ADAPTIVE SELECTION FROM MULTIPLE RESPONSE CANDIDATES IN EXAMPLE-BASED DIALOGUE

Masahiro Mizukami[1], *Hideaki Kizuki*[2], *Toshio Nomura*[2], *Graham Neubig*[1]
Koichiro Yoshino[1], *Sakriani Sakti*[1], *Tomoki Toda*[1,3], *Satoshi Nakamura*[1]
[1]Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan.
[2]SHARP Corporation
1-2-3 Shibaura, Minato-ku, Tokyo 105-0023, Japan.
masahiro-mi@is.naist.jp, {kizuki.hideaki, nomura.toshio}@sharp.co.jp,
{neubig, koichiro, ssakti, tomoki, s-nakamura}@is.naist.jp

ABSTRACT

In spoken dialogue systems, dialogue modeling is one of the most important factors for contributing to user satisfaction improvement. Especially in Example-Based Dialogue Modeling (EBDM), effective methods to build dialogue example databases and to select response utterances from examples are the keys for improving dialogue quality. In dialogue corpora, it often have plural appropriate responses for one utterance. However, the system merges these plural appropriate responses into the one system response, thus, it does not try to use plural responses properly by user preference. In fact, responses that each user thinks to be preferable are different. In this paper, we propose a framework that select an appropriate response from plural appropriate response candidates satisfies users. It has a multi-response example database, and selects an appropriate response based on collaborative filtering. Experimental results showed that the proposed framework were successfully choosing appropriate responses, considering multi-response candidates improves user satisfaction to 4.1 from 3.7 of single response, and the adaptive response selection method increased user satisfaction from 3.7 to 4.3.

Index Terms— Example-based Dialogue System, Response Selection, User Adaptation, Collaborative Filtering

1. INTRODUCTION

Example-based dialogue modeling (EBDM) is a data-driven approach for deploying dialogue systems that generates responses from a dialogue example database [1, 2]. Particularly for non-task-oriented dialogues, EBDM framework which chooses appropriate responses from an existing database presents a light-weight yet feasible alternative to more traditional methods (which require separate language understanding, dialogue management, and generation modules). Within the framework, two elements contribute to the success of the dialogue modeling: the size and quality of the example database, and response selection method from the database. Databases are generally constructed from available data sources such as human-to-human conversation log databases [3], movie or television scripts [4, 5], or Twitter logs [6].

Within this framework, the example database consists of pairs of a query and a response $\langle q, r \rangle$. Given a user utterance q', the EBDM system calculates the similarity between q' and every q to fing the most similar query $\hat{q}$ to the user utterance q', and uses the corresponding response $\hat{r}$ to $\hat{q}$. The similarity between q and q' are defined by TF-IDF weighted vector space similarity [7], WordNet-based syntactic-semantic similarity [5], or recursive neural network-based paraphrase detection [8] in previous researches of EBDM.

General example database consist of queries which have a single response r each other, however, this assumption does not always works. These are some queries that can have several response candidates. For example, for the query "What shall we eat today?" we can easily imagine some responses as "Anything to eat is good for me," "Today is cold, let's make stew," or "I know you're going to make me cook dinner," and the choice of a response is affected by some states, such as preference of user.

In this paper, we constructed a example database for EBDM which has multiple responses for one query, and proposed two methods to select the most appropriate response for users to improve their satisfaction. The first method is the *static* which chooses the response has the highest average evaluation score by human annotators (Section 4.2.1). The second is the *adaptive*with the selection score which considers the user's preference (Section 4.2.2).

Specifically, the ADAPTIVE method is based on two components: *predicting user satisfaction* using user feedback and *collaborative filtering*. For predicting user satisfaction, we

[3] He is now at Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan. tomoki@icts.nagoya-u.ac.jp

978-1-4799-7291-3/15/$31.00 ©2015 IEEE

focus on utterances in which the user gives feedback about the system response to estimate satisfaction using a Support Vector Regression [9] model (Section 3). Next, we adapt the response utterances of EBDM with the predicted satisfaction and collaborative filtering [10]. Collaborative filtering is a technique used in recommendation systems to estimate a user's preference based on the preference of other similar users. We apply this technique by comparing estimated satisfaction sequences of users. In the experimental evaluation (Section 5), the *static* method is better than baselines which does not consider multiple response examples, and the *adaptive* method is the best in all models.

