Automatic detection of task-incompleted dialog for spoken dialog system based on dialog act N-gram

Sunao Hara[1], *Norihide Kitaoka*[1], *Kazuya Takeda*[1]

[1]Graduate School of Information Science, Nagoya University, Japan

{naoh, kitaoka, kazuya.takeda}@nagoya-u.jp

Abstract

In this paper, we propose a method of detecting task-incompleted users for a spoken dialog system using an N-gram-based dialog history model. We collected a large amount of spoken dialog data accompanied by usability evaluation scores by users in real environments. The database was made by a field test in which naive users used a client-server music retrieval system with a spoken dialog interface on their own PCs. An N-gram model was trained from sequences that consist of user dialog acts and/or system dialog acts for two dialog classes, that is, the dialog completed the music retrieval task or the dialog incompleted the task. Then the system detects unknown dialogs that is not completed the task based on the N-gram likelihood. Experiments were conducted on large real data, and the results show that our proposed method achieved good classification performance. When the classifier correctly detected all of the task-incompleted dialogs, our proposed method achieved a false detection rate of 6%.

Index Terms: spoken dialog system, dialog act, dialog history, n-gram

1. Introduction

With the spread of such IP phone systems as Skype, the number of users who possess microphones has increased, and opportunities will continue to increase for PC users to exploit voice interactive systems with their own PCs and microphones. Such situations may have various environments, human errors, and so on, resulting in unexpected decreases not only of speech recognition accuracy but also task completion rate. To improve Spoken Dialog System (SDS) performance, speech data must be collected in the environment in which the system is used [1]. A number of studies have collected speech data in real environments, e.g., bus schedules and routing information by phone lines in Pittsburgh by Raux et al. [2] and Kyoto by Komatani et al. [3]. However, few studies [4] have addressed speech collection in real PC-based speech applications. Since our data were collected in such real PC environments, they have more various acoustic characteristics than data collected through phone lines.

The topic of the evaluation of system performance is as interesting as data collection. Speech recognition accuracy is the most important and commonly used measure of the performance of speech recognition systems [5]. On the other hand, user satisfaction and task completion rate are also crucial metrics for measuring the performance of such integrated systems as SDSs [6]. An important previous study on building such a performance measure was reported and related to the DARPA Communicator project [7, 8] to comparatively evaluate the different travel planning systems that participated. Walker et al. [9] proposed PARADISE as a general framework for characterizing user satisfaction with SDSs and used it for evaluations.

In general, the task completion rate is calculated based on manually labeled transcriptions of dialog data. Thus, it is difficult to detect users whose task completion rate is extremely low, e.g., those who could not use the SDS. And it is also difficult to compare with a current system and a functionally improved one if the current system is actually running. If a spoken dialog system can estimate its performance without manually labeled transcription, it can modify its dialog strategies and reduce the risk of problematic dialogs. A number of studies have focused on detecting problematic dialogs in Interactive Voice Responses (IVRs) installed in call centers. Walker et al. [10] proposed a problematic dialog predictor based on the *SLU-success* feature that encodes whether the spoken language understanding (SLU) component correctly captured the meaning of each exchange. They reported binary classification accuracy of 93% using the whole dialog and 86% accuracy even if only using the first two exchanges. Kim [11] focused on on-line prediction and proposed an N-gram-based call quality monitoring system and achieved problematic call detection accuracy of 83% after five turns. However, he only used user utterances in the modeling. Herm et al. [12] proposed a combined model of a system log with an emotion recognition result and reported 79% classification accuracy of problematic/non-problematic calls after only the first four turns [1].

The aim of this study is to construct a model to detect task-incompleted dialogs for spoken dialog systems based on real-world data. A task-incompleted dialog is defined as a dialog that failed to find five songs using our music retrieval system with the spoken dialog interface. Based on this definition, the system can easily determine when the dialog has completed its task; but our true aim is to identify dialogs whose users become so frustrated that they begin to hate the system. From user perspectives, they can only observe the system output (speech prompts or responses), not its internal states. Therefore, it is reasonable that the system outputs are heavily related to user impressions that directly affect task completion or incompletion. In this paper, we propose a method to detect task-incompleted dialogs for a spoken dialog system using an N-gram-based dialog history model. To consider the domain knowledge, a detection model is effective that consists of domain-specific concepts. To generalize and accurately make the model, utterances are encoded to the level of dialog acts. That is, the N-gram model is trained from user and/or system dialog act sequences for each dialog's class to determine whether they are task-completed.

