Sequential Labeling for Tracking Dynamic Dialog States

Seokhwan Kim, Rafael E. Banchs

Human Language Technology Department
Institute for Infocomm Research
Singapore 138632
{kims,rembanchs}@i2r.a-star.edu.sg

Abstract

This paper presents a sequential labeling approach for tracking the dialog states for the cases of goal changes in a dialog session. The tracking models are trained using linear-chain conditional random fields with the features obtained from the results of SLU. The experimental results show that our proposed approach can improve the performances of the sub-tasks of the second dialog state tracking challenge.

1 Introduction

A dialog manager is one of the key components of a dialog system, which aims at determining the system actions to generate appropriate responses to users. To make the system capable of conducting a dialog in a more natural and effective manner, the dialog manager should take into account not only a given user utterance itself, but also the dialog state which represents various conversational situations obtained from the dialog session progress. Dialog state tracking is a sub-task of dialog management that analyzes and maintains this dialog state at each moment. The major obstacle to dialog state tracking is that the inputs to the tracker are likely to be noisy because of the errors produced by automatic speech recognition (ASR) and spoken language understanding (SLU) processes which are required to be performed prior to the tracking.

Thus, many researchers have focused on improving the robustness of dialog state trackers against ASR and SLU errors. The simplest ways to tackle this problem have been based on handcrafted rules mainly on the confidence scores obtained from ASR and SLU modules (Nakano et al., 1999; Wang and Lemon, 2013). However, these approaches have the limitation that building the quality rules manually is expensive and, what is worse, the confidence scores could be unreliable and inconsistent in some cases.

The other direction of dialog state tracking approaches have utilized statistical machine learning techniques to obtain the distribution over a set of hypotheses. Although the most widely studied approaches have been based on generative models (Williams and Young, 2007; Williams, 2010; Young et al., 2010; Thomson and Young, 2010; Gašić and Young, 2011; Raux and Ma, 2011), recently, some researchers have reported that discriminative models (Bohus and Rudnicky, 2006; Lee, 2013; Zilka et al., 2013) achieved comparable, or even better, performances than generative models, especially in the tasks of the first dialog state tracking challenge (DSTC) (Williams et al., 2013).

This work focuses on the second phase of DSTC (Henderson et al., 2014). The major difference of DSTC 2 from the previous challenge is that user goals can be changed even in a single dialog session. This aspect can cause the limitations of the previous approaches assuming the fixed user goal for each session. To solve this dynamic state tracking problem, we propose a sequential labeling approach using linear-chain conditional random fields (CRFs) (Lafferty et al., 2001). This approach aims to improve the performances of the tracker in the case of goal changes by jointly performing prediction and segmentation of dialog states.

2 Problem Definition

A dialog state defined in DSTC 2 consists of the following three components: goals, method, and requested slots.

2.1 Goals Tracking

Goals represent the constraint values which are truly intended by a user at each moment. These values can be represented by using a slot filling

Utterance		Goals	
		Food	Area
S_1	Hello, How may I help you?		
U_1	I need a Persian restaurant in the south part of town.	Persian	South
S_2	What kind of food would you like?		
U_2	Persian.	Persian	South
S_3	I'm sorry but there is no restaurant serving persian food		
U_3	How about Portuguese food?	Portuguese	South
S_4	Peking restaurant is a nice place in the south of town.		
U_4	Is that Portuguese?	Portuguese	South
S_5	Nandos is a nice place in the south of town serving tasty Portuguese food.		
U_5	Alright. Whats the phone number?	Portuguese	South
S_6	The phone number of nandos is 01223 327908 .		
U_6	And the address?	Portuguese	South
S_7	Sure, nandos is on Cambridge Leisure Park Clifton Way.		
U_7	Thank you good bye.		

