



Detection of task-incomplete dialogs based on utterance-and-behavior tag N-gram for spoken dialog systems

Sunao Hara¹, Norihide Kitaoka¹, Kazuya Takeda¹

¹Graduate School of Information Science, Nagoya University, Japan

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

Abstract

We propose a method of detecting “task incomplete” dialogs in spoken dialog systems using N-gram-based dialog models. We used a database created during a field test in which inexperienced users used a client-server music retrieval system with a spoken dialog interface on their own PCs. In this study, the dialog for a music retrieval task consisted of a sequence of user and system tags that related their utterances and behaviors. The dialogs were manually classified into two classes: the dialog either completed the music retrieval task or it didn't. We then detected dialogs that did not complete the task, using N-gram probability models or a Support Vector Machine with N-gram feature vectors trained using manually classified dialogs. Off-line and on-line detection experiments were conducted on a large amount of real data, and the results show that our proposed method achieved good classification performance.

Index Terms: spoken dialog system, breakdowns in dialog, N-gram, task incomplete dialog detection

1. Introduction

Predicting performance is a central issue in designing spoken dialog systems (SDS). User satisfaction and task completion rates are crucial metrics for measuring the performance of such integrated systems [1]. Creating a standard measure to characterize the performance of spoken dialog systems remains critical and difficult. An important previous study on establishing such a measurement of performance was reported and related to the DARPA Communicator project [2, 3] to comparatively evaluate the participating travel planning systems. Walker et al. [4] proposed PARADISE as a general framework for characterizing user satisfaction with SDSs and used it for evaluations.

Generally, the task completion rate is calculated based on manually labeled transcriptions of dialog data. If a spoken dialog system can estimate its performance without manually labeled transcription, it can modify its dialog strategies itself 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. [5] proposed a problematic dialog predictor based on an *SLU-success* feature that encodes whether the spoken language understanding (SLU) component correctly captured the meaning of each exchange. They reported a binary classification accuracy rate of 93% using the whole dialog and 86% accuracy even if only the first two exchanges were used. Kim [6] focused on on-line prediction and proposed an N-gram-based call quality monitoring system, which achieved a problematic call detection accuracy rate of 83% after five turns. However, he only used user utterances in the modeling. Herm et al. [7] proposed a SLIPPER-based classifier for problematic dialog prediction, which creates a strong classifier comprised of a combination of weaker classifiers of features related to speech recognition, natural language understanding and dialog management components. They reported 79% classification accuracy of problematic/non-problematic calls after only the first

four turns. Schmitt et al. [8] proposed an N-gram modeling method for on-line prediction. They used N-grams of interaction parameters based on turn level, in other words, the input feature vector includes only the last *N* turns for predicting task completion in the current turn.

The aim of this study is to construct a model to detect “task incomplete” dialogs in spoken dialog systems using real-world data. A “task incomplete” dialog is defined as a dialog that failed to find the desired song using our music retrieval system which was equipped with a spoken dialog interface. Based on this definition, our system can easily determine when the dialog has completed its task; but our true aim is to immediately identify such failure dialogs in an on-line manner. It is assumed that detecting the failure dialogs through the assessment of dialog context is a useful approach for estimating the task completion rate and user satisfaction. Users can only observe the system's output (speech prompts or responses), not its internal states. Therefore, it is reasonable to assume that the system outputs strongly affect user impressions that directly affect task completion or incompleteness. In this paper, we apply a generative approach by using an N-gram probabilistic model, and a discriminative approach by using Support Vector Machine (SVM) [9]. To evaluate the knowledge contained in a domain, an effective detection model must consist of domain-specific concepts. To generalize and accurately make the model, utterances are encoded to the level of concept tags. That is, the N-gram model is trained using user and system tag sequences for each dialog's class to determine whether or not dialogs are “task complete”.

The rest of this paper consists of four sections. In Section 2, we outline the field test and data collection of the spoken dialog corpus. In Section 3, we present the formulations of the dialog data and their N-gram modeling. In Section 4, we build N-gram probabilistic models and SVM discriminators with N-gram feature vectors for detecting “task incomplete” dialogs from the tag sequences and evaluate them. In Section 5, we summarize the paper.

