STATISTICAL DIALOG MANAGEMENT APPLIED TO WFST-BASED DIALOG SYSTEMS

Chiori Hori†‡, Kiyonori Ohtake†‡, Teruhisa Misu†‡,
Hideki Kashioka†‡, Satoshi Nakamura†‡

† National Institute of Information and Communications Technology (NICT)
‡Advanced Telecommunications Research Institute International (ATR)
chiori.hori@{nict.go.jp, atr.jp}

ABSTRACT

We have proposed an expandable dialog scenario description and platform to manage dialog systems using a weighted finite-state transducer (WFST) in which user concept and system action tags are input and output of the transducer, respectively. In this paper, we apply this framework to statistical dialog management in which a dialog strategy is acquired from a corpus of human-to-human conversation for hotel reservation. A scenario WFST for dialog management was automatically created from an N-gram model of a tag sequence that was annotated in the corpus with ***Interchange Format*** (IF). Additionally, a word-to-concept WFST for spoken language understanding (SLU) was obtained from the same corpus. The acquired scenario WFST and SLU WFST were composed together and then optimized. We evaluated the proposed WFST-based statistic dialog management in terms of correctness to detect the next system actions and have confirmed the automatically acquired dialog scenario from a corpus can manage dialog reasonably on the WFST-based dialog management platform.

Index Terms— WFST-based dialog system, statistical dialog management, Interchange Format

1. INTRODUCTION

A flexible platform to integrate different dialog strategies and modalities is indispensable for expandable and adaptable dialog systems. We have proposed a dialog scenario description and a platform of dialog management using weighted finite-state transducers (WFSTs) [1]. Although WFSTs are mainly used in speech and language processing [2], we use them for dialog management where input symbols of the WFST are concept tags indicating user's intention while its output symbols are action tags indicating system actions. Concept tags are translated into the corresponding system actions by the WFST.

To construct a dialog system, it is necessary to manually or automatically design a scenario which handles dialog in response to user' input so as to accomplish a task efficiently.

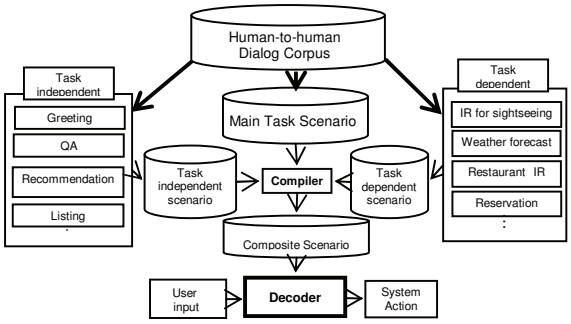

Figure 1: A corpus based dialog system

Recently, there have been studied into statistical approaches to dialog management [3] [4] [5]. The WFST-based dialog management enables us to use such statistical dialog scenarios automatically acquired from annotated corpora.

Figure 1 shows our WFST-based dialog system. The main task and task dependent/independent scenarios are compiled using the WFST operations [6]. This paper presents an automatic dialog system construction using a human-to-human dialog corpus for hotel reservations in which Interchange Format (IF) is annotated. The IF is an interlingua for machine translation used by the C-STAR consortium for task oriented dialogs [7]. We used the IF tags as the user concept and the system action tags. The spoken language understanding (SLU) and the dialog scenario WFSTs were automatically obtained from the annotated corpus.

The statistical model of the system action [5] and that of the user's action [8] were investigated independently. In this study, we constructed a dialog scenario WFST obtained from the IF tag sequence of both the clerk and customer sides. This model determines the system's next action based on probabilities of multiple user concept hypotheses conditioned by the previous system action. We evaluated the detection accuracy for the system's next actions in response to the user's inputs with and without the conditional probabilities of the user concept. This paper presents a potential of the WFST-based dialog management platform for statistical dialog management acquired from a corpus.

2. WFST-BASED DIALOG MANAGEMENT

To design a complex scenario, we may need to combine several scenarios written in different fashions such as finite-state automaton, frame-based representation, if-then rules, etc. Since the WFST framework provides us a general representation, many types of scenarios can be converted into WFSTs. Once a scenario is represented in a WFST, it can be combined with other WFSTs and driven with our WFST-based dialog manager. Furthermore, additional knowledge can be easily incorporated in scenario WFSTs. For example, after a human draw a non-deterministic WFST as a scenario, *n*-gram probabilities of tags in a dialog corpus can be attached to the scenario WFST so that the dialog system behaves naturally as in the corpus.

