Structured Discriminative Model For Dialog State Tracking

Sungjin Lee
Language Technologies Institute,
Carnegie Mellon University,
Pittsburgh, Pennsylvania, USA
sungjin.lee@cs.cmu.edu

Abstract

Many dialog state tracking algorithms have been limited to generative modeling due to the influence of the Partially Observable Markov Decision Process framework. Recent analyses, however, raised fundamental questions on the effectiveness of the generative formulation. In this paper, we present a structured discriminative model for dialog state tracking as an alternative. Unlike generative models, the proposed method affords the incorporation of features without having to consider dependencies between observations. It also provides a flexible mechanism for imposing relational constraints. To verify the effectiveness of the proposed method, we applied it to the Let's Go domain (Raux et al., 2005). The results show that the proposed model is superior to the baseline and generative model-based systems in accuracy, discrimination, and robustness to mismatches between training and test datasets.

1 Introduction

With the recent remarkable growth of speech-enabled applications, dialog state tracking has become a critical component not only for typical telephone-based spoken dialog systems but also for multi-modal dialog systems on mobile devices and in automobiles. With present *Automatic Speech Recognition* (ASR) and *Spoken Language Understanding* errors, it is impossible to directly observe the true user goal and action. It is crucial, therefore, to accurately estimate the true dialog state from erroneous observations as a dialog unfolds.

Since the *Partially Observable Markov Decision Process* (POMDP) framework has offered a well-founded theory for both state tracking and decision making, most earlier studies adopted generative temporal models, the typical way to formulate belief state updates for POMDP-based systems (Williams and Young, 2007). Several approximate methods have also emerged to tackle the vast complexity of representing and maintaining belief states, e.g., partition-based approaches (Gasic and Young, 2011; Lee and Eskenazi, 2012a; Williams, 2010; Young et al., 2010) and Bayesian network (BN)-based methods (Raux and Ma, 2011; Thomson and Young, 2010).

To verify the effectiveness of these techniques, some were deployed in a real user system for the Spoken Dialog Challenge (Black et al., 2010). The results demonstrated that the use of statistical approaches helps estimate the true dialog state and achieves increased robustness to ASR errors (Thomson et al., 2010b; Lee and Eskenazi 2012b; Williams, 2011; Williams, 2012). However, further analysis also raised several fundamental questions about the formulation of the belief update as a generative temporal model: limitation in modeling correlations between observations in different time slices; and the insensitive discrimination between true and false dialog states (Williams, 2012). There are more potential downsides of generative models, which will be discussed in detail in Section 2.

On the other hand, natural language processing, computer vision and other machine learning research areas have increasingly profited from discriminative approaches. Discriminative approaches directly model the class posteriors, allowing them to incorporate a rich set of features without worrying about their dependencies on one another. This could result in a deficient probability distribution with generative models (Sutton and McCallum, 2006).

Proceedings of the SIGDIAL 2013 Conference, pages 442–451,
Metz, France, 22-24 August 2013.

The aim of this paper is to describe a first attempt to adopt a structured discriminative model for dialog state tracking. To handle nonlinearity of confidence score and variable cardinality of the possible values of output variables, the traditional approaches applied to other tasks have been modified.

To verify the effectiveness of the proposed method, we applied it to the Let's Go[1] domain (Raux et al., 2005). The proposed model was compared with its unstructured version without relational constraints, the baseline system which always takes the top ASR hypothesis in the entire dialog, and finally the AT&T Statistical Dialog Toolkit[2] (ASDT) which is one of the state-of-the-art generative model-based systems.

This paper is structured as follows. Section 2 describes previous research and the novelty of our approach. Section 3 elaborates on our proposed structured discriminative approach. Section 4 explains the experimental setup. Section 5 presents and discusses the results. Finally, Section 6 concludes with a brief summary and suggestions for future research.

2 Background and Related Work

A statistical dialog system needs to update its dialog state when taking the action a_s and observing o. Since the POMDP framework assumes the Markovian property between states, updating a belief state involves only the previous belief state, the system action, and the current observation:

$$b'(s') = k \cdot P(o'|s') \sum_{s \in S} P(s'|s, a_s) b(s) \quad (1)$$

where $b(\cdot)$ denotes the probability distribution over states s, $P(o|s)$ the likelihood of o given the state s, $P(s'|s, a_s)$ the state transition probability, and k is a normalizing constant.

In practice, however, belief state updates (Equation 1) in many domains are often computationally intractable due to the tremendously large size of the belief state space. In order to reduce the complexity of the belief states, the following belief state factorization has been commonly applied to the belief update procedure (Williams et al., 2005):

$$b'(g', a'_u, h') \propto \quad (2)$$

$$\underbrace{P(o'|a'_u)}_{observation\ model} \cdot \underbrace{P(a'_u|g', a'_s)}_{user\ action\ model} \cdot$$

$$\sum_h \underbrace{P(h'|h, a'_u, a'_s)}_{history\ model} \cdot \sum_g \underbrace{P(g'|g, a'_s)}_{goal\ model} \cdot \sum_{a_u} b(g, a_u, h)$$

where g, h, a_u, represents the user goal, the dialog history, and the user action, respectively.