2. Spoken dialog corpus of a music retrieval task

We used the *Musicnavi2* database, which consists of large-scale spoken dialogs with subjective usability evaluation results in real user environments [10]. Inexperienced subjects used the system anywhere they liked (typically, in their homes) until they had listened to five or more songs that were associated with at least one of two conditions; either (a) a minimum of 20 question-answer dialogs, or (b) which lasted for a minimum of 40 minutes.

Users tried several different dialogs to listen to their desired songs with *MusicNavi2* during the experiments. If a song was played as the result of correct speech recognition on the dialog, we defined it as a “task complete” dialog (COMPLETE). The others were defined as “task incomplete” (INCOMPLETE). In this paper, we used 515 subjects from the database. Then we manually labeled the dialog borders and extracted 6,170 di-

System’s prompt / response and user’s utterances	Utterance and behavior tags	\mathbf{x}	\mathbf{p}	\mathbf{r}
USR: “SIMON AND GARFUNKEL”.	REQUEST-BYARTIST	x_1		r_1
SYS: Do you want to retrieve songs by “Simon and Garfunkel”?	CONFIRM-REQUESTED	x_2	p_1	
USR: <i>Yes.</i>	ANSWER-YES	x_3		r_2
SYS: Now retrieving songs by “SIMON AND GARFUNKEL.”	INFO-SEARCHBYARTIST	x_4	p_2	
SYS: 60 songs were found.	INFO-SEARCHSUCCESS	x_5	p_3	
SYS: “I AM A ROCK”.	SUGGEST-SONGTITLE	x_6	p_4	
SYS: “BRIDGE OVER TROUBLED WATER”.	SUGGEST-SONGTITLE	x_7	p_5	
USR: <i>That one, please.</i>	CMD-THESONG	x_8		r_3
SYS: Now playing “BRIDGE OVER TROUBLED WATER” by “SIMON AND GARFUNKEL.” (The system plays the song.)	PLAY-SONG	x_9	p_6	
USR: <i>Stop.</i>	CMD-STOP	x_{10}		r_4
SYS: OK, the song is finished.	REPLY-CMDSTOP	x_{11}	p_7	

Figure 1: Example of a dialog and its corresponding encoded tags

Table 1: Specifications of dialog data in the corpus

	INCOMPLETE	COMPLETE
# of dialogs	2803	3367
Avg. length of a dialog [sec.]	62.4	75.6
Avg. # of turns	23.6	26.3
Avg. # of user turns	10.9	11.8

dialogs which contained at least one exchange. Class COMPLETE was composed of 3,367 dialogs, and class INCOMPLETE was composed of 2,803 dialogs, as shown in Table 1.

Because of the “task complete” definition, i.e., “users could listen to their desired song,” we couldn’t detect whether the task was COMPLETE or INCOMPLETE by an occurrence of a “song was played” event. For example, if a user repeatedly searched and played songs for short intervals and eventually failed to find his or her song, we treated such a dialog as INCOMPLETE. Therefore, we must detect COMPLETE or INCOMPLETE tasks based not only on the “song was played” event, but also on the utterance sequence that led to the “song was played” event.

3. N-gram models of dialog tag sequences

An N-gram model is a model used to predict the next item in a sequence. There have been several studies on dialog understanding based on dialog act N-grams. Higashinaka et al. [11] estimated the dialog state in dialog understanding based on a trigram model trained from a manually labeled corpus. Hori et al. [12] constructed a Weighted Finite State Transducer (WFST) for a dialog scenario and estimated the next appropriate dialog act for the respondent’s utterance based on an integrated WFST. In comparison to these studies, our method detects the success or failure of SDS dialogs using automatically labeled data.

3.1. Encoding utterances and behaviors as tags

We encoded system utterances and behaviors as 19 system tags and encoded user utterances and behaviors as 19 user tags. We used automatically collected features to define the system and user tags and called them “utterance and behavior tags.” For the user tags, we used the automatic speech recognition results instead of manual transcriptions, and thus user utterances were automatically encoded to user tags. Since the user tags were directly associated with non-terminal symbols in the recognition grammar, they were easily mapped to the tags from the speech recognition results. As for the system tags, they were directly associated with the defined system prompts and responses.

