Causal analysis of task completion errors in spoken music retrieval interactions

† Sunao Hara, ‡ Norihide Kitaoka, ‡ Kazuya Takeda

† Graduate school of information science, Nara Institute of Science and Technology,
8916-5, Takayama-cho, Ikoma, Nara, Japan
‡ Graduate school of information science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Aichi, Japan
† hara@is.naist.jp, ‡ {kitaoka, kazuya.takeda}@nagoya-u.jp

Abstract

In this paper, we analyze the causes of task completion errors in spoken dialog systems, using a decision tree with N-gram features of the dialog to detect task-incomplete dialogs. The dialog for a music retrieval task is described by a sequence of tags related to user and system utterances and behaviors. The dialogs are manually classified into two classes: completed and uncompleted music retrieval tasks. Differences in tag classification performance between the two classes are discussed. We then construct decision trees which can detect if a dialog finished with the task completed or not, using information gain criterion. Decision trees using N-grams of manual tags and automatic tags achieved 74.2% and 80.4% classification accuracy, respectively, while the tree using interaction parameters achieved an accuracy rate of 65.7%. We also discuss more details of the causality of task incompletion for spoken dialog systems using such trees.

Keywords: Spoken dialog, Task incompletion, Interaction parameters

1. Introduction

Estimating performance is a key issue in the design of spoken dialog systems, and the task completion rate is commonly used as a performance metric. Generally, the task completion rate is calculated based on manually labeled transcriptions of dialog data. However, automatic performance assessment (e.g. assessment based on objective features) would generally be preferable from a cost standpoint. In addition, if a spoken dialog system can estimate its own performance on-line and in real time, the system can change its dialog strategies, reducing the risk of problematic dialogs.

There have been a number of studies focused on detecting problematic dialogs in Interactive Voice Responses (IVRs) installed in call centers, e.g. (Walker et al., 2002; Kim, 2007; Herm et al., 2008; Engelbrecht and Möller, 2010). We have also proposed methods for predicting task completion errors during music retrieval tasks (Hara et al., 2010; Hara et al., 2011a; Hara et al., 2011b). These studies were aimed at the detection of problematic dialogs, however the causality of these problematic dialogs has still not been investigated sufficiently.

We based our study on the construction a model to estimate user satisfaction while using spoken dialog systems. From the users' point of view, only the system's output, such as speech prompts or responses, can be observed, not the system's internal states. Therefore, a system's outputs are assumed to strongly influence a user's impression of the system, which directly affects user satisfaction. Previously, we proposed an estimation method of user satisfaction for a spoken dialog system using an N-gram-based dialog history model (Hara et al., 2011a). The N-gram model was trained with user and system utterance sequences, which were encoded according to the users' and the system's dialog utterances for each user satisfaction level. In this paper, we investigate why users could not finished their music retrieval dialog tasks. For this purpose, we analyzed corpus specifications, focusing on tag classification performance, which is the first step toward understanding overall system performance, and heavily related to our proposed method for detecting incomplete dialog problems. We then constructed decision trees for each of two features using our method; N-gram features and interaction parameter features. The trees were assumed to clarify the rank of importance of the component containing each feature.

The rest of this paper consists of four sections. In Section 2, we outline the field test used to collect spoken dialog data, and give some brief statistics regarding the data collection process. In Section 3, we present the specifications of the dialog data in terms of interaction parameters and N-gram features. In Section 4, we construct the decision trees and discuss their structures and components. In Section 5, we summarize the findings of our paper.

2. Corpus of Spoken dialog from music retrieval tasks

2.1. MusicNavi2: an interactive, spoken dialog music search system

Data collection was performed through field trials of the MusicNavi2 music retrieval system (Hara et al., 2010), with which users can search and play music stored 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. The MusicNavi2 client uploads the input speech and the system behavior log so that the server can automatically collect a huge amount of speech data for the database. The client's speech interface used a grammar-driven speech recognition interface, whose the vocabulary consisting of music player control words, song titles, artist names, and the album names of the music files stored on the user's PC. Julius (Kawahara et al., 2004) was used as the speech recognition engine. An example of a dialog using the system is shown in Figure 1.

The system was used to collect speech data in a field test. Untrained 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.