Figure 1: Examples of user goals tracking on a dialog in the restaurant information domain

(a) Goal chain on the food slot

(b) Method chain

(c) Requested chain on the phone slot

Figure 2: Examples of dialog state tracking as sequential labeling with liner-chain CRFs
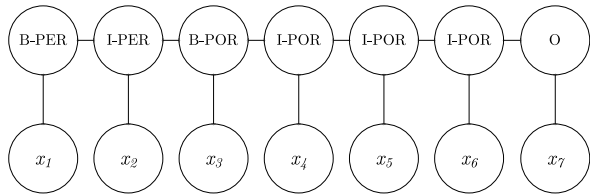
over the following four categories: area, food, name, and price range. Assuming the possible value set for each slot is fixed, this task can be considered to be a problem of finding the distributions over these hypotheses. While the previous challenge aims at identifying a single fixed goal for each session, the models for DSTC 2 should be able to handle goal changes during a session, as shown in Figure 1.
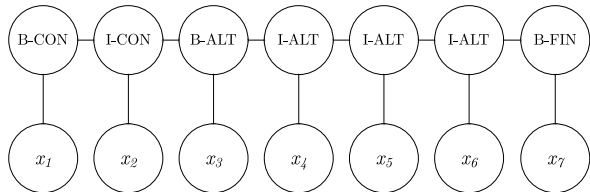
2.2 Method Tracking

Method tracking is performed by classifying the way of requesting information by a user into the following four categories: 'by constraints', 'by alternatives', 'by name', and 'finished'. The probability distribution over these four hypotheses is computed for each turn. For example, a methods sequence {byconstraints, byconstraints, byalternatives, byalternatives, byalternatives, byalternatives, finished} can be obtained for the dialog session in Figure 1.

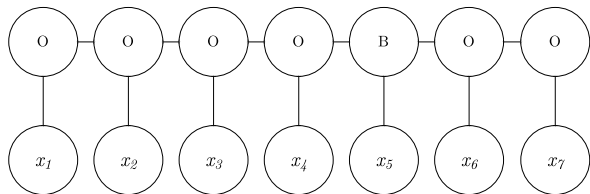## 2.3 Requested Slots Tracking

The other component for dialog state tracking is to specify the slots requested by a user. The tracker should output the binary distributions with the probabilities whether each slot is requested or not. Since the requestable slots are area, food, name, pricerange, addr, phone, postcode, and signature, eight different distributions are obtained at each turn. In the previous example dialog, 'phone' and 'addr' are requested in the 5th and 6th turns respectively.

3 Method

Although some discriminative approaches (Lee, 2013; Zilka et al., 2013; Lee and Eskenazi, 2013; Ren et al., 2013) have successfully applied to the dialog state tracking tasks of DSTC 1 by exploring various features, they have limited ability to perform the DSTC 2 tasks, because the previous models trained based on the features mostly extracted under the assumption that the user goal in a session is unchangeable. To overcome this limitation, we propose a sequential labeling approach using linear-chain CRFs for dynamic dialog state tracking.

3.1 Sequential Labeling of Dialog States

The goal of sequential labeling is to produce the most probable label sequence $\mathbf{y} = \{y_1, \cdots, y_n\}$ of a given input sequence $\mathbf{x} = \{x_1, \cdots, x_n\}$, where n is the length of the input sequence, $x_i \in \mathcal{X}$, $\mathcal{X}$ is the finite set of the input observation, $y_i \in \mathcal{Y}$, and $\mathcal{Y}$ is the set of output labels. The input sequence for dialog state tracking at a given turn t is defined as $\mathbf{x}_t = \{x_1, \cdots, x_t\}$, where x_i denotes the i-th turn in a given dialog session, then a tracker should be able to output a set of label sequences for every sub-task.

For the goals and requested slots tasks, a label sequence is assigned to each target slot, which means the number of output sequences for these sub-tasks are four and eight in total, respectively. On the other hand, only a single label sequence is defined for the method tracking task.

Due to discourse coherences in conversation, the same labels are likely to be located contiguously in a label sequence. To detect the boundaries of these label chunks, the BIO tagging scheme (Ramshaw and Marcus, 1999) is adopted for all the label sequences, which marks beginning of a chunk as 'B', continuing of a chunk as 'I', and outside a chunk as 'O'. Figure 2 shows the examples of label sequences according to this scheme for the input dialog session in Figure 1.