2. DATA COLLECTION

2.1. Example Database Collection

As mentioned in the introduction, there are many methods to construct large and high-quality example databases for EBDM. However, it do not try to construct the multiple response example database. In this paper, we followed Murao et al, [3] in simply constructing our database manually, generally used in example database constructing.

Note that each example has responses created by multiple writers. Because of this, each example utterance q has a set of multiple potential responses $\mathbf{r}$, and thus our example database $\mathbf{e}$ can be expressed as a set of pairs of queries and response sets

$$\langle q, \mathbf{r} \rangle \in \mathbf{e}. \tag{1}$$

This is a generalization of traditional EBDM, where each query q has only a single response, and thus $|\mathbf{r}| = 1$.

As our target, we assume a situation where a user talks to a dialogue system on a daily basis, and thus choose utterances that appear in daily life. We create 14 events that the user may experience in a day, and have 7 human utterance writers create utterances to match each event, resulting in a total of 511 unique utterances. To create responses to these utterances, we ask 15 human response writers to fill in blanks following each user utterance, and an example database is created using the examples constructed by each human writer. Out of the 15 examples, there were averages of 12 unique responses for each user utterance. In Table 1, we show an example of an event, utterances and responses created by the writers.

2.2. Satisfaction Annotation

The aim of our proposed method is to find responses that maximize user satisfaction, and thus the next step in our data collection is to collect responses annotated with satisfactions, as well as annotator feedback utterances.

In this annotation, the definition of satisfaction is important. In the well-known PARADISE framework [11, 12] for task-based dialogue, satisfaction is calculated by asking the user several subjective questions after the dialogue completes, and averaging the scores for each question into a total satisfaction score. These questions are related to task success, response delay, response quality, and other topics, with a heavy weight on task success. However, in the case of non-task-oriented dialogue, these questions cannot be applied directly. Therefore, following Yang et al. [13], we judge overall satisfaction with responses with a single question "Do you think that this is a satisfactory response?", and have the user reply to this single question on a 1—6 Likert scale, where 1 is "I don't think so" and 6 is "I think so."

To collect this data, the annotator first views the reply of the dialogue system for each input and decides a *satisfaction score* on a scale of 1–6. The user then can make a *feedback utterance*, which is a verbal expression of their satisfaction with the system's response. We show an example of inputs, system responses, feedback utterances, and annotated satisfactions in Table 2, and we explain these in detail in the following paragraphs.

In most cases it is an unreasonable burden on the user to annotate explicit satisfaction scores while the dialogue progresses. On the other hand, in many cases the user may provide implicit feedback regarding the goodness of the response. For example, when the dialogue system makes a funny joke, the user may laugh or praise the system. These user feedback utterances express the user's opinion or feeling about the response, and it is useful to estimate satisfaction at test time using these utterances, removing the need for explicit annotation. Thus, we propose to predict user satisfaction score directly by analyzing this feedback. The satisfaction prediction method is used with our proposed adaptive method (Section 4.2) which requires knowledge of satisfaction of the actual user. To create the data for the satisfaction prediction method, annotators are told to perform a feedback utterance as a option (non-essential). When there is no annotator feedback, it is treated as an instance of "null" feedback.

For convenience, we define a triplet of user utterance, response utterance, and annotator feedback utterance as a "tri-turn" [5]. In the end, we collected satisfaction annotated tri-turns from 5 annotators for 15 example databases corresponding to each response writer. The corpus totals 2,555 tri-turns including 2,056 non-null annotator feedbacks. We normalize satisfaction by Z-score[1] for each annotator for the purpose of reducing differences between annotators. In the annotation, annotators viewed response and gave feedback by using a text-based chat interface, using each database separately (explained in Section 4.1).

3. PREDICTING SATISFACTION FROM FEEDBACK

In this section, we propose a method for predicting numerical user satisfaction score using user feedback utterances. Some

[1] Z-score is a method that normalizes score so $\mu = 0, \sigma^2 = 1$.

Table 1. Examples of events and pairs of utterance and responses (translated from Japanese)

Events	Utterances	Responses
Eating dinner	What shall we eat today?	Today is cold, let's make stew. Anything to eat is good for me. I know you're going to make me cook dinner.
	Let's eat.	Yes, go ahead. Please eat a lot. Sure, let's eat.
	Let's have some liquor.	What will you have to drink? Me too. Try to drink in moderation.

