Recent Advances in WFST-based Dialog System

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

Spoken Language Communication Group
MASTAR project
National Institute of Information and Communications Technology (NICT)
chiori.hori@nict.go.jp

Abstract

To construct an expandable and adaptable dialog system which handles multiple tasks, we proposed a dialog system using a weighted finite-state transducer (WFST) in which users concept and system action tags are input and output of the transducer, respectively. To test the potential of the WFST-based dialog management (DM) platform using statistical DM models, we constructed a dialog system using a human-to-human spoken dialog corpus for hotel reservation, which is annotated with Interchange Format (IF). A scenario WFST and a spoken language understanding (SLU) WFST were obtained from the corpus and then composed together and optimized. We evaluated the detection accuracy of the system next actions. In this paper, we focus on how WFST optimization operations contribute to the performance of the system. In addition, we have constructed a full WFST-based dialog system by composing SLU, scenario and sentence generation (SG) WFSTs. We show an example of a hotel reservation dialog with the fully composed system and discuss future work.

Index Terms: spoken dialog, weighted finite-state transducer (WFST), statistical dialog management, Interchange Format (IF)

1. Introduction

There are many platforms to support rapid prototyping of dialog systems and control dialogs while handling speech or multimodal interfaces. Some of them employ a textual language such as VoiceXML\(^1\) to describe dialog scenarios. Some systems accept table or automaton representation that can be designed with a graphical editor [1]. Some other systems enable developers to use existing programming languages [2]. To construct an expandable and adaptable dialog system which handles multiple tasks, we have proposed an efficient approach to manage a dialog system using a weighted finite-state transducer (WFST) in which users concept and system action tags are input and output of the transducer, respectively [3]. Since the WFST-based dialog manager can deal with many types of scenarios from rule-based to statistical ones but does not have excessive freedom as general programming languages, WFSTs are suitable for a unified description of dialog management. In addition, once all scenarios are represented in WFSTs, these can be combined with other WFSTs and driven with our WFST-based dialog management platform as shown in Fig. 1.

\(^1\) Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Recommendation 16 March 2004
by composing SLU, DM and sentence generation (SG) WFSTs. The SG WFST was also acquired from the corpus. We show an example of a hotel reservation dialog with the fully composed system.

2. Weighted Finite-State Transducer based Dialog Management

Figure 2 shows an example of a scenario 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 of the double circle possesses a final weight.

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’s concept such as From_<city> and To_<city>. A meta symbol “ε” indicates no symbol to input or output is needed in taking the transition.

3. Corpus-based Dialog System Construction

Dialog scenarios and user concept understanding in the conventional dialog systems are mostly generated by hand. However, linguistic expressions are varied human to human even speakers’ intentions are same. Furthermore, humans’ dialog behaviors are not always the same. It is very difficult to prepare all rules for possible human responses in dialog manually. To construct more robust dialog systems, we exploit corpus-based dialog models. Humans have typical patterns of dialog especially when a target task is limited and thus statistical models of dialog scenario could be learned from a dialogue corpus. To cover more pairs of intentions and linguistic expressions, corpus-based spoken language understanding is needed.

In this study we constructed a corpus-based dialog system using a human-to-human dialog corpus for hotel reservation, which is annotated with Interchange Format (IF) [3]. The representation of the Interchange Format (IF) is “Speaker ID: speech act + concept* (argument*)”. To construct spoken language understanding (SLU) and sentence generation (SG) WFSTs, we extracted natural language expressions for each argument from the utterances. The dialog system driven by the corpus-based dialog scenario responds as the hotel clerks behave in the corpus.

3.1. Spoken Language Understanding WFST

A spoken-language-understanding (SLU) WFST for each system was constructed using a set of n-word phrases (n ≤ 5) extracted from the transcripts of the conversations based on relative frequency of n-word phrases. These phrases were automatically selected as representative expressions for each IF tag [3]. If slots are designed and slot values are defined, the slot value can be extracted as keywords. The representative expressions are defined as a combination of key-phrases and keywords. The SLU WFST was designed as a key-phrase detector that translates sentences including such phrases to the corresponding concept tags. This loose key phrase matching enables the system to accept the phrases which have the same meaning with the small difference in spontaneous expressions. Figure 3 shows an example of a SLU WFST.

