

Automatic Agenda Graph Construction from Human-Human Dialogs using Clustering Method

Cheongjae Lee, Sangkeun Jung, Kyungduk Kim, Gary Geunbae Lee

Department of Computer Science and Engineering

Pohang University of Science and Technology

Pohang, South Korea

{lcj80,hugman,getta,gblee}@postech.ac.kr

Abstract

Various knowledge sources are used for spoken dialog systems such as task model, domain model, and agenda. An agenda graph is one of the knowledge sources for a dialog management to reflect a discourse structure. This paper proposes a clustering and linking method to automatically construct an agenda graph from human-human dialogs. Preliminary evaluation shows our approach would be helpful to reduce human efforts in designing prior knowledge.

1 Introduction

Data-driven approaches have been long applied for spoken language technologies. Although a data-driven approach requires time-consuming data annotation, the training is done automatically and requires little human supervision. These advantages have motivated the development of data-driven dialog modelings (Williams and Young, 2007, Lee et al., 2009). In general, the data-driven approaches are more robust and portable than traditional knowledge-based approaches. However, various knowledge sources are still used in many spoken dialog systems that have been developed recently. These knowledge sources contain task model, domain model, and agenda which are powerful representation to reflect the hierarchy of natural dialog control. In the spoken dialog systems, these are manually designed for various purposes including dialog modeling (Bohus and Rudnicky, 2003, Lee et al., 2008), search space reduction (Young et al., 2007), domain knowledge (Roy and Subramaniam, 2006), and user simulation (Schatzmann et al., 2007).

We have proposed an example-based dialog modeling (EBDM) framework using an agenda graph as prior

knowledge (Lee et al., 2008). This is one of the data-driven dialog modeling techniques and the next system action is determined by selecting the most similar dialog examples in dialog example database. In the EBDM framework for task-oriented dialogs, agenda graph is manually designed to address two aspects of a dialog management: (1) Keeping track of the dialog state with a view to ensuring steady progress towards task completion, and (2) Supporting n-best recognition hypotheses to improve the robustness of dialog manager. However, manually building such graphs for various applications may be labor intensive and time consuming. Thus, we have tried to investigate how to build this graph automatically. Consequently, we sought to solve the problem by automatically building the agenda graph using clustering method from an annotated dialog corpus.

2 Related Work

Clustering techniques have been widely used to build prior knowledge for spoken dialog systems. One of them is automatic construction of domain model (or topic structure) which is one of the important resources to handle user's queries in call centers. Traditional approach to building domain models is that the analysts manually generate a domain model through inspection of the call records. However, it has recently been proposed to use an unsupervised technique to generate domain models automatically from call transcriptions (Roy and Subramaniam, 2006). In addition, there has been research on how to automatically learn models of task-oriented discourse structure using dialog act and task information (Bangalore et al., 2006). Discourse structure is necessary for dialog state-specific speech recognition and language understanding to improve the performance by predicting the next possible dialog states. In addition, the discourse structure is essential to determine whether the current utterance in the dialog is part of the current subtask or starts a new task.

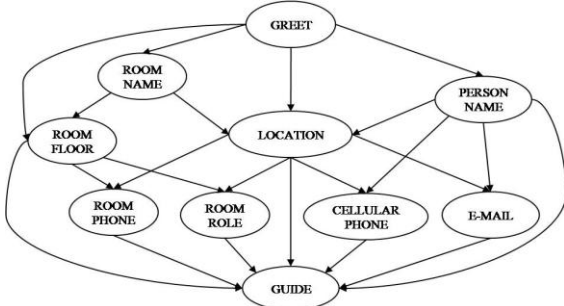


Figure 1: Example of an agenda graph for building guidance domain

More recently, it has been proposed stochastic dialog management such as the framework of a partially observable Markov decision process (POMDP). This framework is statistically data-driven and theoretically principled dialog modeling. However, detailed dialog states in the master space should be clustered into general dialog states in summary space to scale up POMDP-based dialog management for practical applications (Williams and Young, 2007). To address this problem, an unsupervised automatic clustering of dialog states has been introduced and investigated in POMDP-based dialog manager (Lefevre and Mori, 2007).