We used not only utterances but also behaviors to help determine if a task was complete or incomplete. The system tags contain the events related to system behavior; e.g., playing a song (PLAY-SONG), ignoring the input because noise was detected (IGNORE-BYGMM), and ignoring the input because the trigger button wasn’t pushed (IGNORE-BYNOTRIGGER). The

user tags also contain such user behaviors as pushing the initialize button (PUSH-INITIALIZE) and detecting noises from the input interface (CATCHNOISE-). Figure 1 shows a dialog example and its corresponding encoded tags.

3.2. Training classifiers based on tag N-gram

A tag sequence is created for every dialog by sequentially arranging both the system and user tags according to the time indicated. System tag sequence \mathbf{p} , user tag sequence \mathbf{r} , and its integrated sequence \mathbf{x} are denoted as follows:

$$\mathbf{p} = \{p_1, \dots, p_s, \dots, p_S\}, \quad (1)$$

$$\mathbf{r} = \{r_1, \dots, r_t, \dots, r_T\}, \quad (2)$$

$$\begin{aligned} \mathbf{x} &= \{p_1, r_1, p_2, \dots, p_s, r_t, \dots, p_S, r_T\} \\ &= \{x_1, x_2, x_3, \dots, x_\ell, \dots, x_{S+T}\}, \end{aligned} \quad (3)$$

where S is the number of system turns and T is the number of user turns. Note that this definition allows two or more consecutive user or system tags.

For the purpose of on-line detection, we defined shrunk tag sequence $\mathbf{x}^{(t)}$ as follows:

$$\begin{aligned} \mathbf{x}^{(t)} &= \{p_1, r_1, p_2, \dots, p_s, r_t\} \\ &= \{x_1, x_2, x_3, \dots, x_{s+t}\}, \end{aligned} \quad (4)$$

where t is the number of user turns and \mathbf{x}_t is always terminated by the user tag.

Now, we consider the problem as a prediction of the dialog class c (-1: COMPLETE or +1: INCOMPLETE) using the tag sequences $\mathbf{x}^{(t)}$. The N-gram probabilistic models were trained from sets of tag sequences. The SVM functions were trained from the sets of features consisting of the frequencies of tag N-grams¹.

We modeled tag sequence \mathbf{x} using N-gram probabilistic model \mathcal{M} :

$$\mathcal{M} = \{\mathcal{M}_c; c = -1, +1\}, \quad (5)$$

where models \mathcal{M}_{-1} and \mathcal{M}_{+1} are trained using dialogs labeled COMPLETE and INCOMPLETE. The probability of shrunk tag sequence $\mathbf{x}^{(t)}$ when given dialog class c , which is a likelihood, is approximated by N-gram probability as follows:

$$P(\mathbf{x}^{(t)} | \mathcal{M}_c) \simeq \prod_{\ell=1}^t P(x_\ell | x_{\ell-1}, \dots, x_{\ell-(N-1)}, \mathcal{M}_c). \quad (6)$$

To construct the classifier of the INCOMPLETE dialog, we introduced a log-likelihood ratio (LLR) classifier:

$$\hat{c}^{(t)} = \begin{cases} +1 & \text{if } \ln P(\mathbf{x}^{(t)} | \mathcal{M}_{+1}) - \ln P(\mathbf{x}^{(t)} | \mathcal{M}_{-1}) > \alpha_t, \\ -1 & \text{otherwise,} \end{cases} \quad (7)$$

¹The feature of word 1-gram is known as “Bag-of-Words” feature.

where α_t is a threshold parameter for the number of user turns t . Discriminative functions of SVM were trained for each number of user turns using $\mathbf{x}^{(t)}$. The classifiers were defined as follows:

$$\hat{c}^{(t)} = \begin{cases} +1 & \text{if } \Phi_t(B(\mathbf{x}^{(t)})) > \alpha_t, \\ -1 & \text{otherwise,} \end{cases} \quad (8)$$

where $B(\cdot)$ is the function constructing feature vector, and α_t and $\Phi_t(\cdot)$ are a threshold parameter and discriminative functions of SVM for the t , respectively.

