the categorization of task success. Answer options to Q1-Q4 are on a 5-point scale, from 1 for the worst to 5 for the best, while answer options to Q5 refer to the definition of task success in [12] and are on a 7-point scale. Each HIT consists of the text transcription of one dialog and one such questionnaire to collect the Worker's assessments. More details about the collecting process can be found in [13]. In about 45 days, we collect more than 5,000 dialog ratings in total.

Table 3. *Questions constituting the HITs for SDS Evaluation.*

Q1	Do you think you understand from the dialog what the user wanted?
Q2	Do you think the system is successful in providing the information that the user wanted?
Q3	Does the system work the way you would expect it?
Q4	Overall, do you think that this is a good system?
Q5	What category do you think the dialog belongs to?

As this is among the early attempts of using crowdsourcing for *spoken dialog evaluation*, we present some interesting observations in this procedure which may facilitate similar work in the future. We use Cohen's weighted kappa to measure the inter-rater agreement for each question. Q2 and Q5 achieve values around 0.5, indicating moderate inter-rater agreement. Q2 and Q5 are about task completion which can gain "official" or somehow objective ratings from reliable raters, so the moderate agreement partially shows the reliability of MTurk Workers and provides support for the utilization of MTurk as a judgment collection platform. On the other hand, Q3 and Q4 have low values below 0.3, which is indicative of a lack of agreement. Recall that Q3 and Q4 are both about user satisfaction (see Table 3), so the low agreement shows the diversity in human perceptions and may lead to low predictive accuracy measured with R^2 as analyzed in [8].

As will be seen in Section 4, our data is used in a 10-fold cross validation style in the first experiment. In the second step, we divide the corpus into training and test set, containing 4,000 and 1,000 dialogs respectively.

4. EXPERIMENTAL RESULTS AND ANALYSIS

We conduct two experiments to investigate how ICFM and EICFM can improve user satisfaction prediction. **Experiment I** compares R^2 in predicting user satisfaction for ICFM, EICFM, and the linear regression model (LRM) using 10-fold cross validation. **Experiment II** is to compare 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 this feature 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. The weighting w is set to 0.1 empirically through the use of a validation set. We use R^2 to measure the prediction accuracy in **Experiment I**,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{\sum_{i=1}^{n}(r_i - \bar{r})^2}, \tag{4}$$

where 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 $\{r_i\}_{i=1}^{n}$. The higher R^2 is, the higher the prediction accuracy is.

4.1. Prediction of User Satisfaction

In **Experiment I**, we use 10-fold cross validation on the data corpus (5,000 rated dialogs, see Section 3.2) 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 3) cover the user's expectation and overall impression, therefore the responses to Q3 and Q4 for each dialog are averaged to yield a single user satisfaction rating ranging from 1 to 5 as the output of the prediction model, while the input is the 10-dimensional feature vector introduced in Section 3.1.

Fig. 1 shows R^2 of predicting user satisfaction changing with the cluster number M for the three prediction models. Since LRM is unrelated to the cluster number, 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.

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 entire database, which may introduce noise to the prediction. Compared with ICFM, EICFM is even better, which may be due to EICFM's having a better balance between the influences from user style and system quality on the overall judgment about the system. 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 keeps stable within the same scale. This drop is reasonable. Because it requires certain number of samples to train a good regression model, the error increases as the cluster number

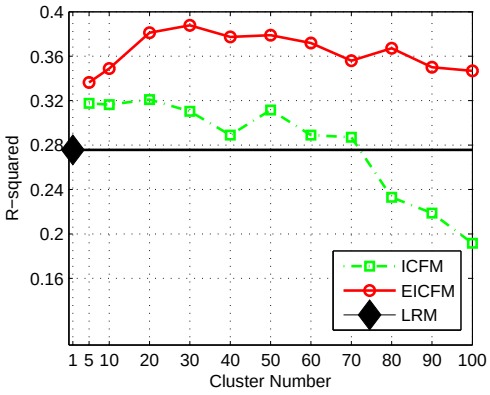

Fig. 1. *R^2 for user satisfaction prediction, in relation to the number of clusters M for the three prediction models. EICFM shows distinct improvement and is less sensitive to M.*

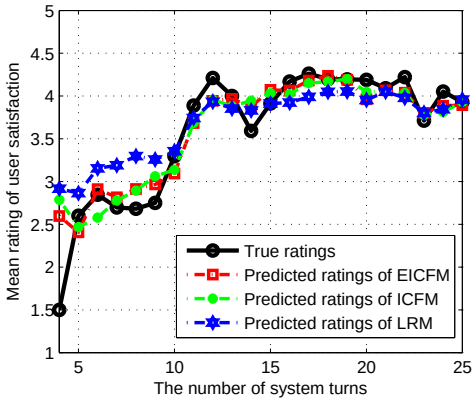

Fig. 2. *Average ratings of user satisfaction for dialogs over different #System Turns. The solid line with circle markers is for true ratings, and the other lines are for the predictions.*

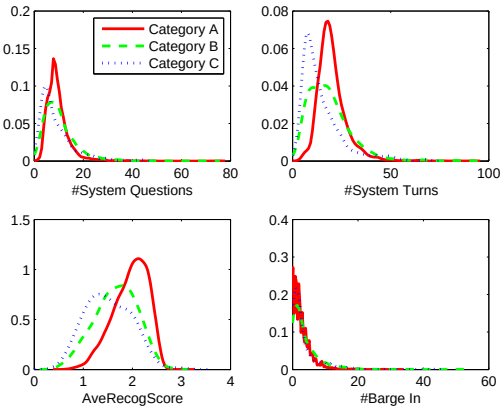

Fig. 3. *The probability density plots of #System Questions, #System Turns, AveRecogScore, and #Barge In for dialogs rated high (A), medium (B), and low (C). The plots of other features are similar to that of #Barge In.*

becomes larger (hence samples in each cluster decrease). Further, ICFM drops quicker than EICFM due to its longer input feature vector, i.e., more samples are required in general.