The rest of this paper consists of four sections. In Section 2, we outline the field test and the data collection of the spoken di-

[1]Note that they also reported that the recognized emotional states have limited effectiveness for predicting problematic calls with their corpus. Kim also reported a similar result in [13].

	System's prompt / response and user's utterance	Act symbols
USR:	*Hello.*	USR_CMD_HELLO
SYS:	Hello.	SYS_INFO_GREETING
USR:	*"SIMON AND GARFUNKEL".*	USR_REQUEST_BYARTIST
SYS:	Do you want to retrieve songs by "Simon and Garfunkel"?	SYS_CONFIRM_KEYWORD
USR:	*Yes.*	USR_CMD_YES
SYS:	Now retrieving songs by "SIMON AND GARFUNKEL".	SYS_INFO_SEARCHBYARTIST
SYS:	60 songs were found.	SYS_INFO_SEARCHSUCCESS
SYS:	"I AM A ROCK".	SYS_INFO_SONGTITLE
SYS:	"BRIDGE OVER TROUBLED WATER".	SYS_INFO_SONGTITLE
USR:	*That one, please.*	USR_CMD_THATSONG
SYS:	Now playing "BRIDGE OVER TROUBLED WATER" by "SIMON AND GARFUNKEL." (The system plays the song.)	SYS_PLAY_SONG
USR:	*Stop.*	USR_CMD_STOP
SYS:	OK, the song is finished.	SYS_INFO_STOPPED
:	:	:

Figure 1: Example of dialog and its corresponding encoded symbols

alog corpus. In Section 3, we present formulations of the dialog data and their N-gram modeling. In Section 4, we build N-gram models for detecting task-incompleted dialogs from dialog act sequences and evaluate them. In Section 5, we summarize the paper.

2. Spoken dialog corpus collected in real user environments

Data collection was performed through field trials with the *Musicnavi2* music retrieval system [14, 15], with which users can look for and play music files on PCs through spoken dialogs. The client system can be downloaded and installed on PCs and works with a server program connected through the Internet. *Musicnavi2* uploads the input speech and the system behavior log so that the server can automatically collect a huge amount of speech data to make the database. The client's speech interface was implemented using a grammar-driven speech recognition interface with limited vocabulary and consists of player control words, song titles, artist names, and the album names of the music files stored on the user's PC. Julius 3.5.3 [16] was used as the speech recognition engine. An example of a dialog with the system is shown in Fig. 1.

Our system was used to collect speech data in a field test. Naive subjects were instructed to use the system until they had listened to at least five songs by performing at least twenty Q&A dialogs, or until they had listened to at least five songs using the system for over forty minutes. To avoid the result of randomly "playing a song", we defined one dialog as all utterances to complete "the task". Therefore, the number of users equals the number of dialogs.

These experimental data were maintained as a *Musicnavi2* database consisting of large-scale spoken dialogs with subjective usability evaluation results in real user environments [15]. 1,359 users participated in this experiment, and the sum of their usage time was about 488 hours. While raw recorded data contained a lot of unnecessary data, the data was automatically segmented by *Musicnavi2* using speech power level and zero-cross count, and we obtained about 29 hours of speech segments, corresponding to about sixty thousand utterances.

In this paper, we used 515 subjects from the database and classified their dialogs into two classes: completing the music retrieval task or failure to complete it, `COMPLETE` and `INCOMPLETE`, respectively. Class `COMPLETE` was composed of 449 subjects (dialogs), and class `INCOMPLETE` was composed of 66 subjects (dialogs).

Due to the nature of the task and the system architecture, most of the utterances were isolated-word utterances of an artist name, an album name, a song title, a short sentence including such proper names, or a short command sentence. On the other hand, the task vocabulary of *Musicnavi2* often contains uncommon phonetic contexts rarely seen in such general Japanese texts as newspaper articles because foreign words or even neologisms are used in music/song contexts.

3. N-gram model of dialog act sequence

During a dialog, the spoken dialog system sometimes makes bad or no responses to user utterances because of voice activity detection error, speech recognition error, dialog management error, or misunderstanding of the system by users. These unexpected responses create strange dialog contexts and decrease the dialog naturalness. Such unnaturalness, i.e., negative experiences, is accumulated during conversations with the system and thus users become frustrated and quit.