3.2. Scenario WFST

A statistical dialog scenario was trained using a sequence of IF tags in the corpus. Although there are alternatives in choosing responses for user, the scenario WFST enables for the dialog system to determine which system action is taken in response to a user input according to each state of a dialog discourse.

3.3. Sentence Generation WFST

In this paper, we have introduced a sentence generation (SG) WFST which translates a system action tag sequence into the corresponding sentences. The SG WFST can also be obtained from the corpus. We have designed it as a back-off bigram model of tags where the each system action tag is mapped to a set of sentences with the same tag in the corpus. By using the bigram model, the SG WFST can select a reasonable sentence at the current dialog context.

3.4. Dialog Management WFST

The SLU, scenario, and SG WFSTs were combined and then optimized using the WFST operations. The final composed WFST is denoted a dialog management (DM) WFST. To optimize the DM WFST, pushing and determinization operations were performed. Weights are shifted to the initial state by pushing and the number of transitions is minimized by determinization.

4. Statistical Dialog Management

4.1. N-gram based scenario WFST

An N-gram model can be converted into a WFST [8]. 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. Figure 4 shows an example of a tag sequence.

![Figure 4: Example of tag sequence.](image)

Supposing \( N \) is 3, i.e. a trigram model, the next system action is determined by

\[
\hat{A}_t = \arg \max_{A_t} \left\{ \max_{C_t \in C(U_t)} P(C_t|A_{t-1})P(A_t|A_{t-1},C_t) \right\},
\]

where \( C(U_t) \) is a set of concepts which accept utterance \( U_t \), \( P(C_t|A_{t-1}) \) is a prediction probability of user concept, and \( P(A_t|A_{t-1},C_t) \) is a prediction probability of system action. Figure 5 shows an example of choosing the best system’s next action among the multiple hypotheses using trigram.

Since this 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. When w/o predicting user concept \( C_t \), \( P(C_t|A_{t-1})=1 \) is applied.

### 4.2. Function of Back-off and Slot-handling

The dialog system needs to let users to input freely as long as the input is limited in the target domain. The corpus-based scenario cannot handle user inputs which are not accepted by spoken language understanding and an unobserved concatenation of system action and user response in the corpus. Suppose dialog discourse is the same as recorded in the corpus, the system just take the same action in the corpus. However, the size of training data is not sufficient to cover all dialog discourse in the real world. Although our scenario model does not cover all tag sequences, concatenating part of tag sequence in the corpus has potential to let users to accomplish the target task.

To accept unexpected user input, the system next action is determined based on back-off N-gram probabilities, i.e., \( \alpha (A_t, C_t)P(A_t|C_t) \) is applied instead of \( P(A_t|A_{t-1},C_t) \), where \( \alpha (A_t, C_t) \) is the back-off coefficient. Since the dialog history is cancelled by backing off, the system can find the most likely action even for the unexpected input.

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. We defined the system’s slots using the argument tags of the IF tags in this work.

### 4.3. Evaluation Experiment

#### 5.1. Evaluation data

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. 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 number of user concept tags and system action tags are 94 and 131 in J-J and 59 and 86 in E-J. The number of tags in J-J is larger than that of E-J because the conversation in J-J was very spontaneous without interpretation. The averaged number of turns in each dialog is roughly 17 in J-J and 11 in E-J [3].

A “turn” was defined as a set of user concept \( (c_t) \) and system action \( (a_t) \) tags from a user input right after a system action to a system action before next user input. Suppose \( \{c_0, c_1, c_2, a_0, a_1, a_2, a_3, a_4, c_5, a_5\} \) is given, it is split into three turns as \( \{c_0, c_1, c_2, a_0, a_1\}, \{c_5, a_2, a_3\} \) and \( \{c_4, c_5, a_4\} \).