These experimental data was preserved as a large-scale MusicNavi2 database consisting of spoken dialogs in real user environments, and included subjective usability evaluation results collected from users (Hara et al., 2008; Hara et al., 2010). A total of 1,359 users participated in this experiment, and the total usage time was about 488 hours. The raw recorded data contained many sections that were unnecessary for retrieval dialogs, so the data was automatically segmented by MusicNavi2 using speech value and zero-cross count. We then obtained about 29 hours of speech segments, corresponding to about sixty thousand utterances. Preliminary analysis of the database, e.g. recognition accuracy, SNR, speaking rate, etc. were presented in another paper (Hara et al., 2008). These analyses were focused on utterances of users. In this paper, we focused on the intermediate component of speech recognition systems, that is, the interactive dialogs. A total of 515 subjects were selected for analysis from the database. Each of them engaged in several dialogs when they used the system. We segmented and labeled each dialog from the database, and classified them into two classes: those in which the dialog resulted in completion of the music retrieval task (`COMPLETE`) and those which failed to complete it (`INCOMPLETE`). Class `COMPLETE` was composed of 3,720 dialogs, and class `INCOMPLETE` was composed of 3,828 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. It should be noted that the task vocabulary for MusicNavi2 often contains unusual phonetic contexts rarely seen in typical Japanese texts as newspaper articles because foreign words or even neologisms are used in music/song contexts.

2.2. Encoding utterances and behaviors as tags

We encoded system utterances and their actions into 19 system-related tags, and encoded user utterances and their actions into 19 user-related tags. We called them "utterance-and-behavior tags". The system-related tags were derived from words in the system prompts or responses. User related tags, on the other hand, were obtained using two methods; automatic tags and manual tags. Since the user-related tags appeared in MusicNavi2's recognition vocabulary as non-terminal symbols in the grammar, they were easily mapped to the tags from the speech recognition results. We called these tags automatic user-related tags. These tags clearly displayed the results of speech recognition errors. The manual user-related tags were obtained from manual transcription. These manual transcription-based tags were likely to be very helpful for tendency analysis in determining what kind of errors tended to severely affect task completion/incompletion.

We used not only utterances but also behaviors to help determine if a task was completed or not. The system tags contain events related to system behavior: playing a song (PLAY-SONG), ignoring the input because the input was noise (IGNORE-BYGMM), and ignoring the input because the trigger button was not pushed (IGNORE-BYNOTRIGGER). The user tags also contain user behaviors such as pushing the initialize button (PUSH-INITIALIZE) and input being classified as noise by the input interface (CATCHNOISE-TOOSHORT/-GMM/-ASR). Figure 1 shows an example of a dialog and its corresponding encoded tags. Tables 1 and 2 show the definitions of the 19 user tags and 19 system tags (behavior tags are noted in the tables with "*"). These tables also show the relative frequencies with which the tags appeared in the database. These tables are discussed in more detail in the next section.

3. Specification of corpora

3.1. Performance of tag classification task

The user tags were automatically encoded by MusicNavi2 as mentioned above. Its automatic encoding performance, in other words, its tag classification performance, was assumed to be related to the performance of the comple-

Table 1: Relative frequency of user's utterance-and-behavior tags in the corpus.

Definition of User tags	**Relative frequency of manual tags**			**Relative frequency of automatic tags**		
	Total	COMPLETE	INCOMPLETE	Total	COMPLETE	INCOMPLETE
REQUEST-BYARTIST	9.5 %	9.8 %	14.2 %	5.8 %	8.1 %	6.6 %
REQUEST-BYMUSIC	14.7 %	21.9 %	15.5 %	11.8 %	19.2 %	10.8 %
ANSWER-YES	7.3 %	10.8 %	7.8 %	5.9 %	8.6 %	6.4 %
ANSWER-NO	2.1 %	2.1 %	3.2 %	5.2 %	4.4 %	8.6 %
CMD-THESONG	1.9 %	3.4 %	1.5 %	3.2 %	4.0 %	4.2 %
CMD-NEXTSONG	1.7 %	3.1 %	1.2 %	2.8 %	4.0 %	3.1 %
CMD-PREVIOUSSONG	0.8 %	1.4 %	0.6 %	1.5 %	1.8 %	1.9 %
CMD-HELLO	1.4 %	0.8 %	2.7 %	1.3 %	1.0 %	2.3 %
CMD-MICTEST	0.8 %	0.3 %	1.8 %	1.0 %	0.6 %	2.0 %
CMD-RESPEECH	0.7 %	1.0 %	0.7 %	0.9 %	1.1 %	1.2 %
CMD-BACKHISTORY	0.2 %	0.2 %	0.2 %	0.8 %	0.6 %	1.4 %
CMD-STOP	6.2 %	8.3 %	7.3 %	5.3 %	7.1 %	6.3 %
CMD-INITIALIZE	1.2 %	0.7 %	2.4 %	1.9 %	1.1 %	3.6 %
CMD-EXIT	1.5 %	0.6 %	3.3 %	1.7 %	0.9 %	3.3 %
PUSH-INITIALIZEBUTTON	1.2 %	0.7 %	2.4 %	1.9 %	1.1 %	3.6 %
CATCHNOISE-TOOSHORT	0.3 %	0.3 %	0.3 %	0.4 %	0.4 %	0.4 %
CATCHNOISE-GMM	39.0 %	27.8 %	67.4 %	43.9 %	31.7 %	76.1 %
CATCHNOISE-ASR	0.0 %	0.0 %	0.0 %	2.4 %	2.2 %	3.8 %
UNDEFINED	10.2 %	7.4 %	18.4 %	3.6 %	2.8 %	6.3 %