Partition-based approaches (Gasic and Young, 2011; Lee and Eskenazi, 2012; Williams, 2010; Young et al., 2010) attempt to group user goals into a small number of partitions and split a partition only when this distinction is required by observations. This property endows it with the high scalability that is suitable for fairly complex domains. In partition-based approaches, the *goal model* in Equation 2 is further approximated as follows:

$$\sum_g P(g'|g, a'_s) = \sum_p P(p'|p) \quad (3)$$

where p is a partition from the current turn. One of the flaws of the partition-based approaches is that when one defines a partition to be a Cartesian product of subsets of possible values of multiple concepts, it will be difficult to adopt sophisticated prior distributions over partitions. That may lead to either employing very simple priors such as uniform distribution or maintaining partition structures separately for each concept. This is one of the main reasons that the previous partition-based approaches could not incorporate probabilistic or soft relational constraints into the models.

To allow for relational constraints and alleviate the complexity problem at the same time, *Dynamic Bayesian Networks* (DBN) with more detailed structures for the user goal have also been developed (Thomson and Young, 2010). Nevertheless, there is still a limitation on the types of constraints they can afford. Since DBN is a directed network, it is not quite suitable for specifying undirected constraints. For example, in the Let's Go domain, users can say the same name for the arrival place as the departure place if they are distracted, missing the prompt for the arrival place and so repeating themselves with the departure place. It is also possible for some place names with similar pronunciations to be recognized as the same (e.g. *Forbes* and *Forward*). The system can, in this

[1] In this task, users call the spoken dialog system to
[2] http://www2.research.att.com/sw/tools/asdt/

case, use the constraint that the departure and arrival places may not be identical.

Another drawback of both approaches is that it is hard to incorporate a rich set of observation features, which are often partly dependent on each other. One can create a feature which reflects ASR error correlations between observations in different time slices. For example, a hypothesis that repeats with low confidence scores is likely to be a manifestation of ASR error correlations. Thus, the highest confidence score that a hypothesis has attained so far could be a useful feature in preventing repeated incorrect hypotheses from defeating the correct hypothesis (which had a higher score but was only seen once). Another useful feature could be the distribution of confidence scores that a hypothesis has attained thus far, since it may not have the same effect as having a single observation with the total score due to the potential nonlinearity of confidence scores. There are many other potentially useful features. The entire list of features is found in Section 3.2.

Dynamic Probabilistic Ontology Trees (Raux and Ma, 2011) is another method based upon DBN which does not impose explicit temporal structures. Since it does not impose temporal structures, it is more flexible in considering multiple observations together. However, it is still difficult to capture co-dependent features, which are exemplified above, without introducing probabilistic deficiency due to its generative foundation (Appendix E). Moreover, the quality of the confidence score will be critical to all generative models up to that point since they do not usually try to handle potential nonlinearity in confidence scores.

As far as discriminative models are concerned, the *Maximum Entropy* (MaxEnt) model has been applied (Bohus and Rudnicky, 2006). But the model is limited to a set of separate models for each concept, not incorporating relational dependencies. Also, it is restricted to maintain only top K-best hypotheses where K is a predefined parameter, resulting in potential degradation of performance and difficulties in extending it to structured models. In Section 3, our structured discriminative model is described. It is designed to take into consideration the aforementioned limitations of generative models and the previous discriminative approach.

3 Structured Discriminative Model

Unlike generative models, discriminative models directly model the class posterior given the observations. *Maximum Entropy* is one of most powerful undirected graphical models (Appendix A). But for some tasks that predict structured outputs, e.g. a dialog state, MaxEnt becomes impractical as the number of possible outputs astronomically grows. For example, in the Lets Go domain, the size of possible joint output configurations is around 10^{17}. To address this problem, *Conditional Random Field* (CRF) was introduced which allows dependencies between output variables to be incorporated into the statistical model (Appendix B).

3.1 Model Structure for Dialog State Tracking

We now describe our model structure for dialog state tracking in detail using the Let's Go domain as a running example. The graphical representation of the model is shown in Fig. 1. The global output nodes for each concept (clear nodes in Fig. 1) are unlike other temporal models, where a set of output nodes are newly introduced for each time slice. Instead, as a dialog proceeds, a set of new observations $\boldsymbol{o}_*^t$ (shaded nodes in Fig. 1) are continuously attached to the model structure and the feature

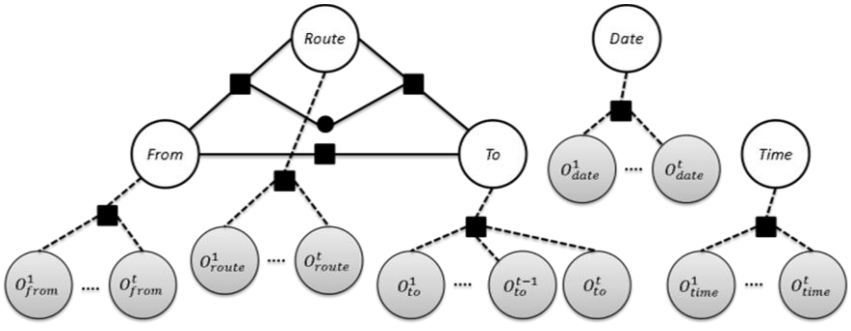


Figure 1: Factor graph representing the structured discriminative model in the Let's Go domain. The shaded nodes show observed random variables. The smaller solid node is the deterministic parameters and explicitly represents parameter sharing between two associated factors.

functions are responsible for producing fixed length feature vectors. The sequence of observations includes not only ASR N-best lists but also system actions from the beginning of the dialog to the current time slice t. Any output node can be freely connected to any other to impose desirable constraints between them whether or not the connections form a loop (solid lines in Fig. 1).

