Dialog State Tracking using Conditional Random Fields

Hang Ren, Weiqun Xu, Yan Zhang,Yonghong Yan
The Key Laboratory of Speech Acoustics and Content Understanding
Institute of Acoustics, Chinese Academy of Sciences
21 North 4th Ring West Road, Beijing, China, 100190
{renhang, xuweiqun, zhangyan, yanyonghong}@hccl.ioa.ac.cn

Abstract

This paper presents our approach to dialog state tracking for the Dialog State Tracking Challenge task. In our approach we use discriminative general structured conditional random fields, instead of traditional generative directed graphic models, to incorporate arbitrary overlapping features. Our approach outperforms the simple 1-best tracking approach.

1 Introduction

Spoken dialog systems have been widely developed in recent years. However, when dialogs are conducted in noisy environments or the utterance itself is noisy, it is difficult for machines to correctly recognize or understand user utterances. In this paper we present a novel dialog state tracking method, which directly models the joint probability of hypotheses on N-best lists. Experiments are then conducted on the DSTC shared corpus, which provides a common dataset and an evaluation framework

The remainder of this paper is organized as follows. Section 2 reviews relevant studies in dialog state tracking. Section 3 introduces our new approach and presents the model and features we used in detail. Section 4 describes experiment settings and gives the result. Section 5 concludes this paper with a discussion for possible future directions.

2 Previous Work

For the task of dialog state tracking, previous research focused on dynamic Bayesian models (DBN)(Young et al., 2013). User goal, dialog history and other variables are modeled in a graphical model. Usually, Markov assumptions are made and in each turn the dialog state is dependent on the ASR outputs and the dialog state of the previous turn. Dependency on other features, such as system action, dialog history could be assumed as long as their likelihood is modeled. For a POMDP-based dialog model, the state update rule is as follows:

$$b_{t+1}(s_{t+1}) = \eta P(o_{t+1}|s_{t+1}, a_t) \sum_{s_t} P(s_{t+1}|s_t, a_t) b_t(s_t) \quad (1)$$

where $b_t(s_t)$ is the belief state at time t, o_{t+1} is the observation at time $t+1$, a_t is the machine action. Thus the dialog states are estimated incrementally turn by turn.

Since each node has hundreds, or even thousands, of possible assignments, approximation is necessary to make efficient computation possible. In POMDP-based dialog systems, two common approaches are adopted (Young et al., 2013), i.e., N-best approximation and domain factorization.

In the N-best approach, the probability distribution of user goals are approximated using N-best list. The hidden information state (HIS) model (Young et al., 2010) makes a further simplification that similar user goals are grouped into a single entity called *partition*, inside which all user goals are assigned the same probabilities. The Bayesian update of dialog state (BUDS) model (Thomson and Young, 2010) is a representative of the second approach and adopts a different approximation strategy, where each node is further divided into sub-nodes for different domain concepts and independence assumptions of sub-nodes across concepts are made. Recent studies have suggested that a discriminative model may yield better performance than a generative one (Bohus and Rudnicky, 2006). In a discriminative model, the *emission* part of the state update rule is modeled discriminatively. Possible flawed assumptions in a completely generative models could be mitigated

Proceedings of the SIGDIAL 2013 Conference, pages 457–461,
Metz, France, 22-24 August 2013.

in this way, such as the approximation of observation probability using SLU scores (Williams, 2012a; Williams, 2012b).

3 Proposed Method

3.1 Discriminative State Tracking Model

Most previous methods model the distribution of user goals for each turn explicitly, which can lead to high computation cost. In our work, the joint probability of all items on the N-best lists from SLU is modeled directly and the state tracking result is generated at a post-processing stage. Thus the state tracking problem is converted into a labeling task as is shown in equation 2, which involves modeling the joint probability of the N-best hypotheses.

$$b_t(s_t) = P(H_{1,1}, H_{1,2}, ..., H_{t,m-1}, H_{t,m}) \quad (2)$$

where $H_{t,m}$ is a binary variable indicating the truthfulness of the m-th hypothesis at turn t.

For each turn, the model takes into account all the slots on the N-best lists from the first turn up to the current one, and those slots predicted to be true are added to the dialog state. The graphical model is illustrated in figure 1. To predict dialog state at turn t, the N-best items from turn 1 to t are all considered. Hypotheses assigned true labels are included in the dialog state. Compared to the DBN approach, the dialog states are built 'jointly'. This approach is reasonable because what the tracker generates is just some combinations of all N-best lists in a session, and there is no point guessing beyond SLU outputs. We leverage general structured Conditional Random Fields (CRFs) to model the probabilities of the N-best items, where *factors* are used to strengthen local dependency. Since CRF is a discriminative model, arbitrary overlapping features can be added, which is commonly considered as an advantage over generative models.