Table 2: Relative frequency of system's utterance-and-behavior tags in the corpus.

Definition of System tags	**Relative frequency of automatic tags**		
	Total	COMPLETE	INCOMPLETE
PROMPT-ANYSONG	1.4 %	1.0 %	2.2 %
PROMPT-RESPEAK	1.4 %	1.4 %	2.0 %
SUGGEST-SONGTITLE	12.0 %	17.3 %	10.4 %
PLAY-SONG	9.3 %	17.1 %	4.4 %
CONFIRM-EXIT	0.9 %	0.5 %	1.6 %
CONFIRM-REQUESTED	10.0 %	12.0 %	9.9 %
INFO-SEARCHFAIL	0.5 %	0.2 %	0.9 %
INFO-SEARCHSUCCESS	2.2 %	3.4 %	1.8 %
INFO-SEARCHBYARTIST	1.8 %	2.8 %	1.4 %
REPLY-CMDHELLO	0.6 %	0.5 %	0.9 %
REPLY-CMDMICTEST	0.3 %	0.2 %	0.5 %
REPLY-CMDRESPEECH	1.0 %	0.8 %	1.4 %
REPLY-CMDBACKHISTORY	0.4 %	0.3 %	0.5 %
REPLY-CMDSTOP	3.9 %	5.0 %	4.1 %
REPLY-CMDINITIALIZE	1.5 %	1.3 %	2.2 %
INFO-GOODBYE	1.4 %	0.4 %	2.8 %
IGNORE-BYGMM*	12.6 %	10.0 %	18.0 %
IGNORE-BYNOTRIGGER*	38.7 %	25.8 %	62.3 %
UNDEFINED	0.0 %	0.0 %	0.1 %

	System's prompt / response and user's utterance	Utterance-and-behavior tags
USR:	*"SIMON AND GARFUNKEL"*.	REQUEST-BYARTIST
SYS:	Do you want to retrieve songs by "Simon and Garfunkel"?	CONFIRM-KEYWORD
USR:	*Yes.*	ANSWER-YES
SYS:	Now, retrieving the songs by "SIMON AND GARFUNKEL".	INFO-SEARCHBYARTIST
SYS:	60 songs were found.	INFO-SEARCHSUCCESS
SYS:	"I AM A ROCK".	SUGGEST-SONGTITLE
SYS:	"BRIDGE OVER TROUBLED WATER".	SUGGEST-SONGTITLE
USR:	*That one, please.*	CMD-THESONG
SYS:	Now, playing the song "BRIDGE OVER TROUBLED WATER" by "SIMON AND GARFUNKEL." (The system plays the song.)	PLAY-SONG
USR:	*Stop.*	CMD-STOP
SYS:	OK, the song is stopped. (The system stops the song.)	REPLY-CMDSTOP

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

tion/incompletion dialog classification task. Tables 3 and 4 show the classification performance regarding "recall", whose denominator is the number of manual tags, and "precision", whose denominator is the number of automatic (estimated) tags.These tables also show the relative frequencies of major and minor errors along with the error rates for each manual and automatic tag.