In practice, models rely extensively on parameter tying, e.g., transition parameters in a *Hidden Markov Model*. One specific example of relational constraints and parameter tying naturally arises in the Let's Go domain: the feature function which indicates whether a place is valid on a given route could use the same weights for both departure and arrival places (the solid node and the associated factor nodes in Fig. 1). Parameter tying is also implicitly taking place. This is crucial for robust estimation of the model parameters in spite of data sparseness. Some concepts such as *from* and *to* can have about 10^4 values but most of them are not seen in the training corpus. Thus we aggregate several feature functions which differ only by output labels into one common feature function so that they can gather their statistics together. For example, we can aggregate the observation feature functions (dotted lines in Fig. 1) associated with each output label except for *None* (Section 3.2). Here, *None* is a special value to indicate that the true hypothesis has not yet appeared in the ASR N-best lists. Since there are generally a large number of values for each concept, the probability of the true hypothesis will be very small unless the true hypothesis appears on the N-best lists. Thus we can make inferences on the model very quickly by focusing only on the observed hypotheses at the cost of little performance degradation. Additionally, the feature function aggregation allows for the entire observed hypotheses to be incorporated without being limited to only the pre-defined number of hypotheses.

3.2 Model Features

In this section, we describe the model features which are central to the performance of discriminative models. Features can be broadly split into observation features and relational features. To facilitate readers' understanding an example of feature extraction is illustrated in Fig. 2.

One of the most fundamental features for dialog state tracking should exploit the confidence scores assigned to an informed hypothesis. The simplest form could be direct use of confidence scores. But often pre-trained confidence measures fail to match the empirical distribution of a given dialog domain (Lee and Eskenazi, 2012; Thomson et al. 2010a). Also the distribution of confidence scores that a hypothesis has attained so far may not have the same effect as the total score of the confidence scores (e.g., in Fig. 2, two observations for 61C with confidence score 0.3 vs. 0.6 which is the sum of the scores). Thus we create a feature function that divides the range of confidence scores into bins and returns the frequency of observations that fall into the corresponding bin:

$$inform_k(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ bin\_freq(k, CS_{inf}(y, \boldsymbol{x}_1^t)) \\ otherwise,\ 0 \end{cases} \quad (4)$$

where $CS_{inf}(\cdot)$ returns the set of confidence scores whose action informs y in the sequence of observations $\boldsymbol{x}_1^t$. $bin\_freq(k,\cdot)$ computes the frequency of observations that fall into the k^{th} bin.

There are two types of grounding actions which are popular in spoken dialog systems, i.e., implicit and explicit confirmation. To leverage affirmative or negative responses to such system acts, the following feature functions are introduced in a similar fashion as the $inform$ feature function:

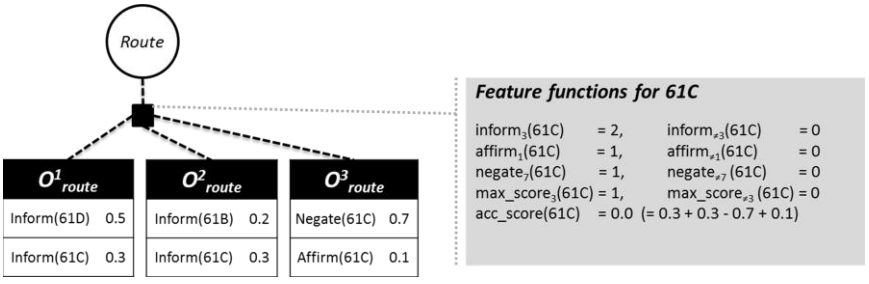


Figure 2: A simplified example of feature extraction for the route concept. It shows the values that each feature will have when three consecutive user inputs are given.

$$affirm_k(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ bin\_freq(k, CS_{aff}(y, \boldsymbol{x}_1^t)) \\ otherwise,\ 0 \end{cases} \quad (5)$$

$$negate_k(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ bin\_freq(k, CS_{neg}(y, \boldsymbol{x}_1^t)) \\ otherwise,\ 0 \end{cases} \quad (6)$$

where $CS_{aff}(\cdot)$ / $CS_{neg}(\cdot)$ returns the set of confidence scores whose associated action affirms / negates y in the sequence of observations $\boldsymbol{x}_1^t$.

$$impl\_affirm(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ I_{impl\_aff}(y, \boldsymbol{x}_1^t) \\ otherwise,\ 0 \end{cases} \quad (7)$$

where $I_{impl\_aff}(\cdot)$ indicates whether or not the user has negated the system's implicit confirmation in the sequence of observations $\boldsymbol{x}_1^t$.