Table 2. A sample of tri-turns and annotation results (translated from Japanese)

User utterance	System response	User feedback	Satisfaction
Do I have any plans today?	Please check your calendar.	No, you tell me!	1
Please be quiet.	Umm... I'm sorry...	"null"	4
What shall we eat today?	Today is cold, let's make stew.	Nice idea!	6

previous works tried to predict user satisfaction using N-gram-based dialog history [14], collaborative filtering [13], or analyzing "competence" and "certainty" [15]. However, these works predicted the satisfaction in batch processing after the each dialogue. In contrast, our method predicts turn-by-turn while the dialogue is progressing for use in response selection.

We predict user satisfaction in each tri-turn using Support Vector Regression [9], which has proven effective in previous work on dialogue quality estimation [16]. For the t-th tri-turn in the training data, we have a labeled satisfaction score s_t which is to be estimated by a regression model $\mathrm{R}(m_t)$ given the user feedback utterance m_t as input. As input variables of the regression, we use occurrences of words, word classes defined by Japanese Word Net [17], and sentiment orientation scores calculated by a sentiment lexicon [18]. Specifically, we use the following features:

- Flag about whether user feedback m_t exists or not. $f_{m_t} \in \{0, 1\}$
- Counts of n-grams in user feedback m_t. $\mathbf{w_t} = \{w_{t,1}, w_{t,2}, \ldots, w_{t,N}\}$
- Counts of word classes in user feedback m_t. $\mathbf{c_t} = \{c_{t,1}, c_{t,2}, \ldots, c_{t,M}\}$
- Flag about whether a word in the sentiment lexicon s_t exists in user feedback m_t or not. $f_{s_t} \in \{0, 1\}$
- Vector containing maximum, smallest and average of sentiment scores for user feedback m_t. $\mathbf{s_t} = \{s_{t,max}, s_{t,min}, s_{t,ave}\}$

Here, the word n-gram features allow the classifier to flexibly learn expressions that represent user satisfaction, and word classes allow these features to generalize. The sentiment lexicon features intuitively capture information such as "utterances including sentimentally charged words express positive or negative opinions about the previous utterance."

Based on these features, we construct the user satisfaction prediction model with Support Vector Regression (SVR) [9], which has previously seen success in dialogue quality estimation [16].

4. RESPONSE SELECTION

4.1. Single Response Baseline

Most EBDM methods select a single response r associated with query q from example database $\mathbf{e}$ with the highest similarity to user utterance q':

$$\langle \hat{q}, \hat{r} \rangle = \mathop{\mathrm{argmax}}_{\langle q,r \rangle \in \mathbf{e}} \mathrm{sim}(q', q). \tag{2}$$

In this paper, we use edit distance as similarity measure $\mathrm{sim}(q', q)$, as it is one of the most simple and effective algorithms to measure string similarity [19].

In our actual data, we have multiple responses $\mathbf{r}$ for each query, so we create two baselines to simulate how standard EBDM systems would act in this situation. The first, RANDOM, randomly chooses from the potential responses $\mathbf{r}$, simulating a situation where we don't consider quality of responses at all. The second baseline, MAXDB notes that in Section 2.2, we have 15 different writers who create example bases, and selects the single *writer* that achieves the highest satisfaction. This simulates a situation where we can collect a high quality single example base from a skilled writer.

4.2. Selection from Multiple Responses

In this section, we describe two proposed method for selection from multiples responses in EBDM. Both methods select the query q that has the highest similarity to user utterance q', and obtain its corresponding response set $\mathbf{r}$ from multi-response example database $\mathbf{e_{multi}}$:

$$\langle \hat{q}, \hat{\mathbf{r}} \rangle = \operatorname*{argmax}_{\langle q,\mathbf{r}\rangle \in \mathbf{e_{multi}}} \mathrm{sim}(q', q). \tag{3}$$

Next, we select a response r that has the highest expected satisfaction $\mathrm{C}(q, r)$ in response utterance candidates $\mathbf{r}$:

$$\langle \hat{q}, \hat{r} \rangle = \operatorname*{argmax}_{\langle q,r\rangle \in \langle \hat{q},\hat{\mathbf{r}}\rangle} \mathrm{C}(q, r). \tag{4}$$

We detail methods to calculate expected satisfaction $\mathrm{C}(q, r)$ in Sections 4.2.1 and 4.2.2.