Figure 2 shows a WFST-based dialog system. In the framework of our WFST-based dialog management, several dialog scenarios and functions of system actions are separately designed. As shown in Fig. 1, to accomplish expandability of a dialog manager, an individual WFST for each task is prepared and then integrated into a main scenario WFST. Additionally, task dependent/independent WFSTs are separately prepared and task independent WFSTs are shared through dialogs.

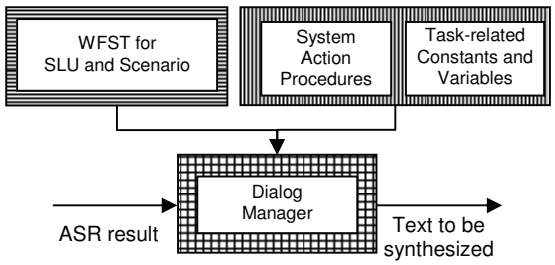

Figure 2: WFST-based dialog system

We use WFSTs for dialog management where the input of the WFST is given by a user, which is a word or concept sequence, and then translated into an output sequence using the WFST. Each symbol in the output sequence corresponds to a system action. Although the WFST-based dialog management is basically equivalent to that by the conventional finite-state automaton, it can be designed with more flexibility since there are many useful operations for WFSTs to combine and optimize.

Figure 3 shows an example of a WFST. The nodes and arcs correspond to states and transitions of the WFST. The label on each arc denotes "input-symbol : output-symbol / weight," and the final state possesses a final weight. A meta symbol "ε" indicates there is no symbol to input or output.

Given an input symbol sequence to a WFST, the output symbol sequence can be obtained as that on the best path with the minimum (or maximum) cumulative weight. The best path can be found efficiently with Dynamic Programming from among successful paths from the initial to one of the finals, which accept the input sequence. In dialog management, however, the system has to respond to the user immediately in each turn. Thus the system needs to choose the most appropriate output sequence according to the current situation. Our WFST-based management includes such a decision process and can also deal with uncertainty during dialog, i.e. when the WFST is non-deterministic, the manager stays at multiple states simultaneously at each turn, which are considered as hidden states. We presented a detailed algorithm of the WFST-based dialog management in the previous paper [1].

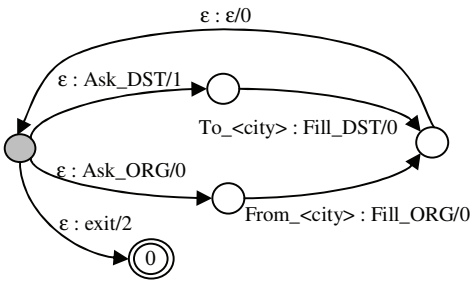

Figure 3: Example of WFST for dialog management (Ask_ORG and Ask_DST indicate system actions to ask the origin and destination cities, respectively. These actions include a meta control that eliminates the transition if the slot has already been filled. Fill_ORG and Fill_DST indicate actions to fill slots according to the user concept such as From_<city> and To_<city>).

3. AUTOMATIC CORPUS-BASED DIALOG SYSTEM CONSTRUCTION

3.1. Interchange Format

The representation of the Interchange Format (IF) is **"Speaker : speech act + concept* (argument*)"** [7]. The speech act represents the speaker's intention. The concept sequence, which may contain zero or more concepts, represents the focus of semantic dialog units. The speech act and concept sequence are collectively referred to as "the domain action". The arguments encode specific information from the utterance using a feature-value format.

3.2. Simulated dialog for hotel reservations

The corpus of simulated dialog for hotel reservation between an English/Japanese speaker and a Japanese speaker were used to construct a dialog system. The dialogs between English and Japanese speakers were done through an interpreter. We denote dialogs between a Japanese hotel clerk and a Japanese customer as **J-J** and those between an English hotel clerk and a Japanese customer as **E-J**. This data is annotated with the IF tags described in the previous section. In this study, each natural language expression in the utterances was attached to each argument to construct spoken