In this paper, we are also interested in exploring methods that would automatically construct the agenda graph as prior knowledge for the EBDM framework.

3 Agenda Graph

In this section, we begin with a brief overview of EBDM framework and agenda graph. The basic idea of the EBDM is that the next system action is predicted by finding semantically similar user utterance in the dialog state space. The agenda graph was adapted to take into account the robustness problem for practical applications. Agenda graph G is a simply a way of encoding the domain-specific dialog control to complete the task. G is represented by a directed acyclic graph (DAG) (Figure 1). An agenda is one of the subtask flows, which is a possible path from root node to terminal node. G is composed of nodes (v) which correspond to possible intermediate steps in the process of completing the specified task, and edges (e) which connect nodes. In other words, v corresponds to dialog state to achieve domain-specific subtask in its expected agenda. Each node includes three different components: (1) A precondition that must be true before the subtask is executed; (2) A description of the node that includes its label and identifier; and (3) Links to nodes that will be executed at the subsequent turn. In this system, this graph is used to rescore n-best ASR hypotheses and to interpret the discourse state such as new task, next task, and new subtask based on topological position on the graph. In the agenda graph G , each node holds a set of relevant dialog

Feature Types	Features	#Size
Word-level features	unigram	175
	bigram	573
	trigram	1034
Utterance-level features	dialog act (DA)	9
	main goal (MG)	16
	slot filling status	8
Discourse-level features	system act (SA)	26
	previous DA	10
	previous MG	17
	previous SA	27

Table 1: List of feature sets

examples which may appear in the corresponding dialog states when a precondition of the node is true. To determine the next system action, the dialog manager first generates possible candidate nodes with n-best hypotheses by using a discourse interpretation algorithm based on the agenda graph, and then selects the focus node which is the most likely dialog state given the previous dialog state. Finally the best example in the focus node is selected to determine appropriate system action.

Human efforts are required to manually design the agenda graph to integrate it into the EBDM framework. However, it is difficult to define all possible precondition rules and to assign the transition probabilities to each link based only on the discretion of the system developer. To solve these problems, we tried to construct the agenda graph from the annotated dialog corpus using clustering technique.

4 Clustering and Linking

4.1 Node Clustering

Each precondition has been manually defined to map relevant dialog examples into each node. To avoid this, the dialog examples are automatically grouped into the closest cluster (or node) by a node clustering. In this section, we explain a feature extraction and clustering method for constructing the agenda graph.

4.1.1 Feature Extraction

Each dialog example should be converted into a feature vector for a node clustering. To represent the feature vectors, we first extract all n-grams which occur more frequently than a threshold and do not contain any stop word as word-level features. We also extract utterance-level and discourse-level features from the annotated dialog corpus to reflect semantic and contextual information because a dialog state can be characterized using semantic and contextual information derivable from the annotations. The utterance is thus characterized by the set of various features as shown in Table 1.

For a set of N dialog examples $X=\{x_i|i=1,\dots,N\}$, the binary feature vectors are represented by using a set of features from the dialog corpus. To calculate the distance of two feature vectors, we used a cosine measure as a binary vector distance measure:

$$d(x_i, x_j) = 1 - \frac{(x_i \cdot x_j)}{\|x_i\| \cdot \|x_j\|}$$

where x_i and x_j denoted two feature vectors. However, each feature vector contains small number of non-zero terms (<20 features) compared to the feature space (>2000 features). Therefore, most pairs of utterances share no common feature, and their distance is close to 1.0. To address this sparseness problem, the distance between two utterances can be computed by checking only the non-zero terms of corresponding feature vectors (Liu, 2005).