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 [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. To evaluated the performance to predict the next system action, each scenario WFST was then composed with the SLU WFST and optimized.

#### 5.2. Evaluation Metrics

To measure the performance of prediction for system’s 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. In \cite{2}, we measured MRR and obtained 0.097 for J-J (roughly rank 10th), and 0.174 for E-J (roughly rank 4th). There actually exist a lot of acceptable responses in the result although they do not match the reference.

5.3. Evaluation Result for WFST optimization

Figure 6 shows the MRR in response to various beam width in dialog decoding. The wider beam resulted in the better MRR. When comparing with and without optimization, the optimized WFST obtained the best MRR even when the beam width is narrow. Especially, determination only worked better than pushing only. Figure 7 shows response time per turn when changing the beam width. Even 100,000 hypotheses are considered, it takes only 25 [msec] to determine the system’s next actions using a standard personal computer. The WFST-based DM can respond to users in time.

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 originally 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.

5.4. Dialog using a fully-composed WFST

Finally we have constructed a full WFST-based dialog system by composing SLU, scenario and sentence generation (SG) WFSTs. The resulting DM WFST translates user’s utterances into system’s response sentences directly. We show an example of a hotel reservation dialog using the DM WFST in Table 1. For the user’s utterances (USR), the system is responding by the natural sentences (SYS). Although some keyword class names in parentheses still remain in the sentences, they can be substituted with the actual keywords based on prior settings and slot values extracted from the user’s utterances. As shown in the table, the fully-composed WFST can manage the dialog and generate reasonable sentences.

\[
\begin{align*}
\text{SYS:} & \text{(affiliation).} \\
\text{USR:} & \text{Hello, My name is Suzuki.} \\
\text{SYS:} & \text{Yes, Mr. (person-name).} \text{How can I be of service?} \\
\text{USR:} & \text{I would like to reserve a room.} \\
\text{SYS:} & \text{And what dates are you interested in, sir?} \\
\text{USR:} & \text{Two nights from the day after tomorrow} \\
\text{SYS:} & \text{Okay, And how many people will be in your party?} \\
\text{USR:} & \text{Two adults} \\
\text{SYS:} & \text{All right. And, what sort of room would you like?} \\
\text{USR:} & \text{A twin room please} \\
\text{SYS:} & \text{Just a moment, please.} \\
\text{USR:} & \text{yes} \\
\text{SYS:} & \text{I’m sorry, Mr. (person-name). Unfortunately all (room-type:size) with baths are fully booked for those dates.} \\
\text{USR:} & \text{Do you have another type of room?} \\
\text{SYS:} & \text{Yes, Yes we do have rooms for (for-whom:quantity).} \\
\text{USR:} & \text{Okay, please.} \\
\text{SYS:} & \text{All right. And what time will you be arriving on the (time:day)?} \\
\text{USR:} & \text{Around three p.m.} \\
\text{SYS:} & \text{Very well then, Mr. (person-name). You have a (room-type:size) reserved for (time:month) (time:day) and (time:day). We’ll be expecting you.} \\
\text{USR:} & \text{Thank you. Bye.}
\end{align*}
\]

Table. 1: Example of Dialog with the Fully-composed WFST

6. Conclusion

This paper proposed an efficient approach to manage a dialog system using weighted finite-state transducers (WFSTs) in which concept tags of user’s and system’s actions are input and output of the transducer, respectively. This framework provides an expandable and adaptable platform of dialog systems. A WFST for dialog scenarios was automatically created using a hotel reservation dialog corpus with Interchange Format (IF) tags. Other WFSTs for spoken language understanding (SLU) and sentence generation (SG) were also created using the same corpus. The SLU, scenario, and SG 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 and responding in real time. The future work involves evaluation experiments by humans via speech input and output. We also need to combine it to a back end system linked to external knowledge such as a database.

7. References

\cite{3} C. Hori et al., “Statistical dialog management applied to WFST-based dialog systems,” In Proc. ICASSP2009.
\cite{5} M. Nagata et al., “First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance,” Speech Communication 1994.