3.2 Linear Chain CRFs

In this work, all the sequential labeling tasks were performed by the tracking models trained using first-order linear-chain CRFs. Linear-chain CRFs are conditional probability distributions over the label sequences $\mathbf{y}$ conditioned on the input sequence $\mathbf{x}$, which are defined as follows:

$$p\left(\mathbf{y}|\mathbf{x}\right) = \frac{1}{Z\left(\mathbf{x}\right)} \prod_{t=1}^{n} \Psi(y_t, y_{t-1}, \mathbf{x}),$$

$$\Psi(y_t, y_{t-1}, \mathbf{x}) = \Psi_1(y_t, \mathbf{x}) \cdot \Psi_2(y_t, y_{t-1}),$$

$$\Psi_1(y_t, \mathbf{x}) = \exp\left(\sum_k \lambda_k f_k(y_t, \mathbf{x})\right),$$

$$\Psi_2(y_t, y_{t-1}) = \exp\left(\sum_k \lambda_k f_k(y_t, y_{t-1})\right),$$

where $Z(\mathbf{x})$ is the normalization function which makes that the distribution sums to 1, $\{f_k\}$ is a set of feature functions for observation and transition, and $\{\lambda_k\}$ is a set of weight parameters which are learnt from data.

3.3 Features

To train the tracking models, a set of feature functions were defined based on the n-best list of user actions obtained from the live SLU results at a given turn and the system actions corresponding to the previous system output.

The most fundamental information to capture a user's intentions can be obtained from the SLU hypotheses with 'inform' action type. For each 'inform' action in the n-best SLU results, a feature

function is defined as follows:

$$f_i(\inf, s, v) = \begin{cases} S_i(\inf, s, v), & \text{if } \inf(s, v) \in \text{UA}_i, \\ 0, & \text{otherwise}, \end{cases}$$

where $S_i(a, s, v)$ is the confidence score of the hypothesis (a, s, v) assigned by SLU for the i-th turn, a is the action type, s is the target slot, v is its value, and UA_i is the n-best list of SLU results.

Similarly, the actions with 'confirm' and 'deny' types derive the corresponding feature functions defined as:

$$f_i(\text{con}, s, v) = \begin{cases} S_i(\text{con}, s, v), & \text{if } \text{con}(s, v) \in \text{UA}_i, \\ 0, & \text{otherwise}, \end{cases}$$

$$f_i(\text{den}, s, v) = \begin{cases} S_i(\text{den}, s, v), & \text{if } \text{den}(s, v) \in \text{UA}_i, \\ 0, & \text{otherwise}. \end{cases}$$

In contrast with the above action types, both 'affirm' and 'negate' don't specify any target slot and value information on the SLU results. The feature functions for these types are defined with (s, v) derived from the previous 'expl-conf' and 'impl-conf' system actions as follows:

$$f_i(\text{aff}, s, v) =$$
$$\begin{cases} \max_j (S_{ij}(\text{aff})), & \text{if expl-conf}(s, v) \in \text{SA}_i, \\ & \text{or impl-conf}(s, v) \in \text{SA}_i \\ 0, & \text{otherwise}, \end{cases}$$

$$f_i(\text{neg}, s, v) =$$
$$\begin{cases} \max_j (S_{ij}(\text{neg})), & \text{if expl-conf}(s, v) \in \text{SA}_i, \\ & \text{or impl-conf}(s, v) \in \text{SA}_i \\ 0, & \text{otherwise}, \end{cases}$$

where SA_i is the system actions at the i-th turn.

The user actions with 'request' and 'reqalts' could be able to play a crucial role to track the requested slots with the following functions:

$$f_i(\text{req}, s) = \begin{cases} S_i(\text{req}, s), & \text{if req}(s) \in \text{UA}_i, \\ 0, & \text{otherwise}, \end{cases}$$

$$f_i(\text{reqalts}, s) = \begin{cases} S_i(\text{reqalts}, s), & \text{if reqalts} \in \text{UA}_i, \\ 0, & \text{otherwise}. \end{cases}$$

The other function is to indicate whether the system is able to provide the information on (s, v) using the 'canthelp' actions as follows:

$$f_i(\text{canthelp}, s, v) = \begin{cases} 1, & \text{if canthelp}(s, v) \in SA_i, \\ 0, & \text{otherwise}. \end{cases}$$