As shown in the Table 3, the recalls of "REQUEST-BYARTIST" and "REQUEST-BYMUSIC" were low because of unknown words. We could also find many major errors due to estimation as CATCHNOISE-GMM. This means that MusicNavi2 was used in acoustically noisy environments. According to MusicNavi2's dialog management strategy, ANSWER-YES and CMD-THESONG are the most important keywords, however there were a high number of errors that were unrecognized as CATCHNOISE-GMM. This error might be one of the reason why users could not finished their music retrieval tasks.

The Table 4 shows that CATCHNOISE-TOOSHORT, CATCHNOISE-GMM, and CATCHNOISE-ASR achieved high precision rates, while UNDEFINED was misrecognized, making it of no use to the system. Attention should be also paid to the low precision rate of ANSWER-NO. This might be the cause of difficulty in recovery from misrecognition errors using the user's negative representation, and it might also be the cause of user dissatisfaction. These results also suggest that the noise rejection function should be performed more carefully.

3.2. N-gram feature of tag sequence

The N-gram, or N-order Markov chain, is a simple assumption for modeling sequential data, but is also a very useful modeling method (Kim, 2007; Engelbrecht and Möller, 2010; Hara et al., 2011a). We compared N-grams with N=1, 2, 3 and denoted them as 1-, 1-2 and 1-3 grams, respectively; e.g., a 1-3 gram represents the features constructed from the frequencies of 1-gram, 2-gram, and 3-gram [1]. The number of unique N-grams in the entire corpus were 38, 1036, 7798, respectively. For this experiment, we calculate the number of unique N-gram tags limited by two or more occurrences, which we called N-gram tag cutoffs, for both the manual tag and the automatic tag corpora. The manual tag corpus contained 38, 780, and 4144 unique cutoff N-grams, and the automatic tag corpus contained 38, 939, and 4576 unique cutoff N-grams. These numbers correspond to the feature vector lengths, which are used to construct decision trees in a following section.

3.3. Interaction parameters

Interaction parameters were used for dialog quality evaluation (Schmitt et al., 2010; Engelbrecht and Möller, 2010; Möller, 2005), and one standardization has been established as the ITU-T Recommendation (ITU-T, 2005). In this study, we extracted 13 interaction parameters whose definitions and average parameters are shown in Table 5. Note that parameters REJ-GMM and REJ-TRG are the extended parameters for our MusicNavi2 system and its corpus. Since the parameters in Table 5 are automatically extracted from the system logs, they include errors caused by the speech recognition function of the system, as was the case with the automatic user tags.

4. Structure of decision trees

We trained decision trees to detect task-incompletion using the WEKA toolkit (Hall et al., 2009). The trees were

[1] We also tried using $N = 4$ and higher N-grams, however they contributed little to improving our analysis, as noted in the discussion below.

Table 3: Recall of tag classification and relative frequencies of major/minor errors for manual tags.

Tag (Manual)	Recall	Rel. frequency of major error		Rel. frequency of minor error	
REQUEST-BYARTIST	46.7 %	CATCHNOISE-GMM	15.3 %	REQUEST-BYMUSIC	6.9 %
REQUEST-BYMUSIC	55.3 %	CATCHNOISE-GMM	12.0 %	UNDEFINED	6.5 %
ANSWER-YES	64.9 %	CATCHNOISE-GMM	28.2 %	ANSWER-NO	3.9 %
ANSWER-NO	83.0 %	CATCHNOISE-GMM	12.3 %	REQUEST-BYMUSIC	2.3 %
CMD-THESONG	66.6 %	CATCHNOISE-GMM	23.7 %	REQUEST-BYMUSIC	3.0 %
CMD-NEXTSONG	79.4 %	CATCHNOISE-GMM	11.7 %	ANSWER-NO	3.1 %
CMD-PREVIOUSSONG	79.3 %	CATCHNOISE-GMM	10.4 %	REQUEST-BYMUSIC	4.6 %
CMD-HELLO	65.6 %	CATCHNOISE-GMM	15.9 %	ANSWER-NO	5.5 %
CMD-MICTEST	61.5 %	CATCHNOISE-GMM	22.7 %	CMD-PREVIOUSSONG	5.7 %
CMD-RESPEECH	77.5 %	CATCHNOISE-GMM	10.4 %	REQUEST-BYMUSIC	2.6 %
CMD-BACKHISTORY	69.8 %	CATCHNOISE-GMM	14.5 %	REQUEST-BYMUSIC	8.2 %
CMD-STOP	62.3 %	CATCHNOISE-GMM	24.2 %	REQUEST-BYMUSIC	3.3 %
CMD-INITIALIZE	81.4 %	CATCHNOISE-GMM	9.9 %	ANSWER-NO	3.2 %
CMD-EXIT	68.0 %	ANSWER-NO	10.4 %	CATCHNOISE-GMM	6.5 %
PUSH-INITIALIZEBUTTON	100.0 %		—		—
CATCHNOISE-TOOSHORT	100.0 %		—		—
CATCHNOISE-GMM	81.0 %	CATCHNOISE-ASR	5.5 %	REQUEST-BYMUSIC	3.1 %
CATCHNOISE-ASR	62.5 %	CATCHNOISE-GMM	25.0 %	ANSWER-YES	12.5 %
UNDEFINED	8.8 %	CATCHNOISE-GMM	35.5 %	REQUEST-BYMUSIC	12.5 %