4. Detection of task-incomplete dialogs

We used our proposed models to detect “task incomplete” dialogs and evaluated its detection performance. A five-fold cross validation (open condition for users) was performed using the data from 515 users. In our corpus, all of the “task complete” dialogs contained the PLAY-SONG tag. PLAY-SONG tags which occurred at the end of a dialog are heavily related to task completion. Therefore, the last PLAY-SONG tag and the tags following it were truncated from the sequence.

The N-gram probabilistic models were trained with the Witten-Bell discounting method using SRILM toolkit [13]. The SVM discriminators were trained using LIBSVM[14]. As kernel functions for SVM, linear function (SVM-Linear) and radial basis function (SVM-RBF) were used, and its hyper-parameters were selected by grid-search and 5-fold cross validation of each training set. For comparison purposes, C4.5 decision trees were also trained with the same features for SVM using J4.8 algorithm of WEKA[15].

We compared N-grams with $N = 1, 2, \dots, 5$, and denoted them as 1-gram, 1-2gram, 1-3gram, 1-4gram and 1-5gram, respectively, e.g., 1-3gram represents the features constructed from the frequencies of 1-gram, 2-gram and 3-gram. The unique numbers of N-grams were 38, 1036, 7798, 32441 and 92342, respectively. These numbers corresponded to the feature vector lengths.

To construct the detector of INCOMPLETE dialogs, we introduced Equations 7 and 8. We changed parameter α_t and evaluated the system performance using the maximum value of the classification accuracy and depicted a Receiver Operating Characteristic (ROC) curve.

4.1. Evaluation of off-line detection

An off-line detection experiment was carried out using the whole tag sequence \mathbf{x} instead of $\mathbf{x}^{(t)}$ of Equations 9 and 10.

The maximum value of the classification accuracy, whether COMPLETE or INCOMPLETE, is shown in Figure 2. The result of the SVM-Linear classifier indicated a highest accuracy of 87.7% with the 1-4gram feature vector. The experimental results show high detection performance, even if the last PLAY-SONG tag, i.e., our task dependent information, did not exist. This suggests that our method might be able to detect failure dialogs before a dialog is finished.

Figure 3(a) shows the ROC curves for the detection test of INCOMPLETE dialogs using the N-gram LLR classifier with 1-gram, 2-gram and 3-gram. The highest performance was achieved using the 2-gram model. On the other hand, performance decreased using the 3-gram model. In fact, the same decreases occurred with the 4-gram and 5-gram models. Therefore, these decreases might be due to overfitting. The results of Figure 2 and Figure 3(a) suggested that 2-gram or 3-gram models are sufficient for N-gram LLR classifiers.

Figure 3(b) shows the ROC curves for detection test of INCOMPLETE dialogs using SVM. This result shows higher performance than the result using an N-gram LLR classifier. The 1-2gram feature achieved higher performance than the 1-gram feature, however, there were few differences between

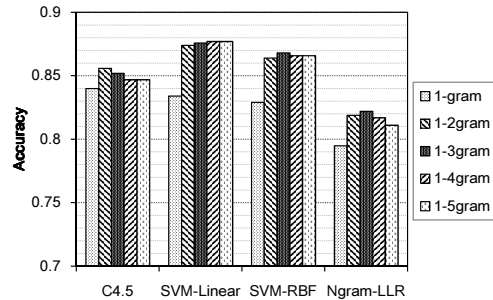


Figure 2: Prediction accuracy of task completion or incompleteness using dialogs. Chance rate is 0.5 and majority vote baseline is 0.55.