		Dev set			Test set		
		Acc	L2	ROC	Acc	L2	ROC
Joint Goals	ME	0.638	0.551	**0.144**	0.596	0.671	0.036
	CRF	**0.644**	**0.545**	0.103	**0.601**	**0.649**	**0.064**
Method	ME	0.839	0.260	**0.398**	0.877	0.204	**0.397**
	CRF	**0.875**	**0.202**	0.181	**0.904**	**0.155**	0.187
Requested Slots	ME	**0.946**	**0.099**	0.000	0.957	0.081	0.000
	CRF	0.942	0.107	0.000	**0.960**	**0.073**	0.000

Table 1: Comparisons of dialog state tracking performances

4 Experiment

To demonstrate the effectiveness of our proposed sequential labeling approach for dialog state tracking, we performed experiments on the DSTC 2 dataset which consists of 3,235 dialog sessions on restaurant information domain which were collected using Amazon Mechanical Turk. The results of ASR and SLU are annotated for every turn in the dataset, as well as the gold standard annotations are also provided for evaluation. We used this dataset following the original division into training/development/test sets, which have 1,612/506/1,117 sessions, respectively.

Using this dataset, we trained two different types of models: one is based on CRFs for our proposed sequential labeling approach; and the other is a baseline using maximum entropy (ME) that performs the prediction for each individual turn separately from others in a given session. All the models for both approaches were trained on the training set with the same feature functions defined in Section 3.3 using MALLET [1] toolkit.

The trained models were used for predicting goals, method, and requested slots of each turn in the development and test sets, the results of which were then organized into a tracker output object defined as the input format to the evaluation script of DSTC 2. Since we omitted the joint goals distributions in the output, the evaluations on the joint goals were performed on the independent combinations of the slot distributions.

Among the various combinations of evaluation variables listed in the results of the evaluation script, the following three featured metrics were selected to report the performances of the tracker in this paper: Accuracy, L2 norm, and ROC CA 5. All these metrics were computed for the predicted joint goals, method and requested slots.

Table 1 compares the performances of our proposed approach (CRF) and the baseline method (ME) for three sub-tasks on the development and test sets. The results indicate that our proposed sequential labeling approach achieved better performances than the baseline for most cases. Especially, CRF models produced better joint goals and method predictions in terms of accuracy and L2 norm on both development and test sets. For the requested slots task, our proposed approach failed to generate better results than the baseline on the development set. However, this situation was reversed on the test set, which means our proposed approach achieved better performances on all three sub-tasks on the test set in two of the three evaluation metrics.

5 Conclusions

This paper presented a sequential labeling approach for dialog state tracking. This approach aimed to solve the cases of goal changes using linear-chain CRFs. Experimental results show the merits of our proposed approach with the improved performances on all the sub-tasks of DSTC 2 compared to the baseline which doesn't consider sequential aspects.

However, these results are still not enough to be competitive with the other participants in the challenge. One possible reason is that our trackers were trained only on the very basic features in this work. If we discover more advanced features that help to track the proper dialog states, they can raise the overall performances further.

The other direction of our future work is to integrate these dialog state trackers with our existing dialog systems which accept the 1-best results of ASR and SLU as they are, then to see their impacts on the whole system level.

---

[1] http://mallet.cs.umass.edu/

References

Dan Bohus and Alex Rudnicky. 2006. A k-hypotheses+ other belief updating model. In *Proc. of the AAAI Workshop on Statistical and Empirical Methods in Spoken Dialogue Systems*.

Milica Gašić and Steve Young. 2011. Effective handling of dialogue state in the hidden information state pomdp-based dialogue manager. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):4.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the SIGdial 2014 Conference*, Baltimore, U.S.A., June.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Sungjin Lee and Maxine Eskenazi. 2013. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *Proceedings of the SIGDIAL 2013 Conference*, pages 414–422.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451.

Mikio Nakano, Noboru Miyazaki, Jun-ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. 1999. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 200–207.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Antoine Raux and Yi Ma. 2011. Efficient probabilistic tracking of user goal and dialog history for spoken dialog systems. In *Proceedings of INTERSPEECH*, pages 801–804.

Hang Ren, Weiqun Xu, Yan Zhang, and Yonghong Yan. 2013. Dialog state tracking using conditional random fields. In *Proceedings of the SIGDIAL 2013 Conference*, pages 457–461.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Jason D Williams. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5382–5385. IEEE.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Lukas Zilka, David Marek, Matej Korvas, and Filip Jurcicek. 2013. Comparison of bayesian discriminative and generative models for dialogue state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 452–456.