Another interesting feature function is the so-called baseline feature which exploits the output of a baseline system. The following feature function emulates the output of the baseline system which always selects the top ASR hypothesis for the entire dialog:

$$max\_score_k(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ bin(k, MAX\_CS_{inf}(y, \boldsymbol{x}_1^t)) \\ otherwise,\ 0 \end{cases} \quad (8)$$

where $MAX\_CS_{inf}(\cdot)$ returns the maximum confidence score whose action informs y in the sequence of observations $\boldsymbol{x}_1^t$. $bin(k,\cdot)$ indicates whether or not the maximum score falls into the k^{th} bin.

Yet another feature function of this kind is the accumulated score which adds up all confidence scores associated with *inform* and *affirm* and subtracts the ones with *negation*:

$$acc\_score(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None', & \sum CS_{inf}(y, \boldsymbol{x}_1^t) \\ & +\sum CS_{aff}(y, \boldsymbol{x}_1^t) \\ & -\sum CS_{neg}(y, \boldsymbol{x}_1^t) \\ otherwise, & 0 \end{cases} \quad (9)$$

Note that such feature functions as $max\_score(\cdot)$ and $acc\_score(\cdot)$ are not independent of the others defined previously, which may cause generative models to produce deficient probability distributions (Appendix E).

It is known that prior information can boost the performance (Williams, 2012) if the prior is well-estimated. One of advantages of generative models is that they provide a natural mechanism to incorporate a prior. Discriminative models also can exploit a prior by introducing additional feature functions:

$$prior_k(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ bin(k, prior\_frac(y)) \\ otherwise,\ 0 \end{cases} \quad (10)$$

where $prior\_frac(y)$ returns the fraction of occurrences of y in the set of true labels.

If the system cannot process a certain user request, it is highly likely that the user change his/her goal. The following feature function is designed to take care of such cases:

$$canthelp(y, \boldsymbol{x}_1^t) = \begin{cases} y \neq 'None',\ I_{ooc}(y) \\ otherwise,\ 0 \end{cases} \quad (11)$$

where $I_{ooc}(\cdot)$ indicates whether or not y is out-of-coverage.

As with other log-linear models, we also have feature functions for bias:

$$bias(y, \boldsymbol{x}_1^t) = 1$$
$$bias_{none}(y, \boldsymbol{x}_1^t) = \begin{cases} y = 'None', & 1 \\ otherwise, & 0 \end{cases} \quad (12)$$

Note that we have an additional bias term for *None* to estimate an appropriate weight for it.

Regarding relational constraints, we have created two feature functions. To reflect the presumption that it is likely for the true hypothesis for the place concepts (i.e. *from* and *to*) to be valid on the true hypothesis for the *route* concept, we have:

$$place\_in\_route(p, r) = \begin{cases} p \neq 'None',\ valid(p, r) \\ otherwise,\ 0 \end{cases} \quad (13)$$

where $valid(p, r)$ indicates whether or not the place p is valid on the route r. Another feature function considers the situation where the same place name for both departure and arrival places is given:

$$has\_same\_places(p_f, p_t) = \begin{cases} both\ p_f, p_t \neq 'None' and\ p_f = p_t,\ 1 \\ otherwise, \quad 0 \end{cases} \quad (14)$$

3.3 Inference & Parameter Estimation

One of the common grounding actions of spoken dialog systems is to ask a confirmation question about hypotheses which do not have sufficient marginal beliefs. This makes marginal inference

to be one of the fundamental reasoning tools for dialog state tracking. In treelike graphs, exact marginal probabilities are efficiently computable by using the *Junction Tree* algorithm (Lauritzen and Spiegelhalter, 1988) but in general it is intractable on structured models with loops.

Since it is highly likely to have loopy structures in various domains (e.g. Fig. 1), we need to adopt approximate inference algorithms instead. Note that CRF (Equation 16) is an instance of the exponential family. For the exponential family, it is known that the exact inference can be formulated as an optimization problem (Wainwright and Jordan, 2008). The variational formulation opens the door to various approximate inference methods. Among many possible approximations, we adopt the *Tree Reweighted Belief Propagation* (TRBP) method which convexifies the optimization problem that it guarantees finding the global solution (Appendix C).

On the other hand, joint inference also becomes important for either selecting a hypothesis to confirm or determining the final joint configuration when there exist strong relational dependencies between concepts. Moreover, we would like to find not just the best configuration but rather the top M configurations. Since the number of concept nodes is generally moderate, we approximate the inference by searching for the top M configurations only within the Cartesian product of the top K hypotheses of each concept. For domains with a large number of concepts, one can use more advanced methods, e.g., *Best Max-Marginal First* (Yanover and Weiss, 2004) and *Spanning Tree Inequalities and Partitioning for Enumerating Solutions* (Fromer and Globerson, 2009).

The goal of parameter estimation is to minimize the empirical risk. In this paper, we adopt the negative of the conditional log likelihood (Appendix D). Given the partial derivative (Equation 26), we employ the *Orthant-wise Limited-memory Quasi Newton* optimizer (Andrew and Gao, 2007) for L1 regularization to avoid model overfitting.