In this study, we used the N-gram model to model this architecture. Although word-level information is informative, a more generalized form such as a dialog act is better for accurate N-gram estimation. In this section, we define the dialog act sequences and model them by N-gram.

3.1. Encoding utterances to dialog acts

We encoded system utterances and their actions to 19 system act symbols and encoded user utterances and their actions to 19 user act symbols. In this study, we automatically used the collected features to define the users and the system dialog acts. Therefore, we used automatic speech recognition results instead of manual transcriptions, and thus user utterances were automatically encoded to user act symbols. Since the user act symbols were implemented in *Musicnavi2*'s recognition word vocabulary as non-terminal symbols in the grammar, they were easily mapped to dialog acts by combining user acts obtained from the speech recognition results. Also, the system act symbols were implemented as words in the system prompts or responses, and a dialog act consisted of a sequence of system acts.

Fig. 1 shows an example of a dialog and its corresponding encoded symbols.

3.2. Training the N-gram model

A dialog act sequence is created for every user by sequentially arranging both the system and user action symbols. Dialog act sequence $\mathbf{x}$ is denoted as follows:

$$\mathbf{x} = \{x_1, \ldots, x_t, \ldots, x_T\}, \quad (1)$$

where t is the dialog turn number.

Then we modeled dialog act sequence $\mathbf{x}$ using N-gram model $\mathcal{M}$:

$$\mathcal{M} = \{\mathcal{M}_c; c = 0, 1\}, \quad (2)$$

where models $\mathcal{M}_0$ and M_1 are trained using the dialogs of COMPLETE and INCOMPLETE users, respectively. The probability of dialog act sequence $\mathbf{x}$ when given dialog class c (0: COMPLETE or 1: INCOMPLETE), which is a likelihood, is approximated by N-gram probability as follows:

$$P(\mathbf{x}|\mathcal{M}_c) = \prod_{t=1}^{T} P(x_t|x_{t-1}, \ldots, x_{t-N-1}, \mathcal{M}_c). \quad (3)$$

The N-gram models were trained with the Witten-Bell discounting method using SRILM toolkit [17].

4. Detection of task-incomplete users

We use our proposed model to detect task-incomplete dialogs and evaluated its detection performance. A leave-one-out cross validation was performed using the data from 515 dialogs. The dialog act sequence of one dialog was used for testing, and the remaining dialog act sequences of 514 dialogs were used for training the model for each test.

We compared N-grams with $N = 1, 2, \ldots, 6$. Moreover, we compared the models trained by the dialog sequences in three conditions: using only the system dialog acts (SYS), using only the user dialog acts (USR), and using both the system and user dialog acts (SYS+USR).

To construct the classifier of the INCOMPLETE dialog, we introduced an a posteriori odds [18] classifier:

$$\hat{\mathcal{M}} = \begin{cases} \mathcal{M}_j & \text{if } \frac{P(\mathcal{M}_j|\mathbf{x})}{P(\mathcal{M}_i|\mathbf{x})} > 1, \\ \mathcal{M}_i & \text{otherwise}. \end{cases} \quad (4)$$

Applying Bayes' rule to Equation (4), we get:

$$\frac{P(\mathcal{M}_j|\mathbf{x})}{P(\mathcal{M}_i|\mathbf{x})} = \frac{P(\mathcal{M}_j)}{P(\mathcal{M}_i)} \frac{P(\mathbf{x}|\mathcal{M}_j)}{P(\mathbf{x}|\mathcal{M}_i)} = \frac{1}{e^{\alpha}} \frac{P(\mathbf{x}|\mathcal{M}_j)}{P(\mathbf{x}|\mathcal{M}_i)}, \quad (5)$$

where e^{α} is an inverse of the a priori odds. Finally, we applied the logarithm to Equation (5) and defined the classifier as follows:

$$\hat{\mathcal{M}} = \begin{cases} \mathcal{M}_j & \text{if } \ln P(\mathbf{x}|\mathcal{M}_j) - \ln P(\mathbf{x}|\mathcal{M}_i) > \alpha, \\ \mathcal{M}_i & \text{otherwise}. \end{cases} \quad (6)$$

We changed parameter α and evaluated the system performance by the maximum value of the classification accuracy and depicted a Receiver Operating Characteristic (ROC) curve.