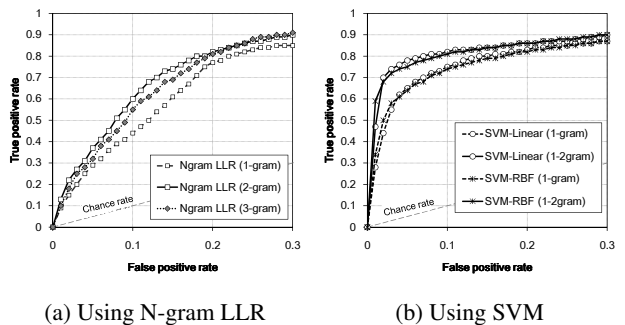


Figure 3: ROC curves for detection test of task-incomplete dialogs

SVM-Linear and SVM-RBF. This might be caused by the sufficiently high dimension of the feature vector. The results also suggested that the 1-2gram feature was sufficient for the SVM with linear kernel, and that, more N of N-gram modeling does not cause the performance to decrease, as was the case with the N-gram LLR classifier results.

4.2. Evaluation of on-line detection performance

We also attempted on-line detection, e.g., detecting “task incomplete” dialogs before they ended. Detection and evaluation were done for each number of user turns t . For ease of discussion, we will focus on the highest performance features/models in Figure 2 for each method; i.e. 1-2gram of C4.5 decision tree, 1-4gram of SVM with linear kernel, 1-3gram of SVM with RBF kernel and 1-3gram of N-gram LLR method.

Figure 4 shows the results of accuracy as a function of the number of user turns. The results for SVMs with linear and RBF kernels show they outperformed the C4.5 decision tree. The lines representing SVM and N-gram LLR crossed when the number of user turns equaled two, however, the performance of SVM was superior when the number of user turns was greater than two.

Figure 5 shows the results of the true positive rate as a function of the number of user turns, where the false positive rate was fixed to 10%. For all methods, the results of 1-2gram model (solid lines) outperformed the result of 1-gram models (dashed lines). The result using the SVM outperformed the result of the N-gram LLR method when the number of user turns was greater than 4, however, tendencies are inverted when the number of user turns is less than or equal to 4. This might be caused by differences in the ways the models were trained, that is, the SVMs were trained as turn specific models, while the N-gram probabilistic models were trained as overall dialog context models. With our feature vector definition, the training vectors tend

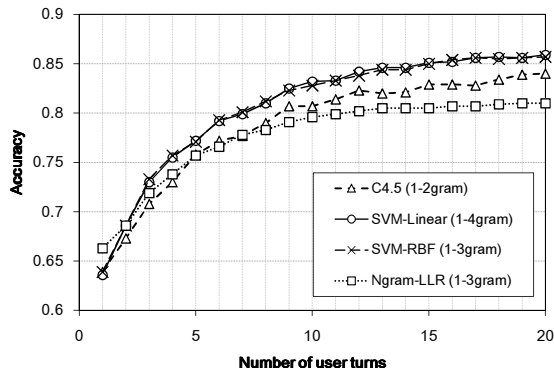


Figure 4: Accuracy as a function of a number of user turns. Chance rate is 0.5 and majority vote baseline is 0.55.

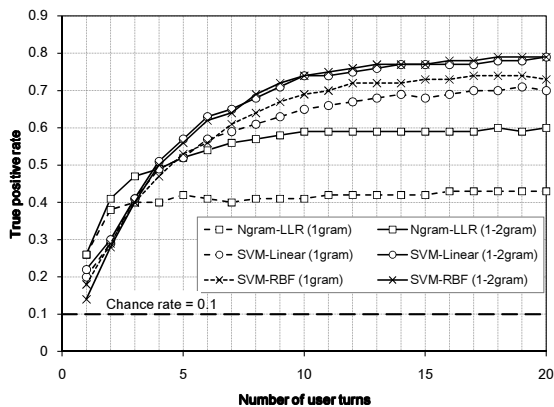


Figure 5: True positive rate as a function of number of user turns, where the false positive rate was fixed at 10%.

to be sparse when the turn numbers are small, as compared with larger turn numbers, therefore, the SVM models show lower performance with low turn numbers. These results also suggested that the prediction performance might be improved by the integration of the N-gram LLR method and the SVM.