Table 4: Precision of tag classification and relative frequencies of major/minor errors for automatic tags.

Tag (Automatic)	Precision	Rel. frequency of major error		Rel. frequency of minor error	
REQUEST-BYARTIST	76.4 %	REQUEST-BYMUSIC	7.2 %	UNDEFINED	6.9 %
REQUEST-BYMUSIC	68.8 %	UNDEFINED	10.7 %	CATCHNOISE-GMM	9.7 %
ANSWER-YES	80.8 %	UNDEFINED	6.8 %	CATCHNOISE-GMM	4.7 %
ANSWER-NO	33.3 %	CATCHNOISE-GMM	17.7 %	UNDEFINED	14.8 %
CMD-THESONG	39.8 %	UNDEFINED	17.7 %	REQUEST-BYMUSIC	16.1 %
CMD-NEXTSONG	47.5 %	REQUEST-BYMUSIC	14.8 %	CATCHNOISE-GMM	11.3 %
CMD-PREVIOUSSONG	41.0 %	UNDEFINED	16.0 %	REQUEST-BYMUSIC	15.2 %
CMD-HELLO	69.0 %	CATCHNOISE-GMM	9.1 %	REQUEST-BYMUSIC	7.6 %
CMD-MICTEST	48.6 %	REQUEST-BYMUSIC	17.2 %	UNDEFINED	14.5 %
CMD-RESPEECH	57.9 %	REQUEST-BYMUSIC	12.2 %	UNDEFINED	12.2 %
CMD-BACKHISTORY	14.6 %	REQUEST-BYMUSIC	26.4 %	UNDEFINED	23.5 %
CMD-STOP	72.7 %	UNDEFINED	8.9 %	REQUEST-BYMUSIC	6.8 %
CMD-INITIALIZE	52.4 %	REQUEST-BYMUSIC	15.1 %	REQUEST-BYARTIST	12.2 %
CMD-EXIT	62.5 %	UNDEFINED	14.0 %	REQUEST-BYMUSIC	12.4 %
PUSH-INITIALIZEBUTTON	100.0 %		—		—
CATCHNOISE-TOOSHORT	67.5 %	UNDEFINED	26.6 %	ANSWER-YES	1.9 %
CATCHNOISE-GMM	71.5 %	UNDEFINED	8.5 %	ANSWER-YES	4.9 %
CATCHNOISE-ASR	0.2 %	CATCHNOISE-GMM	87.1 %	UNDEFINED	8.3 %
UNDEFINED	24.9 %	REQUEST-BYMUSIC	26.7 %	CATCHNOISE-GMM	22.2 %

constructed based on information gain, and its maximum depth was constrained to 5 [2]. Two decision trees were constructed, one for manual tags and one for automatic tags. Input features were vectors consisting of N-gram frequencies of automatic utterance-and-behavior tags for each dia-

[2]We used the 'REPtree' method of WEKA, which can more easily control the depth of the tree than 'J48' method, with only a small decrease in classification performance.

Table 5: Interaction parameters of dialog data in corpus, their value and definitions.