Table 1: Example of E-J hotel reservation dialog annotated with IF

English Hotel Clerk		Japanese Customer	
English/Japanese	IF tag for system action	IF tag for user concept	Japanese/English
Hello, / ありがとうございます、New York City Hotel,/ニューヨークシティホテルでございます。 May I help you? / ご用件をお伺いいたします。	IF[1][1]a:greeting IF[1][2]a:introduce-self(affiliation={{New York City Hotel},{ニューヨークシティホテル}}) IF[1][3]a: offer + help	IF[2][1]c: greeting IF[2][2]c:introduce-self(person-name={{Hiroko Tanaka},{田中弘子}}) IF[2][3]c: request action + reservation + features + hotel	もしもし、/ Hello, わたし田中弘子といいますが/ my name is Hiroko Tanaka and) そちらのホテルの予約をしたいのですが。/ I would like to make a reservation for a room at your hotel.
Very good. / かしこまりました。 May I have the spelling of your name, please? / お客様のお名前のスペルをいただけますでしょうか。	IF[3][1]a:acknowledge IF[3][2]a:request information + spelling (letters={})	IF[4][1]c:acknowledge IF[4][2]c:give information + spelling (letters={{T-A-N-A-K-A},{ティーエーエヌエーケイエー}})	はい。/Yes. ティーエーエヌエーケイエーです。/It's T-A-N-A-K-A.

language understanding WFSTs. Table 1 shows pairs of original utterance and its translation with an IF tag.

3.3. Spoken Language Understanding WFST

A spoken-language-understanding (SLU) WFST for each system was constructed using a set of *n*-word phrases ($n \leq 5$) extracted from the transcripts of the conversations. These phrases were automatically selected as representative expressions for each IF tag. The SLU WFST was designed as a key-phrase detector that translates sentences including such phrases to the corresponding concept tags. Table 2 shows an example of the representative phrases for the tag "c:accept-features-room".

Table 2: Example of frequent *n*-word phrases for c:accept-features-room tag. (Each string in brackets stands for a keyword class at which some corresponding keywords can be accepted.)

a,(room-type:superior-type),(room-type:size),for,(price:quantity)
(room-type:size),for,(price:quantity),(price:currency),please
will,take,a,(room-type:superior-type),(room-type:size)
take,a,(room-type:superior-type),(room-type:size),for
a,(room-type:superior-type),(room-type:size),for
one,that,costs,(price:quantity),(price:currency)
one,at,(price:quantity),(price:currency),please
for,(price:quantity),(price:currency),please
that,costs,(price:quantity),(price:currency)
a,room,for,(price:quantity),(price:currency)
……

3.4. Scenario WFST

A statistical dialog scenario was trained using a sequence of IF tags in the corpus. In this study, we converted the back-off N-gram probabilities of the IF tags into a dialog scenario WFST where the user concept tags are placed on the input side and the system action tags are placed on the output side of the WFST arcs. Since this N-gram model has both prediction powers for system actions and user concepts, which are effective on deciding the next system actions and disambiguating the concepts for the user's natural language input. In this study, we compared the detection accuracy for the system's next action with and without predicting the user's next action.

3.5. Slot-handling

We defined the system's slots using the argument tags of the IF tags. To avoid a system action to request values for slots which have already been filled, the system has a meta control to intercept transitions according to whether the slots are filled or not. If all slots required for task completion are filled, the system can take transitions to final states.

4. EVALUATION EXPERIMENTS

4.1 Evaluation data

To validate performance of WFST-based statistical dialog management, we constructed Japanese and English dialog systems for hotel reservations using the corpus. The Japanese one was acquired from J-J conversations without an interpreter while the English one was acquired from E-J conversations via an interpreter. The features of the corpus are shown in Table 3 and 4.

Table 3. Number of IF tags in the corpus

	# User concept tag	# System action tag
J-J	94	131
E-J	59	86

Table 4. Number of dialogs and turns used in the systems

J-J dialog system			
		User	System
Training (66 dialogs)	#turn/dialog	16.74 (1138/68)	17.60 (1197/68)
	#tag/turn	1.59 (1810/1138)	2.12 (2541/1197)
Test set (3 dialogs)	#turn/dialog	7.00 (21/3)	8.00 (24/3)
	#tag/turn	2.00 (42/21)	4.00 (96/24)
E-J dialog system			
		User	System
Training (22 dialogs)	#turn/dialog	11.18 (246/22)	11.59 (255/22)
	#tag/turn	1.78 (438/246)	2.83 (722/255)
Test set (3 dialgos)	#turn/dialog	8.33 (25/3)	9.33 (28/3)
	#tag/turn	1.84 (46/25)	2.50 (70/28)

4.2. Evaluation result

We constructed bigram, trigram, and 4-gram models of the IF tag sequence and investigated the performance to predict

the next system actions using these models. Table 5 shows the test-set perplexities (PPs) for these models. The unseen tag rates in the test sets are 4.9% and 5.7%, respectively.