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 [6].

Fig. 2 shows that both ICFM and EICFM can better reproduce the relation 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 such training dialogs (around 10), which makes the prediction models do not fit well when #System Turns ≤ 4.

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 by considering the characteristics of the Let's Go system. As Table 1 shows, the system has to get enough information from the user, such as the bus number, the origin, destination, and departure time, in order to retrieve information from the database and provide corresponding results. 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 manager, the dialogs with fewer system turns (less than 10) prone to failure and get low ratings of user satisfaction.

4.2. Analysis of Prediction Results

To better understand the relations between user satisfaction and dialog characteristics, we analyze the prediction results from EICFM. Based on the prediction ratings of 1,000 dialogs from EICFM in **Experiment II** in Section 4.1, 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. 3 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 AveRecogScore, #System Turns, and #System Questions are different, which implies the three features have relatively larger impact on user satisfaction. Dialogs with appropriately higher values of these features tend to get higher ratings. The value of weight w in EICFM is 0.1, which indicates that system-related features are more helpful than user-related ones in determining user satisfaction.

In addition, we analyze the dialog discourse in each category and obtain some interesting observations in category A. We originally hypothesize that a dialog would be more efficient (i.e., a shorter dialog with task completed) and get 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 airport to murray at 10p.m.".

However, in A category there is only a small portion of such dialogs, as shown in Table 1. Most of the dialogs in category A show the users provide one piece of information at a time, and they are guided by the system step by step. Table 4 shows such an example which begins with bus information in user's turn. This discourse is the most typical of category A. Our hypothesis may be invalid because there tends to be more recognition errors in longer interactions when the user includes more information in a single turn. These recognition errors reduce users' overall impressions of the system quality.

Table 4. *A dialog example with a discourse that is the most typical of category A (the highly rated dialogs).*

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.*

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an item-based CF model (ICFM) for spoken dialog system evaluation, which is motivated by the idea of item-based recommendation that the rating of a dialog can be estimated from those of similar ones. In addition, ICFM is extended to EICFM by considering user judgments with respect of *user style* and *system quality*. These models are applied to the dialog corpus from the Let's Go system, the user judgments of which are collected with crowdsourcing on MTurk.

Experimental results show both ICFM and EICFM can significantly improve the R^2 for prediction on test data when the cluster number M is set appropriately. In particular, R^2 for EICFM, ICFM, and LRM are 0.39, 0.31, and 0.27 respectively when $M = 30$. Moreover, EICFM performs the best and is less sensitive to M than ICFM.

In future work, more features will be explored to capture the system quality, such as the appropriateness of system utterances in the current dialog context, the system's ability to recover from errors, etc. Furthermore, deeper analysis of the prediction process will be conducted to gain insight into how different features influence the overall user satisfaction.

6. ACKNOWLEDGEMENT

The project team is a participant in the Spoken Dialog Challenge 2010 (http://dialrc.org/sdc). which is organized by Professor Maxine Eskenazi and Professor Alan Black of the CMU Dialog Research Center. The work is partially supported by a grant from the HKSAR Government Research Grants Council (Project No. 415609) and that from MSRA FY09-RES-OPP-103 (Reference No. 6902682). This work is affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies.

7. REFERENCES

[1] S. Möller, "Parameters for quantifying the interaction with spoken dialogue telephone services," in *Proc. of SIGdial*, 2005.

[2] H.W. Hastie, R. Prasad, and M. Walker, "What's the trouble: automatically identifying problematic dialogues in DARPA communicator dialogue systems," in *Proc. of ACL*, 2002.

[3] M.A. Walker, D.J. Litman, C.A. Kamm, and A. Abella, "PARADISE: a framework for evaluating spoken dialogue agents," in *Proc. of ACL*, 1997.

[4] K. Forbes-Riley and D.J. Litman, "Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters," in *Proc. of HLT on ACL*, 2006.

[5] M.A. Walker, R. Passonneau, and J.E. Boland, "Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems," in *Proc. of ACL*, 2001.

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

[7] 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. of Interspeech*, 2005.

[8] K.P. Engelbrech, F. Gödde, F. Hartard, H. Ketabdar, and S. Möller, "Modeling user satisfaction with Hidden Markov Model," in *Proc. of SIGDIAL*, 2009.

[9] S. Xiaoyuan, M. Taghi, et al., "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, 2009.

[10] I. McGraw, C. ying Lee, L. Hetherington, and J. Glass, "Collecting voices from the crowd," in *Proc. of LREC*, 2010.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. of World Wide Web*, 2001.

[12] ITU P series Rec, "Parameters Describing the Interaction with Spoken Dialogue Systems," *ITU*, 2005.

[13] Z.J. Yang, B.C. Li, Y. Zhu, I. King, G. Levow, and H. Meng, "Collection of User Judgments on Spoken Dialog System with Crowdsourcing," accepted to SLT, 2010.