4 Experimental Setup

In order to evaluate the proposed method, two variants of the proposed method (discriminative model (DM) and structured discriminative model (SDM)) were compared with the baseline system, which always takes the top ASR hypothesis for the entire dialog and outputs the joint configuration using the highest average score, and the ASDT system as being the state-of-the-art partition-based model (PBM). To train and evaluate the models, two datasets from the Spoken Dialog Challenge 2010 are used: a) AT&T system (Williams, 2011), b) Cambridge system (Thomson et. al, 2010b).

	Route	From	To	Date	Time	Joint
Training	378	334	309	33	30	378
Test	379	331	305	54	50	379

(a) Dataset A

	Route	From	To	Date	Time	Joint
Training	94	403	353	18	217	227
Test	99	425	376	18	214	229

(b) Dataset B

Table 1: Counts for each concept represent the number of dialogs which have non-empty utterances for that concept. *From* and *To* concepts add up the counts for their sub-concepts. *Joint* denotes the joint configuration of all concepts.

For discriminative models, we used 10 bins for the feature functions that need to discretize their inputs (Section 3.2). Parameter tying for relational constraints was applied to dataset A but not to dataset B. To make sure that TRBP produces an upper bound on the original entropy, the constants ρ_c were set to be $2/3$ for SDM and 1 for DM (Appendix C). Also the weights for L1 regularization were set to be 10 and 2.5 for the prior features and the other features, respectively. These values were chosen through cross-validation over several values rather than doing a thorough search. For the ASDT system, we modified it to process implicit confirmation and incorporate the prior distribution which was estimated on the training corpus. The prior distribution was smoothed by approximate Good-Turing estimation on the fly when the system encounters an unseen value at run time.

Two aspects of tracker performance were measured at the end of each dialog, i.e. Accuracy and Receiver Operating Characteristic (ROC). Accuracy measures the percent of dialogs where the tracker's top hypothesis is correct. ROC assesses the discrimination of the top hypothesis's score. Note that we considered *None* as being correct if there is no ASR hypothesis corresponding to the transcription. If all turns are evaluated regardless of context, concepts which appear earlier in the dialog will be measured more times than concepts later in the dialog. In order to make comparisons across concepts fair, concepts are only measured when

	All (%)				Joint			
N-best	Baseline	PBM	DM	SDM	Baseline	PBM	DM	SDM
1-best	74.80	77.93	83.65	83.74	53.56	54.62	60.16	60.69
3-best	74.80	84.00	88.83	89.10	53.56	64.38	70.18	70.98
5-best	74.80	84.54	89.54	89.81	53.56	65.70	72.30	73.09
All	74.80	84.81	89.81	**90.26**	53.56	65.96	73.09	**74.67**

(a) Dataset A

	All				Joint			
N-best	Baseline	PBM	DM	SDM	Baseline	PBM	DM	SDM
1-best	65.46	68.73	78.00	**80.12**	11.35	12.23	26.20	**30.13**
3-best	65.46	68.02	78.00	79.51	11.35	11.35	27.51	28.82
5-best	65.46	67.40	77.92	79.15	11.35	11.79	24.89	25.76
All	65.46	66.61	78.00	79.24	11.35	11.79	24.89	25.76

(b) Dataset B

Table 2: Accuracy of the comparative models. The best performaces across the models are marked in bold. *All* means a weighted average accuracy across all concepts.

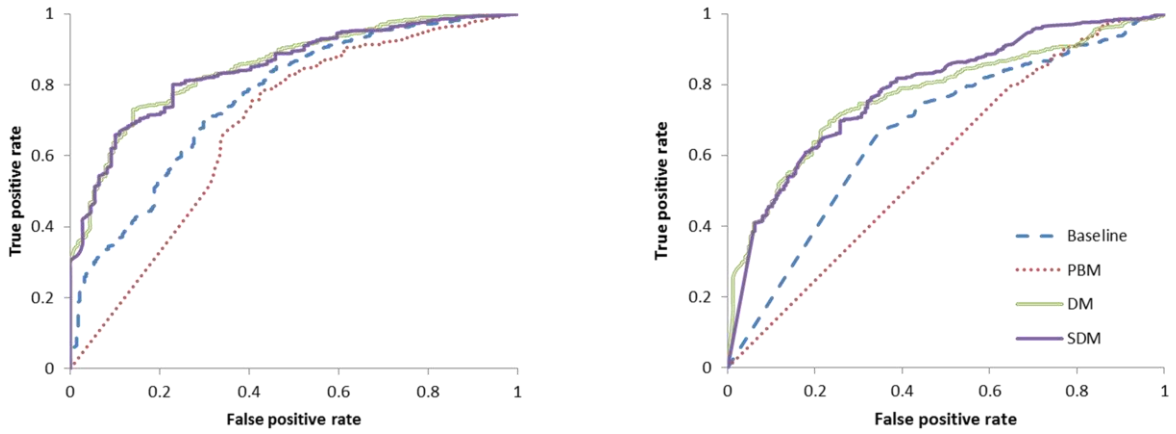
they are in focus. It does not, however, allow for a tracker to receive score for new estimations about concepts that are not in focus. In addition, dialogs with more turns will have a greater effect than dialogs with fewer turns. Therefore we only measure concepts which appear in the dialog at the last turn of the dialog before restart. The statistics of the training and test datasets are summarized in Table 1.