Abbreviation of parameters	Value of parameters INCOMPLETE	COMPLETE	Definition of parameter
DD [sec]	62.4	75.6	Overall duration of dialog
STD [sec]	1.81	1.99	Average duration of system turn
UTD [sec]	1.75	1.58	Average duration of user turn
SRD [sec]	0.38	0.74	Average delay of system response
URD [sec]	4.01	3.50	Average delay of user response
#TURN	23.6	26.3	Overall number of turns uttered in dialog
#STURN	12.7	14.5	Overall number of system turns uttered in dialog
#UTURN	10.9	11.8	Overall number of user turns uttered in dialog
#REJ-TRG	5.62	3.70	Overall number of trigger-activated rejections by user; i.e., count where user didn't push trigger button of MusicNavi2
#REJ-GMM	1.74	1.85	Overall number of automatic rejections by GMM
#REJ-ASR	0.25	0.22	Overall number of null results by Automatic Speech Recognition
#BARGE-IN	1.07	1.47	Overall number of user barge-in attempts in dialog
#CANCEL	0.07028	0.00030	Overall number of user cancels attempted in dialog; i.e., count where user pushed initialize button of MusicNavi2

log, or the interaction parameters for each dialog.

4.1. N-gram feature of tag sequence

Figures 2(a) and 2(b) show the structures of the decision trees using 3-gram frequencies with manual tags and automatic tags, respectively. These trees achieved 74.2% and 80.4% discrimination accuracy by 10-fold cross validation, respectively.

These figures show that the automatic tags needed the 3-gram, but that it was not needed with the manual tags. This might be caused by differences in the complexity of the N-gram tag patterns in each corpus. Interestingly, the accuracy rate achieved using the automatic tags was the same as the rate using the manual tags. This was because the automatic tags included the misrecognition errors of the speech recognition system. In other words, although the automatic tag corpus contains only the system's understanding of results and actions, the tags represent the actual interaction between the user and the system.

4.2. Interaction parameters

Figure 3 shows the structure of the tree using interaction parameters. This tree achieved 65.7% discrimination accuracy by 10-fold cross validation.

The results shown in the figure suggest the importance of the parameter "SRD" for task-incomplete detection. The "#CANCEL" parameter was used in the 2nd stage, however this was not important because users could restart their music retrieval dialog using the "cancel button" on the GUI of MusicNavi2. Regarding the usage of the parameters shown in Figure 3, there were 9 important parameters, namely; SRD, #CANCEL, URD, STD, #REJECT-TRG, DD, UTD, URD, and #STURN.

The N-gram features and interaction parameters could be concatenated, and the concatenated parameters could be used to construct a more accurate decision tree than if only individual features are used, but this is a topic for future work.

5. Summary

The causes of task completion errors in spoken dialog systems were analyzed using detection trees with N-gram features of the music retrieval dialog. Results using this method suggested that higher order N-grams are note particularly effective for increasing the detection rate, but also illustrated the importance of the last part of the dialog for detecting task completion.

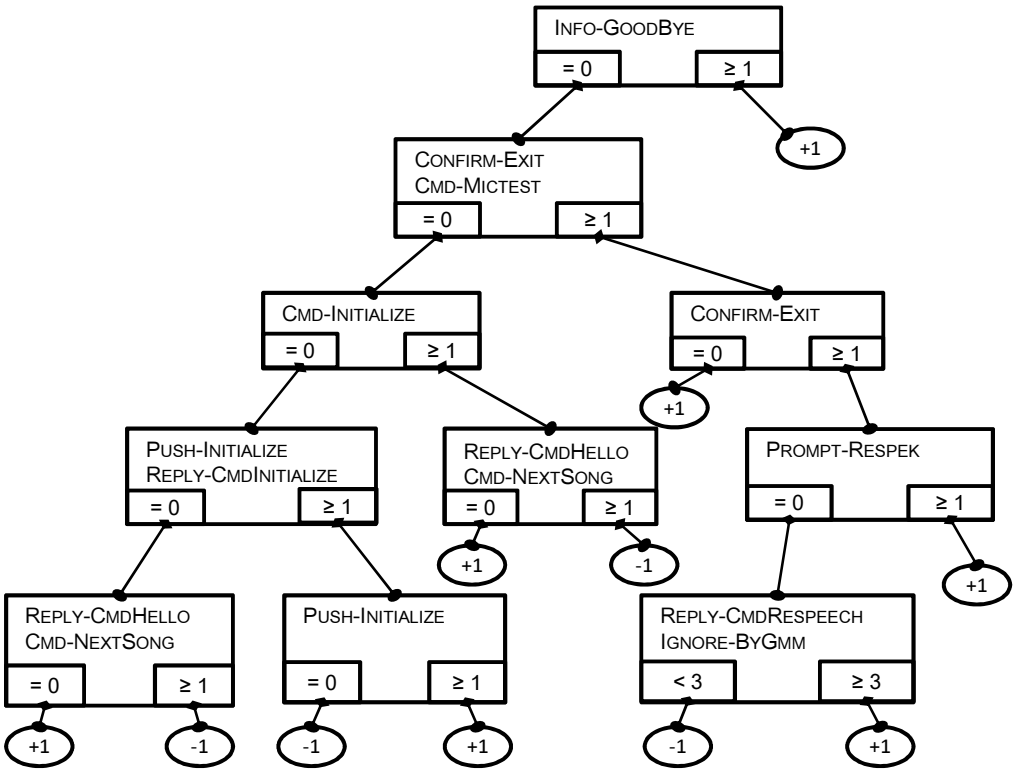
These findings should be applied to methods for detection of task-incompletion dialogs or other problematic dialogs in future work. This study used only linguistic information, however, acoustic information must also be important. This is also a subject for future work.