4.1.2 Clustering

After extracting feature vectors from the dialog corpus, we used K -means clustering algorithm which is the simplest and most commonly used algorithm employing a squared error criterion. At the initialization step, one cluster mean is randomly selected in the data set and $k-1$ means are iteratively assigned by selecting the farthest point from pre-selected centers as the following equation:

$$u_k = \arg \max_{x \in X} \left(\sum_{i=1}^{k-1} d(x, u_i) \right)$$

where each cluster c_k is represented as a mean vector u_k . At the assignment step, each example is assigned to the nearest cluster \hat{c}_i by minimizing the distance of cluster mean u_k and dialog example x_i .

$$\hat{c}_i = \arg \min_{1 \leq k \leq K} (d(u_k, x_i))$$

The responsibilities r_{kt} of each cluster c_k are calculated for each example x_t as the following rule:

$$r_{kt} = \frac{\exp\{-\beta \cdot d(u_k, x_t)\}}{\sum_l \exp\{-\beta \cdot d(u_l, x_t)\}}$$

where β is the stiffness and usually assigned to 1.

During the update step, the means are recomputed using the current cluster membership by reflecting their responsibilities:

$$u_k = \frac{\sum_t r_{kt} x_t}{\sum_t r_{kt}}$$

4.2 Node Linking

From the node clustering step, node v_k for cluster c_k is obtained from the dialog corpus and each node contains similar dialog examples by the node clustering algorithm. Next, at the node linking step, each node should be connected with an appropriate transition probability to build the agenda graph which is a DAG (Figure 2).

Algorithm: NodeLinking(C)

Input: C = a set of clusters $\{c_k | c_k \in C, k=1, \dots, K\}$

Output: $G(V, E)$ = agenda graph (directed acyclic graph)

where each $v_k \in V$ is subtask node, each $e_q \in E$ is its directed edges

```

1   $V \leftarrow \text{GENERATENODES}(C)$ 
2   $V \leftarrow V \cup \{v_{root}, v_{terminal}\}$ 
3   $E \leftarrow \text{INTRODUCEDGES}(V, C)$ 
4  for  $i \leftarrow 1$  to  $K$ 
5  do for  $j \leftarrow 1$  to  $K$ 
6  do  $f(v_i, v_j) = n(x \in v_i \rightarrow v_j) / n(x \in v_i)$ 
7  while all edges are visited
8  do  $e_q \leftarrow \text{FINDMINIMALEDGE}(E)$ 
9  if  $f(v_i, v_j) < \delta$  then  $E \leftarrow E - \{e_q\}$  (underweighted edge pruning)
10 else if  $\text{ISCONNECTED}(v_i, v_j)$  then  $E \leftarrow E - \{e_q\}$  (cycle deletion)
11 for  $i \leftarrow 1$  to  $K$ 
12 do for  $j \leftarrow 1$  to  $K$ 
13 do  $p(v_j | v_i) = f(v_i, v_j) / \sum_l f(v_l, v_i)$ 
14 return  $G=(V, E)$ 

```

Figure 2: Node Linking Algorithm

This linking information can come from the dialog corpus because the task-oriented dialogs consist of sequential utterances to complete the tasks. Using sequences of dialog examples obtained with the dialog corpus, relative frequencies of all outgoing edges are calculated to weight directed edges:

$$f(v_i, v_j) = \frac{n(x \in v_i \rightarrow v_j)}{n(x \in v_i)}$$

where $n(x \in v_i)$ represents the number of dialog examples in v_i and $n(x \in v_i \rightarrow v_j)$ denotes the number of dialog examples having directed edge from v_i to v_j . Next some edges are pruned when the weight falls below a pre-defined threshold δ , and the cycle paths are removed by deleting minimal edge in cycle paths through a depth-first traversal. Finally the transition probability can be estimated by normalizing relative frequencies with the remained edges.

$$p(v_j | v_i) = \frac{f(v_i, v_j)}{\sum_l f(v_l, v_i)}$$

5 Experiment & Result

A spoken dialog system for intelligent robot was developed to provide information about building (e.g., room number, room name, room type) and people (e.g., name, phone number, e-mail address). If the user selects a specific room to visit, then the robot takes the user to the desired room. For this system, we collect a human-human dialog corpus of about 880 user utterances from 214 dialogs which were based on a set of pre-defined 10 subjects relating to building guidance task. Then, we designed an agenda graph and integrated it into the EBDM framework. In addition, a simulated environment with a user simulator and an ASR channel (Jung et

al., 2008) was developed to evaluate our approach by simulating a realistic scenario.