5 Results and Discussion

The results indicate that discriminative methods outperform the baseline and generative method by a large performance gap for both dataset A and B (Table 2). Also, SDM exceeds DM, demonstrating the effectiveness of using relational constraints. Furthermore, the performance of SDM surpasses that of the best system in the *Dialog State Tracking Challenge*[3] (Lee and Eskenazi, 2013). Even though the generative model underperforms discriminative models, it is also shown that dialog state tracking methods in general are effective in improving robustness to ASR errors. Another noteworthy result is that the gains for *Joint* by using discriminative models are much larger than those for *All*. Estimating joint configurations correctly is crucial to eventually satisfy the user's request. This result implies that the proposed model performs evenly well for all concepts and is more robust to the traits of each concept. For example, PBM works relatively poorly for *To* on dataset A. What makes *To* different is that the quality of the ASR hypotheses of the training data is much better than that of test data: the baseline accuracy on the training data is 84.79% while 77.05% on the test data. Even though PBM suffers this mismatch, the discriminative models are doing well without significant differences, implying that the discriminative models achieve robustness by considering not just the confidence score but also several features together.

Since there has been no clear evidence that the use of N-best ASR hypotheses is helpful for dialog state tracking (Williams, 2012), we also report accuracies while varying the number of N-best hypotheses. The results show that the use of N-bests helps boost accuracy across all models on dataset A. However, interestingly it hampers the performance in the case of dataset B. It demonstrates that the utility of N-bests depends on various factors, e.g., the quality of N-bests and dialog policies. The system which yielded dataset A employs implicit and explicit confirmation much more frequently than the system which produced dataset B does. The proposed model trained on dataset A without confirmation features incorporated actually showed a slight degradation in accuracy when using more than 3-bests. This result indicates that we need to take into consideration the type of dialog strategy to determine how many hypotheses to use. Thus, it can be conceivable to dynamically change the range of N-bests according to how a dialog proceeds. That allows the system to reduce processing time when a dialog goes well.

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



(a) Dataset A (b) Dataset B

Figure 3: Weighted average ROC curves across all concepts

The ROC curves assess the discrimination of the top hypothesis' score (Fig. 3). Note that the discriminative models are far better than PBM on both dataset A and B. In fact, PBM turns out to be even worse than the baseline. The better discrimination can give rise to additional values of a tracker. For example, it can reduce unnecessary confirmations for values with sufficiently high belief. Also, it enables a model to adapt to test data in an unsupervised manner by allowing us to set a proper threshold to produce predictive labels.

6 Conclusion

In this paper, we presented the first attempt, to our knowledge, to create a structured discriminative model for dialog state tracking. Unlike generative models, the proposed method allows for the incorporation of various features without worrying about dependencies between observations. It also provides a flexible mechanism to impose relational constraints. The results show that the discriminative models are superior to the generative model in accuracy, discrimination, and robustness to mismatches between training and test datasets. Since we used relatively simple features for this work, there is much room to boost performance through feature engineering. Also, more thorough search for regularization weights can give additional performance gain. Moreover, one can apply different loss functions, e.g., hinge loss to obtain structured support vector machine. In order to further confirm if the performance improvement by the proposed method can be translated to the enhancement of the overall spoken dialog system, we need to deploy and assess it with real users.

Acknowledgments

This work was funded by NSF grant IIS0914927. The opinions expressed in this paper do not necessarily reflect those of NSF. The author would like to thank Maxine Eskenazi for helpful comments and discussion.

References

G. Andrew and J. Gao, 2007. Scalable training of L1-regularized log-linear models. In Proceedings of ICML.

A. Black et al., 2011. Spoken dialog challenge 2010: Comparison of live and control test results. In Proceedings of SIGDIAL.

D. Bohus and A. Rudnicky, 2006. A K hypotheses + other belief updating model. In Proceedings of AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems.

M. Fromer and A. Globerson, 2009. An LP View of the M-best MAP problem. Advances in Neural Information Processing Systems, 22:567-575.

M. Gasic and S. Young, 2011. Effective handling of dialogue state in the hidden information state POMDP-based dialogue manager. ACM Transactions on Speech and Language Processing, 7(3).

S. Lauritzen and D. J. Spiegelhalter, 1988. Local Computation and Probabilities on Graphical Structures and their Applications to Expert Systems. Journal of Royal Statistical Society, 50(2):157–224.

S. Lee and M. Eskenazi, 2012a. Exploiting Machine-Transcribed Dialog Corpus to Improve Multiple Dialog States Tracking Methods. In Proceedings of SIGDIAL, 2012.

S. Lee and M. Eskenazi, 2012b. POMDP-based Let's Go System for Spoken Dialog Challenge. In Proceedings of SLT.

S. Lee and M. Eskenazi, 2013. Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description. Submitted to SIGDIAL, 2013.