4.2.1. Maximum Response using Average Satisfaction

Our first scoring method is entitled MAXR, for "maximum response," MAXR chooses the response that has the highest average evaluation score by human annotators (Section 2.2). This method is similar to MAXDB, but instead of having a single skilled writer create an example base, we have multiple writers create examples, and select the best example for each particular query.

Every pair of query q and response r has several scores annotated by different annotator, thus, we calculate the average satisfaction $\overline{s}_{\langle q,r\rangle}$ from the annotated satisfaction score $s_{u,\langle q,r\rangle}$ of each annotator $u \in U$:

$$\overline{s}_{\langle q,r\rangle} = \frac{1}{|U|} \sum_{u \in U} s_{u,\langle q,r\rangle}. \tag{5}$$

We then define $\mathrm{C}_{\mathrm{maxr}}(q, r) = \overline{s}_{\langle q,r\rangle}$ for the estimated satisfaction in Equation (4). While it considers multiple response candidate, the selected response is static. It is invariant throughout the dialogue, and not tailored to a specific user.

4.2.2. Adaptive Selection using Collaborative Filtering

The other method, named ADAPTIVE, is an adaptive method which uses the satisfaction prediction explained in Section 3, and collaborative filtering to adapt the response utterance to the user based on annotators who has similar preference.

Collaborative filtering is a technique widely used in recommendation systems to fill in estimates of user preference based on the preferences of other similar annotators. In spoken dialogue systems, collaborative filtering has been used to model user utterances or user satisfaction [13, 20]. However, these previous works use collaborative filtering only to evaluate the performance of the dialogue system or to predict user utterances. In contrast, we use collaborative filtering to estimate user preference to select the certain response for user.

We calculate expected satisfaction for the next system utterance based on predicted user satisfaction of the previous utterances. We do this by using collaborative filtering to compare the current user's predicted satisfaction with previous utterances with the tendencies of each annotator in the training data. Specifically, we estimate satisfaction data $\mathbf{s}_{est} = \left\{ s_{est,1}, \ldots, s_{est,|\mathbf{L_e}|} \right\}$ where each value represents the current user's satisfaction with a particular dialogue response in the list $\mathbf{L_e} = \left\{ \langle q_1, r_{1,1} \rangle, \langle q_1, r_{1,2} \rangle, \ldots \langle q_v, r_{v,w_v} \rangle \right\}$ which enumerates all the query-response pairs in example database $\mathbf{e}$. At first, these are filled by 0, which is the middle of the range of the normalized satisfaction score. Whenever a tri-turn passes, and the user makes a feedback utterance m, the system uses the satisfaction prediction model $\mathrm{R}(m)$ of Section 3 to predict the user's satisfaction to the system response. In the t-th tri-turn, user satisfaction data $\mathbf{s}_{est,t} = \left\{ s_{est,1}, \ldots, s_{est,|\mathbf{L_e}|} \right\}$ and user utterance q' are given, and the system selects as a response the n-th example in L_e, and finally the user replies a feedback utterance m_t. The system then estimates the user satisfaction for the example using the satisfaction prediction model $\mathrm{R}(m_t)$, and updates the n-the element of the user satisfaction data for the next $(t + 1)$-th tri-turn:

$$\begin{aligned} &\mathbf{s}_{est,(t+1)} \\ &= \left\{ s_{est,1}, \ldots, s_{est,n-1}, \mathrm{R}(m_t), s_{est,n+1}, \ldots, s_{est,|\mathbf{L_e}|} \right\} \end{aligned} \tag{6}$$

The value of s_{est} corresponding to this system response is then updated to be equal to this predicted value.