5. Conclusion

An N-gram based method for detecting “task incomplete” dialogs, which are defined as situations when users couldn’t listen to their desired songs, was studied using the *MusicNavi2* database, which was collected during field trials of a music retrieval system with a spoken dialog interface. We proposed a detection method based on the N-grams of user and system tag sequences. We evaluated the dialogs using either N-gram probabilistic models or N-gram feature vectors, and carried out a dialog classification prediction experiment.

The proposed model’s effectiveness was experimentally confirmed, but several future studies are needed. First, since the occurrence of some N-gram representations are probably more crucial to system performance, we will investigate which N-gram representations most affected the task completion detection rate. The impact of speech recognition error on the estimation performance must be clarified. Estimating user satisfaction from N-gram likelihood or its ratio is also an interesting topic. Difficulty with spoken dialog systems is not only caused by the users and the system but also by their acoustic environments; therefore, acoustic features may be helpful for detecting “task incomplete” dialogs at an early stage. An expected reduction in system operation cost is also an interesting topic to

consider from the standpoint of commercial usage as in [16]. It might also be important to use “interaction parameters” [17] and/or extended interaction parameters similar to those used in [8, 18].

6. Acknowledgment

This work has been supported in part by the NEDO Grant for Industrial Technology Research Program.

7. References

- [1] D. Gibbon, I. Mertins, and R. K. Moore, Eds., *Handbook of multimodal and spoken dialogue systems*. Kluwer Academic Publishers, 2000.
- [2] M. Walker, J. Aberdeen, J. Bol, E. Bratt, J. Garofolo, and et al., “DARPA communicator dialog travel planning systems: The June 2000 data collection,” in *Proc. of Eurospeech 2001*, Sep. 2001.
- [3] M. A. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, A. Potamianos, and et al., “DARPA communicator: Cross-system results for the 2001 evaluation,” in *Proc. of ICSLP 2002*, pp. 269–272, 2002.
- [4] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, “PARADISE: A framework for evaluating spoken dialogue agents,” in *Proc. of ACL 97*, pp. 271–280, Jul. 1997.
- [5] 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.
- [6] W. Kim, “Online call quality monitoring for automating agent-based call centers,” in *Proc. of INTERSPEECH 2007*, pp. 130–133, Aug. 2007.
- [7] O. Herm, A. Schmitt, and J. Liscombe, “When calls go wrong: How to detect problematic calls based on log-files and emotions?” in *Proc. of INTERSPEECH 2008*, pp. 463–466, Sep. 2008.
- [8] A. Schmitt, M. Scholz, W. Minker, J. Liscombe, and D. Sündermann, “Is it possible to predict task completion in automated troubleshooters?” in *Proc. of INTERSPEECH 2010*, pp. 94–97, Sep. 2010.
- [9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
- [10] S. Hara, N. Kitaoka, and K. Takeda, “Estimation method of user satisfaction using N-gram-based dialog history model for spoken dialog system,” in *Proc. of LREC 2010*, pp. 78–83, May 2010.
- [11] R. Higashinaka and M. Nakano, “Ranking multiple dialogue states by corpus statistics to improve discourse understanding in spoken dialogue systems,” *IEICE Trans. Information and Systems*, vol. E92-D, no. 9, pp. 1771–1782, Sep. 2009.
- [12] C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura, “Statistical dialog management applied to WFST-based dialog systems,” in *Proc. of ICASSP 2009*, pp. 4793–4796, Apr. 2009.
- [13] A. Stolcke, “SRILM – an extensible language modeling toolkit,” in *Proc. of ICSLP 2002*, pp. 901–904, Oct. 2002.
- [14] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [15] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann publishers, 2005.
- [16] E. Levin and R. Pieraccini, “Value-based optimal decision for dialog systems,” in *Proc. of IEEE/ACL 2006 Workshop on Spoken Language Technology*, pp. 198–201, Dec. 2006.
- [17] S. Möller, “Parameters for quantifying the interaction,” in *Proc. of SIGdial 2005*, pp. 166–177, Sep. 2005.
- [18] K.-P. Engelbrecht and S. Möller, “Sequential classifiers for the prediction of user judgments about spoken dialog systems,” *Speech Communication*, vol. 52, no. 10, pp. 816–833, Oct. 2010.