The maximum value of classification accuracy, whether COMPLETE or INCOMPLETE, is shown in Table 1. The result indicated the highest accuracy of 98.8% by the 3-gram model in the SYS+USR condition. Fig. 2 shows the classification result of COMPLETE or INCOMPLETE. We obtained very good performance of the classification of task completion. In the USR condition, the performances by 1-, 2-, and 3-grams were drawn slightly lower than in the SYS condition. This suggests that user acts and their short contexts have more information than the system's context. On the other hand, the SYS performance by more than 4-grams outperformed USR. The longer the context become, the more information the system obtained. This limitation in the USR condition was caused by the inaccuracy of speech recognition. To exploit these models, the SYS+USR model shows promise. Actually, it achieved the highest performance. When the classifier correctly detected all of the task-incomplete dialogs, in other words, when the true detection rate was 100%, our proposed method achieved a false detection rate of only 6%. This confirms the effectiveness of using the context of both user and system acts.

Table 1: Maximum value of classification accuracy

	USR	SYS	SYS+USR
1-gram	90.3%	86.7%	89.8%
2-gram	94.2%	92.9%	96.5%
3-gram	**96.9%**	96.5%	**98.8%**
4-gram	96.3%	97.5%	98.7%
5-gram	95.6%	**98.7%**	98.5%
6-gram	95.6%	98.1%	98.1%

5. Conclusion

An N-gram model for detecting task-incomplete dialogs, which are defined as situations when users quit using a spoken dialog system, was studied based on the field trials of a voice-navigated music retrieval system. We proposed a detection method based on N-gram models of user and system dialog act sequences. Experimental results showed that the false detection rate achieved a good detection performance of 6%.

The proposed model's effectiveness was experimentally confirmed, but several future works remain. First, N-gram model-based prediction of dialog failure must be tested to clarify the dialog act contexts that affect user satisfaction through N-gram analysis and to research the relationships between word error rate and estimation performance. Some keywords are probably more crucial to estimate satisfaction; thus we will investigate a word-dialog act hybrid estimation method. Trouble with spoken dialog systems is not only caused by the users and the system but their usage acoustic environments; therefore, acoustic features will be helpful for detecting task-incomplete users at an early stage. An expected reduction of system operation cost is also an interesting standpoint to consider commercial usage, such as [19].

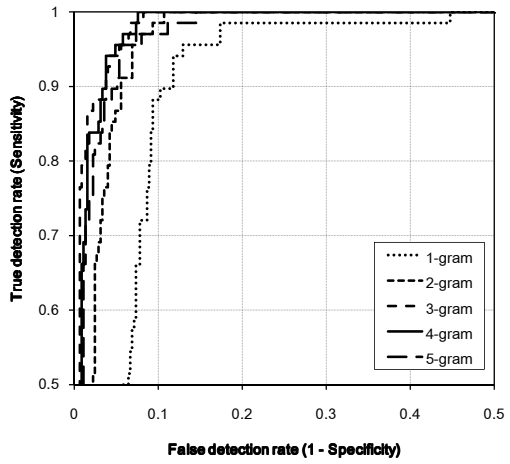
6. Acknowledgment

This work has been partially supported by Strategic Information and Communications R&D Promotion Programme (SCOPE) of Ministry of Internal Affairs and Communications, Japan.

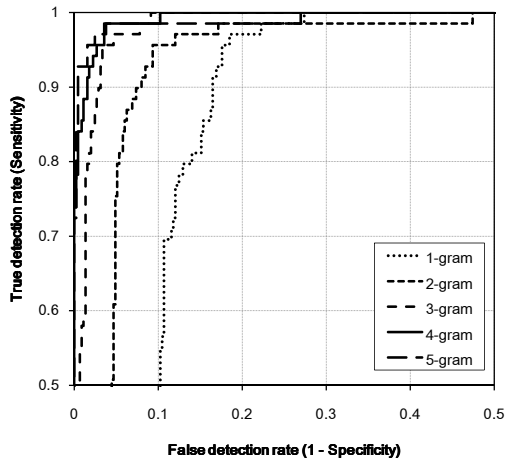
7. References

[1] J. Glass, J. Polifroni, S. Seneff, and V. Zue, "Data collection and performance evaluation of spoken dialogue systems: The MIT experience," in *Proceedings of ICSLP2000*, pp. 1–4, Oct. 2000.