Once the s_{est} calculated, the system compares the current user's predicted satisfaction with each response $\mathbf{s}_{est}$ and annotated data $\mathbf{s}_u = \left\{ s_{u,1}, \ldots, s_{u,|\mathbf{L_e}|} \right\}$ for each annotator $u \in U$ who participated in the satisfaction annotation described in Section 2.2. Finally, the system estimates the satisfaction of each response by multiplying the cosine similarity between $\mathbf{s}_{est}$ and $\mathbf{s}_u$ with the annotator's satisfaction with the response $s_{u,\langle q,r\rangle}$ where $u \in U$, $r \in \mathbf{r}$ and the average satisfaction of all users is $\overline{s}_{\langle q,r\rangle}$:

$$\begin{aligned} \mathrm{C}_{\mathrm{adapt}}(q, r) = &\overline{s}_{\langle q,r\rangle} \\ &+ \sum_{u \in U} (s_{u,\langle q,r\rangle} - \overline{s}_{\langle q,r\rangle}) \mathrm{cos}(\mathbf{s}_{est}, \mathbf{s}_u). \end{aligned} \tag{7}$$

In this formula, we regard the cosine similarity between the two satisfaction vectors s_{est} and s_u as the reliability that the present user is similar to an annotator u in the training data.

5. EVALUATION

We evaluated the proposed method from two viewpoints: accuracy of satisfaction prediction, and effectiveness of response selection.

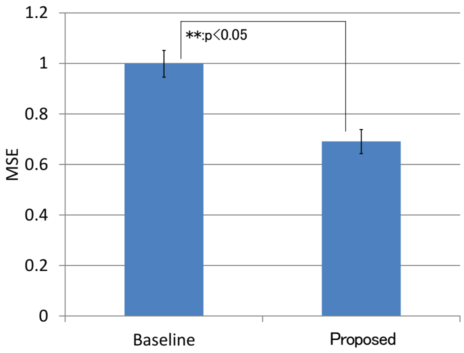

Fig. 1. Evaluation for satisfaction prediction

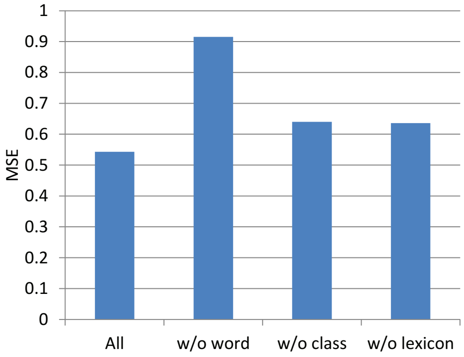

Fig. 2. Ablation tests for satisfaction prediction

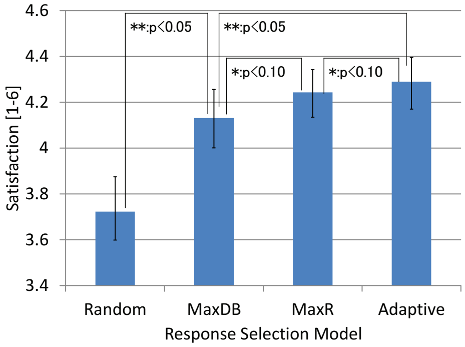

Fig. 3. Evaluation for response selection

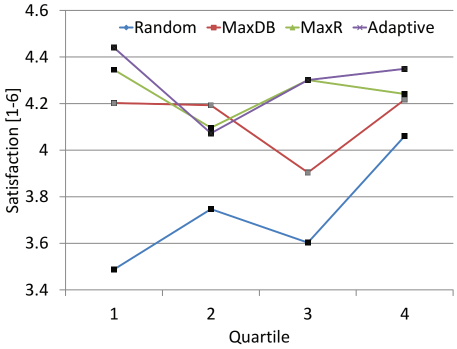

Fig. 4. Satisfactions by quartile of the dialogue

5.1. Evaluation for Predicting Satisfaction

In the evaluation for satisfaction prediction, we measured the Mean Squared Error (MSE) of predicted satisfaction for each tri-turn using 10-fold cross validation. We also show a baseline that always chooses the average satisfaction. We calculated the confidence interval of each evaluation measure using bootstrap resampling [21] with significance level $p < 0.05$. In Figure 1, we show the accuracy of satisfaction prediction. From this result, we can see that when we used the proposed prediction model, error decreased significantly to 0.69 compared to 1.00 of the baseline.

To analyze the effectiveness of features, we show ablation tests where we skip each variety of feature in Figure 2. From this result, we can see that the surface features of words are most effective. Features of word classes and the sentiment lexicon are not as important, but do provide some benefit.