A. Raux, B. Langner, D. Bohus, A. W Black, and M. Eskenazi, 2005. Let's Go Public! Taking a Spoken Dialog System to the Real World. In Proceedings of Interspeech.

A. Raux and Y. Ma, 2011. Efficient Probabilistic Tracking of User Goal and Dialog History for Spoken Dialog Systems. In Proceedings of Interspeech.

C. Sutton and A. McCallum, 2006. An Introduction to Conditional Random Fields for Relational Learning. Introduction to Statistical Relational Learning. Cambridge: MIT Press.

B. Thomson and S. Young, 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. Computer Speech & Language, 24(4):562-588.

B. Thomson, F. Jurccek, M. Gasic, S. Keizer, F. Mairesse, K. Yu, S. Young, 2010a. Parameter learning for POMDP spoken dialogue models. In Proceedings of SLT.

B. Thomson, K. Yu, S. Keizer, M. Gasic, F. Jurcicek, F. Mairesse, S. Young, 2010b. Bayesian dialogue system for the Let's Go spoken dialogue challenge. In Proceedings of SLT.

M. Wainwright and M. Jordan, 2008. Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning, 1(1-2):1–305.

J. Williams and S. Young, 2007. Partially observable Markov decision processes for spoken dialog systems. Computer Speech & Language, 21(2):393-422.

J. Williams, 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In Proceedings of ICASSP.

J. Williams, 2011. An Empirical Evaluation of a Statistical Dialog System in Public Use, In Proceedings of SIGDIAL.

J. Williams, 2012. A Critical Analysis of Two Statistical Spoken Dialog Systems in Public Use. In Proceedings of SLT.

C. Yanover and Y. Weiss, 2004. Finding the M Most Probable Configurations Using Loopy Belief Propagation. In Advances in Neural Information Processing Systems 16. MIT Press.

S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson and K. Yu, 2010. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. Computer Speech and Language, 24(2):150–174.

Appendix A. Maximum Entropy

Maximum Entropy directly models the class posterior given the observations:

$$P(y|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\exp(\boldsymbol{\lambda}^T \boldsymbol{f}(y,\boldsymbol{x})) \quad (15)$$

where $Z(\boldsymbol{x})$ is a normalization function, $\boldsymbol{\lambda}$ the model parameters, and $\boldsymbol{f}(y,\boldsymbol{x})$ the vector of feature functions which are key to performance.

Appendix B. Conditional Random Field

Let G be a factor graph over outputs $\boldsymbol{Y}$. Then, if the distribution $P(\boldsymbol{y}|\boldsymbol{x})$ factorizes according to G and $F = \{\Psi_A\}$ is the set of factors in G, the conditional distribution can be written as:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{\Psi_A \in G}\exp(\boldsymbol{\lambda}_A^T \boldsymbol{f}(\boldsymbol{y}_A,\boldsymbol{x}_A)) \quad (16)$$

In practice, models rely extensively on parameter tying. To formalize this, let the factors of G be partitioned to $\mathcal{C} = \{C_1, C_2, \ldots, C_P\}$, where each C_p is a clique template whose parameters are tied. Each clique template is a set of factors which has an associated vector of feature functions $\boldsymbol{f}_p(\boldsymbol{x}_p,\boldsymbol{y}_p)$ and parameters $\boldsymbol{\lambda}_p$. From these it follows (Sutton and McCallum, 2006):

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{C_p \in \mathcal{C}}\prod_{\Psi_c \in C_p}\exp(\boldsymbol{\lambda}_p^T \boldsymbol{f}(\boldsymbol{y}_c,\boldsymbol{x}_c)) \quad (17)$$

where the normalizing function is:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}}\prod_{C_p \in \mathcal{C}}\prod_{\Psi_c \in C_p}\exp(\boldsymbol{\lambda}_p^T \boldsymbol{f}(\boldsymbol{y}_c,\boldsymbol{x}_c)) \quad (18)$$

Appendix C. Tree-reweighted Belief Propagation

Unlike treelike graphs, computing exact marginal probabilities is in general intractable on structured models with loops. Therefore, we need to adopt approximate inference algorithms instead. Note that CRF (Equation 16) is an instance of exponential family:

$$P(\boldsymbol{y};\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{y}) - A(\boldsymbol{\theta})) \quad (19)$$

where $\boldsymbol{\theta}$ is a function of the observations $\boldsymbol{x}$ and the parameters $\boldsymbol{\lambda}$ above, $\boldsymbol{\phi}(\cdot)$ a vector of sufficient statistics consisting of indicator functions for each configuration of each clique and each variable, and $A(\boldsymbol{\theta})$ is the log-partition

function $\log \sum_x \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{y}))$. For exponential family, it is known that the exact inference can be formulated as an optimization problem (Wainwright and Jordan, 2008):

$$A(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\theta}^T \boldsymbol{\mu} + H(\boldsymbol{\mu}) \quad (20)$$

where $\mathcal{M} = \{\boldsymbol{\mu}' | \exists \boldsymbol{\theta}, \boldsymbol{\mu}' = \boldsymbol{\mu}(\boldsymbol{\theta})\}$ is the marginal polytope, $\boldsymbol{\mu}(\boldsymbol{\theta})$ is the mapping from parameters to marginals, and $H(\boldsymbol{\mu})$ is the entropy. Applying Danskin's theorem to Equation 20 yields:

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \frac{dA}{d\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\theta}^T \boldsymbol{\mu} + H(\boldsymbol{\mu}) \quad (21)$$

Thus both the partition function (Equation 20) and marginals (Equation 21) can be computed at once. The variational formulation opens the door to various approximate inference methods: to derive a tractable algorithm, one approximates the log-partition function $\tilde{A}(\boldsymbol{\theta})$ by using a simpler feasible region of $\boldsymbol{\mu}$ and a tractable $H(\boldsymbol{\mu})$. Then the approximate marginals are taken as the exact gradient of $\tilde{A}$. Among many possible approximations, we adopt the *Tree Reweighted Belief Propagation* (TRBP) method which convexifies the optimization problem that it guarantees finding the global solution. TRBP takes the local polytope as a relaxation of the marginal polytope:

$$\mathcal{L} = \{\boldsymbol{\mu} | \sum_{\boldsymbol{y}_{c \setminus i}} \boldsymbol{\mu}(\boldsymbol{y}_c) = \boldsymbol{\mu}(y_i), \sum_{y_i} \boldsymbol{\mu}(y_i) = 1\} \quad (22)$$

where c and i index each clique and output variable, respectively. TRBP approximates the entropy as follows:

$$H(\boldsymbol{\mu}) = \sum_i H(\boldsymbol{\mu}_i) - \sum_c \rho_c \cdot I(\boldsymbol{\mu}_c) \quad (23)$$

where $I(\cdot)$ denotes the mutual information and the constants ρ_c need to be selected so that they generate an upper bound on the original entropy.

Appendix D. Parameter Estimation For Conditional Random Field

The goal of parameter estimation is to minimize the empirical risk:

$$R(\boldsymbol{\lambda}) = \sum_n L(\boldsymbol{\lambda}, \boldsymbol{y}_n, \boldsymbol{x}_n) \quad (24)$$

where there is summation over all training examples. The loss function $L(\boldsymbol{\lambda}, \boldsymbol{y}_n, \boldsymbol{x}_n)$ quantifies the difference between the true and estimated outputs. In this paper, we adopt the negative of the conditional log likelihood:

$$\ell(\boldsymbol{\lambda}) = \sum_{C_p \in \mathcal{C}} \sum_{\Psi_c \in C_p} \boldsymbol{\lambda}_p^T \boldsymbol{f}(\boldsymbol{y}_c, \boldsymbol{x}_c) + \log Z(\boldsymbol{x}) \quad (25)$$

The partial derivative of the log likelihood with respect to a vector of parameters $\boldsymbol{\lambda}_p$ associated with a clique template C_p is:

$$\frac{\partial \ell}{\partial \boldsymbol{\lambda}_p} = \sum_{\Psi_c \in C_p} \boldsymbol{f}_p(\boldsymbol{y}_c, \boldsymbol{x}_c) - \sum_{\Psi_c \in C_p} \sum_{\boldsymbol{y}'_c} \boldsymbol{f}_p(\boldsymbol{y}'_c, \boldsymbol{x}_c) P(\boldsymbol{y}'_c | \boldsymbol{x}_c) \quad (26)$$

Appendix E. Probabilistic Deficiency

To include interdependent features in a generative model, we have two choices: enhance the model to represent dependencies among the inputs, or make independence assumptions. The first approach is often difficult to do while retaining tractability. For example, it is hard to model the dependence between $inform_k$, $affirm_k$, $negate_k$, max_score_k, and acc_score. On the other hand, the second approach can hurt performance by resulting in poor probability estimates. Let's consider the joint probability $p(x_1, \cdots, x_n, y)$ which the generative approach is based on. Because of the independence assumption, the joint probability can be written as $p(y)p(x_1|y) \cdots p(x_n|y)$. For example, let's assume that we observe two hypotheses 61D and 61B with confidence score 0.6 and 0.2, respectively. Then the conditional probabilities can be written as:

$$p(inform_6 = 1, acc\_score = 0.6|61\text{D}) = p(inform_6 = 1|61\text{D}) \cdot p(acc\_score = 0.6|61\text{D})$$

$$p(inform_2 = 1, acc\_score = 0.2|61\text{B}) = p(inform_2 = 1|61\text{B}) \cdot p(acc\_score = 0.2|61\text{B})$$

Since $inform_k = 1$ and $acc\_score = 0.k$ have a strong correlation, their probability estimates should also be positively correlated. To simplify the discussion, now suppose 61B and 61D are equiprobable, $p(61\text{D}) = p(61\text{B})$ and have similar conditional probabilities:

$$p(inform_k = 1|61\text{D}) \sim p(inform_k = 1|61\text{B})$$
$$p(acc\_score = 0.k|61\text{D}) \sim p(acc\_score = 0.k|61\text{B})$$

Then, multiplying those conditional probabilities, $p(inform_k = 1|y) \cdot p(acc\_score = 0.k|y)$, will increase or decrease the confidence of the classifier too much, even though no new evidence has been added.