Table 5: Test-set Perplexity of N-gram models

	Bigram	Trigram	4-gram
J-J	28.7	23.6	23.7
E-J	43.1	39.0	37.3

Since the trigram PP was almost equal to the 4-gram PP in each set, we used trigram models for constructing the scenario WFSTs. Each scenario WFST was then composed with the SLU WFST and optimized. The sizes of WFSTs are summarized in Tables 6 and 7.

Table 6: Size of WFSTs for J-J

WFSTs	#state	#transition
Scenario 3-gram	1,776	12,926
SLU	19,611	34,057
Composed	167,187	332,907
Optimized	28,880	257,780

Table 7: Size of WFSTs for E-J

WFSTs	#state	#transition
Scenario 3-gram	753	3,453
SLU	26,339	40,259
Composed	21,808	184,295
Optimized	15,724	140,743

To measure the performance to predict the next actions, we force correct transitions to the WFST according to the reference dialog in the test set, and made the system predict the next action tag sequence right after giving the user's input at each turn. We ranked all possible action tag sequences that can be taken by the system, where each possible sequence was weighted according to the corresponding path to the tag sequence. We calculated mean reciprocal rank (MRR) based on the rank of the correct action tags in the reference dialog. MRR is defined as:

$$MRR = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{R_i},$$

where R_i is the rank of the correct system action tag sequence at i-th turn, and M is the number of system turns. A larger MRR indicates a better prediction. Table 6 shows the MRR values in some conditions. We measured MRR with the integrated WFST and another WFST which is weightless only for predicting user concept tags where N-gram probabilities occurring user concepts are all set to 1. In Table 8, we can see the difference of MRR between 0.097 and 0.094 (roughly rank 10th) or between 0.174 and 0.164 (roughly rank 4th). This indicates that prediction of user's input by the scenario N-gram model contributes to select appropriate system actions, which can be easily achieved by composition of SLU and scenario N-gram WFSTs.

Table 8: Prediction performance for system actions(MRR)

	All weighted	*Weightless for concepts*
J-J	0.097	0.094
E-J	0.174	0.164

In this experiment, the speech acts in J-J are much more complicated than those in E-J because the conversation in J-J was very spontaneous without interpretation. As a result, the number of tags in each turn in J-J is larger than that in E-J. We confirmed the resulting WFSTs can be used to manage the hotel reservation dialogs with the user's natural language input and system action output. However, the IF was not designed for dialog management. To enhance the performance of the WFST-based dialog management using the IF, we need to map each argument to more appropriate slots for managing the hotel reservation task.

5. CONCLUSION

We have proposed an efficient approach to manage a dialog system using a weighted finite-state transducer (WFST). A WFST for dialog scenarios was automatically created using a hotel reservation dialog corpus with Interchange Format (IF) tags. Another WFST for spoken language understanding (SLU) was also created using the same corpus. The scenario and SLU WFSTs were composed together and then optimized. In conclusion, we have confirmed the WFST-based dialog system can be used for statistical dialog management, and the integrated WFST has a good performance for predicting the next system actions. The future work involves evaluation experiments by humans via speech input and output. Currently, a sentence generation module has not been implemented yet, which may be designed using WFSTs as well.

6. REFERENCE

[1] C. Hori et al., "Dialog management using weighted finite-state transducers," Interspeech 2008.

[2] M. Mohri et al., "Weighted finite-state transducers in speech recognition," Computer Speech and Language, 2002.

[3] E. Levin et al., "Learning dialogue strategies within the markov decision process framework", ASRU 1997.

[4] J. William et al., "Partially Observable Markov Decision Processes for Spoken Dialog Systems." Computer Speech and Language 21(2): pp. 231-422, 2007.

[5] Hurtado, L.F. et al., "A stochastic approach to dialog management", ASRU 2005.

[6] L. Hetherington: "The MIT finite-state transducer toolkit for speech and language processing", Interspeech 2004.

[7] L. Levin et al., "Evaluation of a practical interlingua for task-oriented dialogue," NAACL-ANLP 2000 Workshop on Applied interlinguas.

[8] M. Nagata et al., "First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance," Speech Communication 1994.