3.2 Conditional Random Fields

CRF is first introduced to address the problem of *label bias* in sequence prediction (Lafferty et al., 2001). Linear-chain CRFs are widely used to solve common sequence labeling problem in natural language processing. General structured CRF has also been reported to be successful in various tasks (Sutton and McCallum, 2012).

In general structured CRF, *factor templates* are utilized to specify both model structure and *parameter tying* (Sutton and McCallum, 2012). Factors are partitioned into a series of templates, and factors inside each template share the same parameters.

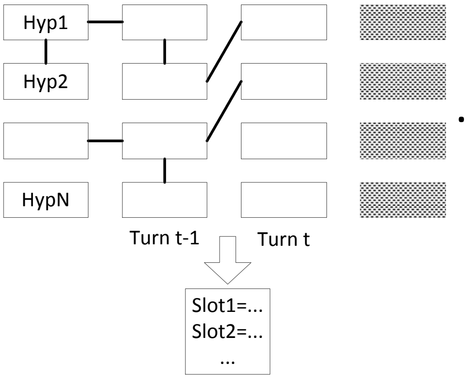

Figure 1: Dialog state update using CRFs, where the 8 rectangles above denote N-best hypotheses for each turn, and the box below represents the dialog state up to the current turn. Connections between rectangles denote 'Label-Label' factors. 'Label-Observation' factors are not shown for simplicity.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x_c}, \mathbf{y_c}; \theta_p), \quad (3)$$

where $\mathcal{C}$ is the set of factor templates and $\mathbf{x}, \mathbf{y}$ are inputs and labels respectively. Template factors are written as

$$\Psi_c(\mathbf{x_c}, \mathbf{y_c}; \theta_p) = \exp \sum_{k=1}^{K(p)} \theta_{pk} f_{pk}\left(\mathbf{x_c}, \mathbf{y_c}\right) \quad (4)$$

and $Z(\mathbf{x})$ is the normalizing function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x_c}, \mathbf{y_c}; \theta_p) \quad (5)$$

In the experiment we use Factorie[1] to define and train the model.

3.3 Model Structure and Features

In the model, slots in every N-best item up to the current turn are represented as binary variables. For simplification of data structure, each slot in a single N-best item is extracted and represented using different label variables, with the same *rank* indicating their

[1] Available from https://github.com/factorie/factorie.

original places in the N-best list. For example, the item `slots: [from: Pittsburgh, data: Tuesday], score: 0.85, rank: 2,` is converted to two slots: `slots: [from: Pittsburgh], score: 0.85, rank: 2` and `slots: [date: Tuesday], score: 0.85, rank: 2`. *Label-label* connections are specified using factor templates between slot pairs, and *Label-observation* templates are used to add slot-wise features. Without label-label connection the model is reduced to a maximum entropy model, and with more connections added, the graph tends to have loopy structures.

Two classes of feature sets (templates) in the experiment are defined as follows.

(1) Label-Label factor templates are used to strengthen the bond between certain slots.

Pairwise-slots of the same rank This template is built for pairs of slots in a turn with the same rank to *bind* their boolean assignment. To avoid creating too many loops and make inference efficient, the factors are added in such an order that the slots involved in a single turn are linked in a linear way.

Pairwise-slots with identical value Slots with identical value may appear in the N-best list for multiple times. Besides, user can mention the same slot in different turns, making these slots more reliable. Similar ordering mechanism is utilized to avoid redundant loops.

(2) Label-observation templates are used to add features for the identification of the truthfulness of slots.

SLU score and rank of slot The score generated by the ASR and SLU components is a direct indicator of the correctness degree of slots. However, a slot's true *reliability* is not necessarily linear with its score. The relationship is quite different for various ASR and SLU algorithms, and scores produced by some ASR are not valid probabilities. As we adopt a data-driven approach, we are able to learn this relationship from data. In addition to the SLU score, the slot rank is also added to the feature set.

Dialog history (grounding information) In most spoken dialog systems, explicit and implicit groundings are adapted to indicate the correctness of the system belief. This information is useful to determine the correctness of slots. The grounding information includes grounding type (implicit or explicit grounding), user reply (negation or confirmation) and corresponding SLU scores.

Count of slots with identical value As previously mentioned, slots with identical values can appear several times and slots with more frequent occurrences are more likely to be correct.

Domain-specific features Slots for some domain concepts often have values with specific forms. For example, in the DSTC data sets, the route slots are usually filled with values like '61d', '35b', and SLU often generates noisy outputs like '6d', '3d'. Thus the lexical form is a very useful feature.

Baseline Tracker The simple and fast 1-best tracking algorithm is used as the baseline tracker and exhibits a satisfying performance. Thus the tracking result is added as an additional feature. This indicates the possibility of combining tracking outputs from different algorithms in this discriminative model, which may improve the overall tracking performance.