First we measured the clustering performance to verify our approach for constructing the agenda graph. We used the manually clustered examples by a set of pre-condition rules as the reference clusters. Table 2 shows error rates when different feature sets are used for K -means clustering in which K is equal to 10 because a hand-crafted graph included 10 nodes. The error rate was significantly reduced when using all feature sets.

Feature sets	Error rate (%)
Word-level features	46.51
+Utterance-level features	34.63
+Discourse-level features	31.20

Table 2: Error rates for node clustering ($K=10$)

We also evaluated the dialog system performance with the agenda graphs which are manually (HC-AG) or automatically designed (AC-AG). We also used 10-best recognition hypotheses with 20% word error rate (WER) for a dialog management and 1000 simulated dialogs for an automatic evaluation. In this result, although the system with HC-AG slightly outperforms the system with AC-AG, we believe that AC-AG can be helpful to manage task-oriented dialogs with less human costs for designing the hand-crafted agenda graph.

System	TCR (%)	AvgUserTurn
Using HC-AG	92.96	4.41
Using AC-AG	89.95	4.39

Table 3: Task completion rate (TCR) and average user turn (AvgUserTurn) (WER=20%)

6 Conclusion & Discussion

In this paper, we address the problem of automatic knowledge acquisition of agenda graph to structure task-oriented dialogs. We view this problem as a first step in clustering the dialog states, and then in linking between each cluster based on the dialog corpus. The experiment results show that our approach can be applicable to easily build the agenda graph for prior knowledge.

There are several possible subjects for further research on our approach. We can improve the clustering performance by using a distance metric learning algorithm to consider the correlation between features. We can also discover hidden links in the graph by exploring new dialog flows with random walks.

Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program

supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0045).

References

- Bangalore, S., Fabrizio, G.D. and Stent, A. 2006. Learning the structure of task-driven human-human dialogs. *Proc. of the Association for Computational Linguistics*, 201-208.
- Bohus, B. and Rudnicky, A. 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *Proc. of the European Conference on Speech, Communication and Technology*, 597-600.
- Jung, S., Lee, C., Kim, K. and Lee, G.G. 2008. An Integrated Dialog Simulation Technique for Evaluating Spoken Dialog Systems. *Proc. of Workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, International Conference on Computational Linguistics*, 9-16.
- Lee, C., Jung, S. and Lee, G.G. 2008. Robust Dialog Management with N-best Hypotheses using Dialog Examples and Agenda. *Proc. of the Association for Computational Linguistics*, 630-637.
- Lee, C., Jung, S., Kim, S. and Lee, G.G. 2009. Example-based Dialog Modeling for Practical Multi-domain Dialog System. *Speech Communication*, 51(5):466-484.
- Lefevre, F. and Mori, R.D. 2007. Unsupervised State Clustering for Stochastic Dialog Management. *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 550-555.
- Liu, Z. 2005. An Efficient Algorithm for Clustering Short Spoken Utterances. *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 593-596.
- Roy, S. and Subramaniam, L.V. 2006. Automatic generation of domain models for call centers from noisy transcriptions. *Proc. of the Association for Computational Linguistics*, 737-744.
- Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H. and Young, S. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. *Proc. of the Human Language Technology/North American Chapter of the Association for Computational Linguistics*, 149-152.
- Williams, J.D. and Young, S. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21:393-422.
- Young, S., Schatzmann, J., Weilhammer, K. and Ye, H. 2007. The Hidden Information State Approach to Dialog Management. *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 149-152.