6. Acknowledgements

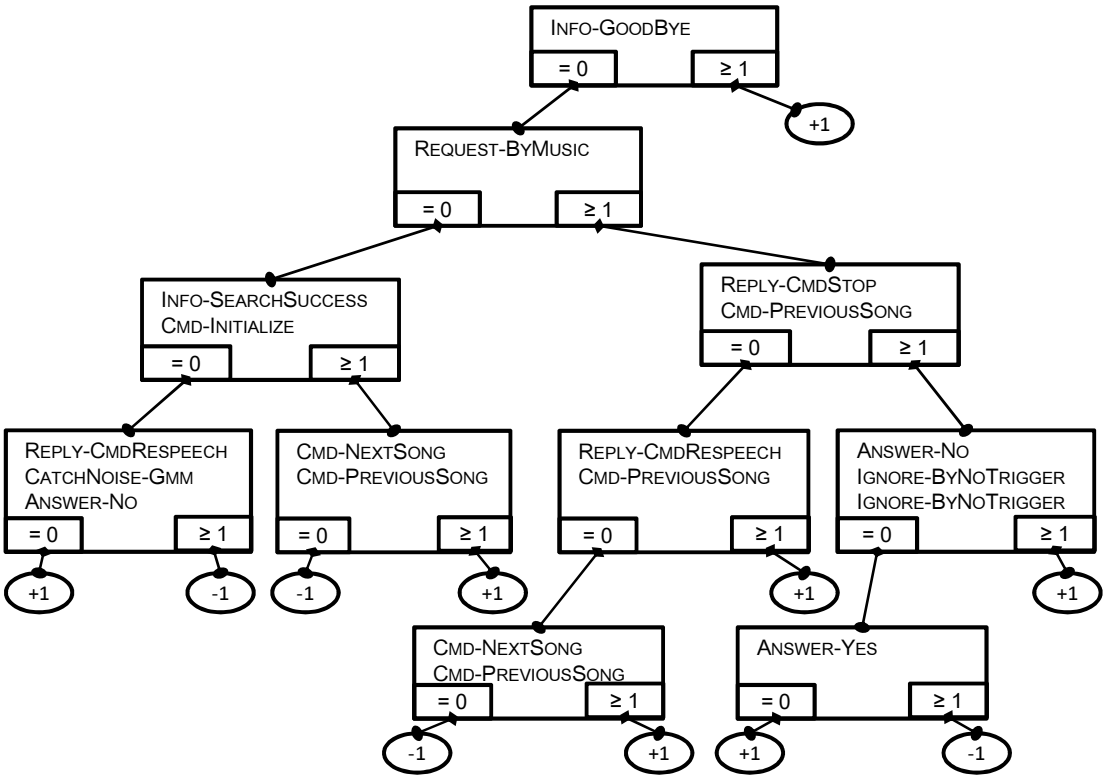
This work was supported in part by the NEDO Grant for Industrial Technology Research Program (07A12203a) and KAKENHI (23500209).

7. References

Klaus-Peter Engelbrecht and Sebastian Möller. 2010. Sequential classifiers for the prediction of user judgments



(a) Using the manual tags



(b) Using the automatic tags

Figure 2: Structure of decision tree based on N-gram features.Each branch contains a 1-gram, 2-gram or 3-gram. Figures in circles represent the chosen class.