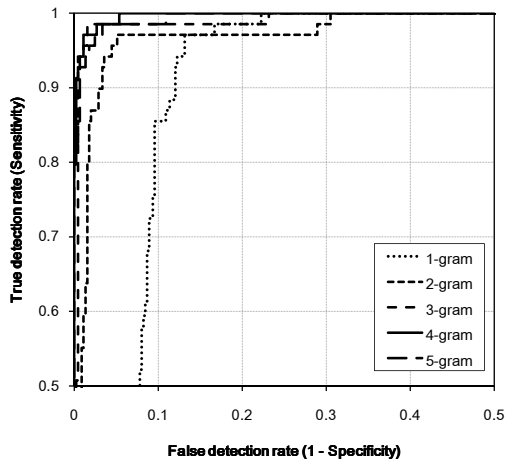
[2] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi,



(a) USR condition



(b) SYS condition



(c) SYS+USR condition

Figure 2: ROC curve for detection test of task-incomplete dialog ($c = 1$; INCOMPLETE)

"Let's Go Public! Taking a spoken dialog system to the real world," in *Proceedings of INTERSPEECH 2005*, pp. 885–888, Sep. 2005.

[3] K. Komatani, S. Ueno, T. Kawahara, and H. G. Okuno, "User modeling in spoken dialogue systems to generate flexible guidance," *User Modeling and User-Adapted Interaction*, vol. 15, no. 1, pp. 169–183, 2005.

[4] R. Nisimura, J. Miyake, H. Kawahara, and T. Irino, "Development of speech input method for interactive voiceweb systems," in *Proceedings of Human-Computer Interaction, Part II*, pp. 710–719. Springer, Jul. 2009.

[5] L. Dybkjar, N. O. Bernsen, and W. Minker, "Overview of evaluation and usability," in *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*. Springer, 2005, ch. 13, pp. 221–246.

[6] D. Gibbon, I. Mertins, and R. K. Moore, Eds., *Handbook of multimodal and spoken dialogue systems*. Boston: Kluwer Academic Publishers, 2000.

[7] M. Walker, J. Aberdeen, J. Bol, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, S. Seneff, D. Stallard, and S. Whittaker, "DARPA communicator dialog travel planning systems: The june 2000 data collection," in *Proceedings of Eurospeech 2001*, Sep. 2001.

[8] M. A. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, A. Potamianos, R. Passonneau, S. Roukos, G. S, S. Seneff, and D. Stallard, "DARPA communicator: Cross-system results for the 2001 evaluation," in *Proceedings of ICSLP2002*, pp. 269–272, Sep. 2002.

[9] M. Walker, D. Litman, C. Kamm, and A. Abella., "PARADISE: A framework for evaluating spoken dialogue agents," in *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, ACL 97*, pp. 271–280, Jul. 1997.

[10] M. A. Walker, I. Langkilde-Geary, H. W. Hastie, J. Wright, and A. Gorin, "Automatically training a problematic dialogue predictor for a spoken dialogue system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 293–319, May 2002.

[11] W. Kim, "Online call quality monitoring for automating agent-based call centers," in *Proceedings of INTERSPEECH 2007*, pp. 130–133, Aug. 2007.

[12] O. Herm, A. Shmitt, and J. Liscombe, "When calls go wrong: How to detect problematic calls based on log-files and emotions?" in *Proceedings of INTERSPEECH 2008*, pp. 463–466, Sep. 2008.

[13] W. Kim, "Using prosody for automatically monitoring human-computer call dialogues," in *Proceedings of Speech Prosody 2008*, pp. 79–82, May 2008.

[14] S. Hara, C. Miyajima, K. Itou, N. Kitaoka, and K. Takeda, "Data collection and usability study of a PC-based speech application in various user environments," in *Proceedings of Oriental COCOSDA 2008*, pp. 39–44, Nov. 2008.

[15] S. Hara, N. Kitaoka, and K. Takeda, "Estimation method of user satisfaction using N-gram-based dialog history model for spoken dialog system," in *Proceedings of LREC2010*, pp. 78–83, May 2010.

[16] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1691–1694, Sep. 2001.

[17] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP 2002*, pp. 901–904, Oct. 2002.

[18] E. T. Jaynes, "Model comparison," in *Probability Theory: The Logic of Science*, G. L. Bretthorst, Ed. Cambrige: Cambridge University Press, 2003, ch. 20, pp. 601–614.

[19] E. Levin and R. Pieraccini, "Value-based optimal decision for dialog systems," in *Proceedings of IEEE/ACL 2006 Workshop on Spoken Language Technologies (SLT 06)*, pp. 198–201, Dec. 2006.