4 Experiment

4.1 Task and Data

The Dialog State Tracking Challenge (DSTC)[2] aims at evaluating dialog state tracking algorithms on shared real-user dialog corpus. In each dialog session, ASR and SLU results are annotated, making it possible to conduct direct comparison among various algorithms. For further details, please refer to the DSTC handbook (Williams et al., 2013b).

4.2 Corpus Preprocessing

The ASR and SLU components can generate many noisy hypotheses which are completely wrong, rendering the dialog corpus seriously imbalanced at the level of slots (there are more wrong slots than true slots). We use resampling to prevent

[2] `http://research.microsoft.com/en-us/events/dstc/`

the model from biasing towards making negative judgements. Before training, the total number of correct slots in a set is counted, and equal number of wrong slots are sampled from the subset of all the wrong slots. These chosen negative slots plus all the positive slots together constitute the effective slot set for model training, with remaining slots fixed to their true value and regarded as observed variables. To make full use of the dialog corpus, this process is repeated for eight times, producing a bigger and balanced corpus.

4.3 Model Training

In the training phase, the log-likelihood function is optimized using the LBFGS method with L2-regularization. Loopy belief propagation is used as the inference routine. It can be shown that for factor graphs without loops, the marginal probabilities produced by loopy belief propagation are exact. Model selection is done according to the log-likelihood on the development set.

4.4 Predicting and Tracking

For each dialog session, the predicted slot labels are mapped to tracking results. To produce a N-best list of tracking results, the top N configurations of slots and corresponding probability scores are generated. Gibbs sampling is adopted. The sampling is repeated for 1000 times in each corpus, after each sampling the configuration of slots is mapped to certain tracking state. More efficient inference routines, such as M-best belief propagation (Yanover and Weiss, 2004), could be utilized, which would be suitable for practical real-time application.

4.5 Results

After tracker outputs are generated based on the sampling results, they are scored using evaluation tools provided by the DSTC organizers. Several metrics are evaluated, including precisions, ROC performance, etc. Individual and joint slots are scored respectively. And different schedules are used, which indicats the turns included for evaluation. Partial results are shown in table 1. A systematic analysis by the organizers is in the DSTC overview paper (Williams et al., 2013a). The complete challenge results can be found on DSTC website. On the test sets of test1, test2 and test3, the CRF-based model achieves better performance than the simple baseline in most metrics. However, on test4, the performance degrades seriously when there is a mismatch between training data and test data, since test4 is produced by a different group and does not match the training set. It shows that the CRF-based model is very flexible and is able to learn the properties of ASR and SLU, thus adapting to a specific system. But it has a tendency of overfitting .

	Test1		**Test4**	
Metric	CRF	BASE	CRF	BASE
ACC	**0.987**	0.983	0.960	**0.986**
L2	**0.020**	0.021	0.046	**0.017**
MRR	**0.990**	0.988	0.980	**0.990**
CA05	**0.987**	0.983	0.960	**0.986**
EER	**0.015**	0.983	0.021	**0.012**

Table 1: Results of slot 'Date' on Test1 and Test4 (schedule1 is used). The tracker used on Test4 is trained on Test3. Details of the metrics can be found in DSTC handbook(Williams et al., 2013b)

5 Conclusions and Future Directions

We proposed a CRF-based discriminative approach for dialog state tracking. Preliminary results show that it achieves better performance than the 1-best baseline tracker in most metrics when the training set and testing set match. This indicates the feasibility of our approach which directly models joint probabilities of the N-best items.

In the future, we will focus on the following possible directions to improve the performance. Firstly, we will enrich the feature set and add more domain-related features. Secondly, interactions of slots between dialog turns are not well modeled currently. This problem can be alleviated by tuning graph structures, which deservers further studies. Moreover, it is challenging to use online training methods, so that the performance could be improved incrementally when more training samples are available.

6 Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 10925419, 90920302, 61072124, 11074275, 11161140319, 91120001), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Nos. XDA06030100, XDA06030500), the National 863 Program (No. 2012AA012503) and the CAS Priority Deployment Project (No. KGZD-EW-103-2).

References

Dan Bohus and Alex Rudnicky. 2006. A "k hypotheses + other" belief updating model. In *Proceedings of the 2006 AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, pages 13–18, Menlo Park, California. The AAAI Press.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Charles A. Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

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

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013a. The dialog state tracking challenge. In *Proceedings of the 14th SIGdial workshop on Discourse and Dialogue*.

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013b. Dialog state tracking challenge handbook. Available from `http://research.microsoft.com/apps/pubs/?id=169024`.

Jason D. Williams. 2012a. Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *Selected Topics in Signal Processing, IEEE Journal of*, 6(8):959 –970.

Jason D. Williams. 2012b. A critical analysis of two statistical spoken dialog systems in public use. In *SLT*, pages 55–60. IEEE.

Chen Yanover and Yair Weiss. 2004. Finding the m most probable configurations using loopy belief propagation. *Advances in Neural Information Processing Systems*, 16:289–296.

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 and Language*, 24(2):150–174.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.