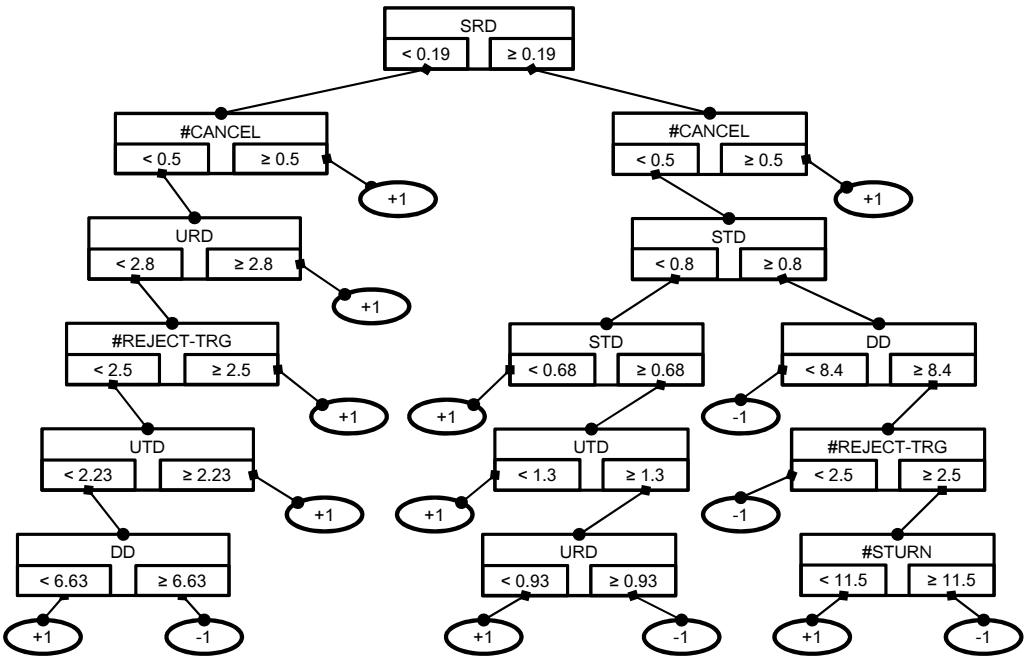


Figure 3: Structure of decision tree based on interaction parameters.

about spoken dialog systems. *Speech Communication*, 52(10):816–833, October.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations Newsletter*, 1(1):10–18, nov.

Sunao Hara, Chiyomi Miyajima, Katsunobu Itou, Norihide Kitaoka, and Kazuya Takeda. 2008. Data collection and usability study of a PC-based speech application in various user environments. In *Proceedings of Oriental COCOSDA 2008*, pages 39–44, November.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2010. Estimation method of user satisfaction using N-gram-based dialog history model for spoken dialog system. In *Proceedings of LREC 2010*, pages 78–83, May.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2011a. Detection of task-incomplete dialogs based on utterance-and-behavior tag n-gram for spoken dialg systems. In *Proceedings of INTERSPEECH 2011*, pages 1305–1308, October.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2011b. On-line detection of task incompletion for spoken dialog systems using utterance and behavior tag N-gram vectors. In Ramon Lopez-Cozar Delgado and Tetsunori Kobayashi, editors, *Proceedings of the Paralinguistic Information and Its Integration in Spoken Dialogue Systems Workshop*, pages 215–225. Springer.

Ota Herm, Alexander Schmitt, and Jackson Liscombe. 2008. When calls go wrong: How to detect problematic calls based on log-files and emotions? In *Proceedings of INTERSPEECH 2008*, pages 463–466, September.

ITU-T. 2005. Parameters describing the interaction with spoken dialogue systems. Recommendation Series P Suppl. 24, International Telecommunication Union, Geneva, May.

Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR engine Julius and Japanese model repository; software of continuous speech recognition consortium. In *Proceedings of ICSLP 2004*, pages 3069–3072, October.

Woosung Kim. 2007. Online call quality monitoring for automating agent-based call centers. In *Proceedings of INTERSPEECH 2007*, pages 130–133, August.

Sebastian Möller. 2005. Parameters for quantifying the interaction. In *Proceedings of SIGdial 2005*, pages 166–177, September.

Alexander Schmitt, Michael Scholz, Wolfgang Minker, Jackson Liscombe, and David Sündermann. 2010. Is it possible to predict task completion in automated troubleshooters? In *Proceedings of INTERSPEECH 2010*, pages 94–97, September.

Marilyn A. Walker, Irene Langkilde-Geary, Helen Wright Hastie, Jerry Wright, and Allen Gorin. 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, 16:293–319, May.