5.2. Evaluation for Response Selection

In the response selection evaluation, we took 8 subjects who evaluate the responses provided by each response selection model. The subjects view replies selected by each response selection model for each input and assign satisfaction values for each reply. Thus, each subject gave a satisfaction score for 168 selected responses for 42 queries with 4 methods (RANDOM, MAXDB, MAXR, ADAPTIVE). The subjects also selected a response to which they want to reply and makes a feedback utterance for the selected response.

We compared the two baseline systems using random selection RANDOM, and the the database of the most proficient writer MAXDB, as described in Section 2.2, with the proposed static response seletion method MAXR as in Section 4.2.1, and the adaptive method ADAPTIVE as in Section 4.2.2.

Error bars are obtained with bootstrap resampling, and we perform a pairwise bootstrap to measure significance of differences between each model ($p < 0.05$). In Figure 3, we show the evaluation for response selection.

First, focusing on the difference between RANDOM and MAXDB, we can see that we obtain a significant improvement by going from an example database in which quality or consistency of the response is not considered to having an example database with the highest average satisfaction. This demonstrates the validity of our premise that not all responses are created equal, and it is necessary to consider the quality

Table 3. Examples of response selection by each model (translated from Japanese)

Turn	User	User Utterance	System Responses	Sat.	Model
6	A	I take a shower.	Certainly.	3	RANDOM
			When you take a shower, It seem to catch cold.	4	MAXDB
			Have a nice shower, please warm.	6	MAXR
			I prepare a change of clothes and a towel.	6	ADAPTIVE
37	B	Let me sleep a little more.	You said woke you up.	2	RANDOM
			It's fine today.	5	MAXDB
			I wake you up again at five minutes later.	6	MAXR, ADAPTIVE
29	C	Ah...	What did you happen?	4	RANDOM
			What did you say?	4	MAXDB
			Huh?	6	MAXR, ADAPTIVE

and the expected satisfaction of the response in EBDM systems.

Second, focusing on the difference between MAXDB and MAXR, we also obtain a slight improvement. This demonstrates the utility of considering multiple responses for each utterance.

Finally, focusing on the difference between MAXDB and ADAPTIVE, we can see a significant improvement with the highest average satisfaction to having adaptive response from all example databases. In addition, focusing on the difference between MAXR and ADAPTIVE, we can see a marginal significant improvement. These results indicate that performing adaptive response selection can increase in response quality.

In Table 3, we show an example of responses selected by each model. In the 6-th turn of user A, MAXR and ADAPTIVE got the best satisfaction score from the user. These two system responses cause the interaction more kindly in comparison with other two system responses, and it is thought the reason which make user satisfactory. Similarly, in the 37-th turn of user B, MAXR and ADAPTIVE selected the same system response which was kindly interaction, and got a highest evaluation for a user. On the other hand, like the 29-th turn of user C, we were often able to observe the situation where the user wished the system did not strongly perform an interaction. From these results, user satisfaction is considered enough as well as the appropriateness for the system responses.

Finally, in Figure 4, we show average satisfactions for 4 quartiles of the dialogue (each period is approximately 10 triturns). From this result, we can see that ADAPTIVE is the same as MAXR in the 2nd and 3rd quartile's satisfactions, but in the final quartile, ADAPTIVE improves satisfaction, possibly indicating that the model has adapted to the user somewhat by the end of the dialogue.

6. CONCLUSION

In this paper, we proposed methods constructing example-based dialogue system based on examples that pair one query and multiple responses, and adaptive response selection. In multi-response example database construction, we proposed the structure of example which has response candidates corresponding to a query. In response selection, we proposed two selection methods. The first is the STATIC method that considers the maximization of average satisfaction score, and the other is the ADAPTIVE method that uses collaborative filtering over explicit user feedback utterances. In an evaluation, we found that both proposed methods were effective, with adaptive response selection resulting in the highest satisfaction scores.

While the experimental results showed that the adaptive method is able to successfully select better response utterances, there are still a number of future challenges related to refining the example database and response selection model. The main potential for improvement lies in constructing response selection model acquired from larger training data. In collaborative filtering, the utility of performing collaborative filtering is largely influenced by whether a user similar to the current user can be found in the data. Therefore, it is important that there are a large number of diverse users in the database. Despite the fact that the database we used in this research was relatively small (5 annotators), we were still able to achieve an improvement in accuracy, but it is likely that larger databases could lead to further improvements in accuracy. In addition, we also plan to build the training data for response selection using un-annotated dialogue corpora. Specifically, when a dialogue is carried out by a new user, it may be beneficial to add the predicted satisfaction data as training data for collaborative filtering.

7. REFERENCES

[1] Cheongjae Lee, Sungjin Lee, Sangkeun Jung, Kyungduk Kim, Donghyeon Lee, and Gary Geunbae Lee, "Correlation-based query relaxation for example-based dialog modeling," in *Proc. ASRU*, 2009, pp. 474–478.

[2] Kyungduk Kim, Cheongjae Lee, Donghyeon Lee, Junhwi Choi, Sangkeun Jung, and Gary Geunbae Lee,

"Modeling confirmations for example-based dialog management," in *Proc. SLT*, 2010, pp. 324–329.

[3] Hiroya Murao, Nobuo Kawaguchi, Shigeki Matsubara, Yukiko Yamaguchi, and Yasuyoshi Inagaki, "Example-based spoken dialogue system using WOZ system log," in *Proc. SIGDIAL*, 2003, pp. 140–148.

[4] Rafael E. Banchs, "Movie-DiC: a movie dialogue corpus for research and development," in *Proc. ACL*, 2012, pp. 203–207.

[5] Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, Mirna Adriani, and Satoshi Nakamura, "Developing non-goal dialog system based on examples of drama television," in *Proc. IWSDS*, 2012, pp. 315–320.

[6] Fumihiro Bessho, Tatsuya Harada, and Yasuo Kuniyoshi, "Dialog system using real-time crowdsourcing and twitter large-scale corpus," in *Proc. SIGDIAL*, 2012, pp. 227–231.

[7] Rafael E. Banchs and Haizhou Li, "IRIS: a chat-oriented dialogue system based on the vector space model," in *Proc. ACL*, 2012, pp. 37–42.

[8] Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Improving the robustness of example-based dialog retrieval using recursive neural network paraphrase identification," in *Proc. SLT*, 2014, pp. 306–311.

[9] Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

[10] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. SIGIR*, 1999, pp. 230–237.

[11] Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella, "PARADISE: A framework for evaluating spoken dialogue agents," in *Proc. EACL*, 1997, pp. 271–280.

[12] Melita Hajdinjak and France Mihelič, "The PARADISE evaluation framework: Issues and findings," *Computational Linguistics*, vol. 32, no. 2, pp. 263–272, 2006.

[13] Zhaojun Yang, Baichuan Li, Yi Zhu, Irwin King, Gina-Anne Levow, and Helen M. Meng, "Collaborative filtering model for user satisfaction prediction in spoken dialog system evaluation.," in *Proc. SLT*, 2010, pp. 472–477.

[14] Sunao Hara, Norihide Kitaoka, and Kazuya Takeda, "Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system.," in *Proc. LREC*, 2010, pp. 78–83.

[15] Klaus-Peter Engelbrecht and Sebastian Möller, "A user model to predict user satisfaction with spoken dialog systems," in *Proc. IWSDS*, pp. 150–155. 2010.

[16] Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker, "Modeling and predicting quality in spoken human-computer interaction," in *Proc. SIGDIAL*, 2011, pp. 173–184.

[17] Francis Bond, Hitoshi Isahara, Sanae Fujita, Kiyotaka Uchimoto, Takayuki Kuribayashi, and Kyoko Kanzaki, "Enhancing the Japanese wordnet," in *Proc. ALR*, 2009, pp. 1–8.

[18] Hiroya Takamura, Takashi Inui, and Manabu Okumura, "Extracting semantic orientations of words using spin model," in *Proc. ACL*, 2005, pp. 133–140.

[19] Gonzalo Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, 2001.

[20] Ryuichiro Higashinaka, Noriaki Kawamae, Kohji Dohsaka, and Hideki Isozaki, "Using collaborative filtering to predict user utterances in dialogue," in *Proc. IWSDS*, 2009.

[21] Philipp Koehn, "Statistical significance tests for machine translation evaluation.," in *Proc. EMNLP*, 2004, pp. 388–395.