INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Target-based state and tracking algorithm for spoken dialogue system

Miao Li, Zhiyang He, Ji Wu

Department of Electronic Engineering, Tsinghua University, Beijing, China
miao-li10@mails.tsinghua.edu.cn, {zyhe_ts, wuji_ee}@mail.tsinghua.edu.cn

Abstract

Conventional spoken dialogue systems use frame structure to represent dialogue state. In this paper, we argue that using target distribution to represent dialogue state is much better than using frame structure. Based on the proposed target-based state, two target-based state tracking algorithms are introduced. Experiments in an end-to-end spoken dialogue system with real users are conducted to compare the performance between the target-based state trackers and frame-based state trackers. The experimental results show that the proposed target-based state tracker achieve 97% of dialogue success rate, comparing to 81% of frame-based state tracker, which suggests the advantage of target-based state.

Index Terms: Spoken dialogue system, dialogue management, target-based dialogue state representation, target-based state tracking

1. Introduction

Spoken dialogue systems (SDS) enable human users to acquire information and services through nature language conversations. The dialogue management is a very important module in a modern SDS architecture. During a conversation, the dialogue management module should solve two problems. Firstly, it should maintain a dialogue state to represent user goal at any point in a conversation[1]. Besides, it should determine how to generate a response to the user based on current dialogue state. These two problems often refer as dialogue state tracking [1, 2, 3] and dialogue policy [4, 5, 6, 7] in the literature, which have been widely studied by recent researchers. In fact, before we thinking about these two problems, a more fundamental problem is what data structure we should use to represent a dialogue state.

In most of the task-oriented systems, the dialogue system can be thought as an interface to a back-end database. The system needs to collect enough information from the user to query the database and then offer the proper database entries (or options in some literatures) to the user, such as a restaurant, a hotel or a flight. A frame structure is often used to represent the dialogue state, called as frame-based state hereafter in this paper. The frame structure consists of slots that are filled with the values elicited from the user. The slots and values in the frame structure are related to attributes and values of the domain database.

The work was supported in part by the National Natural Science Funds of China under Grant 61571266, and in part by the Electronic Information Industry Development Fund of China under project The R&D and Industrialization on Information Retrieval System Based on Man-Machine Interaction with Natural Speech.

[1]Generally, a dialogue state may contain not only user goal but also other information such as dialogue history. In this paper, we focus on the user goal part, since user goal is the most important part of a dialogue state.

To handle errors created by the automatic speech recognition (ASR) and spoken language understanding (SLU), the most popular way is to maintain a distribution over dialogue state hypotheses at each turn in the dialogue. While in real world applications each slot can have many possible values, therefore the number of dialogue state hypotheses grows exponentially. In this situation, using frame structure as the representation of dialogue states has some drawbacks. Firstly, many frame hypotheses maintained by the system may not actually exist in the database. Secondly, to tackle the exponential hypotheses, approximate approaches must be used. One example is the hidden information state (HIS) model [5], which maintains an n-best list of state hypotheses. Such a system will get into trouble when the n-best hypotheses list doesn't contain the real user goal. Another example is to factor the frame state into slots, such as the Bayesian update of dialogue state (BUDS) model [6]. But the independent assumptions between slots may not hold in databases of real world applications.

The frame-based state is actually an intermediate representation of user goal, the real user goal is one of the entries (which we call a "target" in the rest of the paper) in the database. A target distribution (which we call target-based state) contains more precise and richer information than approximate frame-based belief state. The target-based state has been used in a probabilistic framework in our previous work [8, 9]. By summarizing the target-based state at each turn of the dialogue, an efficient dialogue strategy is proposed to control the conversation. In this paper, we argue that using target-based state is more reasonable. In order to track the target-based state at each dialogue turn, two target-based state trackers are introduced. Experiments in an end-to-end spoken dialogue system with real users are conducted to compare the target-based state trackers with conventional frame-based state tracker.

The rest of the paper is organized as follows. Section 2 describes the definition and properties of target-based state tracking. Two target-based state trackers are proposed in Section 3. And the experimental results and analysis appear in Section 4. Finally, we conclude the paper in Section 5.

2. Target-Based State Tracking

The frame structure is wildly used in most of the task-oriented SDSs to represent dialogue states. Various state tracking algorithms are proposed based on the frame-based state representation. Equal to the combinatorial numbers of slots and values, the number of potential states is quite large in real world application, which is much larger than the number of entries in the back-end database. Maintaining a distribution over all potential states becomes infeasible. So, many state tracking algorithms maintain marginal distributions instead. In these algorithms, the dialogue states and the SLU outputs are factorized to slots. The marginal distribution of each slots is updated independent-

http://dx.doi.org/10.21437/Interspeech.2016-800

ly and the joint distribution of the frame states is calculated as a product of marginal distributions.

We consider that maintaining a distribution over targets is a better choice for state representation. In most cases, the number of potential states is much larger than targets of the domain database because many combinations do not exist. So maintain a complete distribution over targets without any approximation can be feasible. The target distribution can bring us a clearer picture of the dialogue process. If we don't consider any prior knowledge, the distribution is flat in the beginning of a dialogue. And it keeps going sharper and sharper during the dialogue. When the probability of a target is large enough, the system can offer it to the user. Based on the target distribution, marginal distributions of each slots can be easily obtained. Many existing policy designing and learning approaches based on marginal distributions can also be incorporated.

If the target distribution is used to represent the dialogue state, the task of updating the target distribution at each turn of a dialogue is called Target-Based State Tracking (TBST). The formulation of TBST is as below.

In general, there is a back-end database $\mathbf{D} = \{d_i | i = 1, 2, ..., I\}$ for a task-oriented SDS, where each data entry d_i of the database represent a potential target wanted by a user. Each entry is often associated with a set of slots $\mathbf{S} = \{s_k | k = 1, 2, ..., K\}$, each slot s_k has a set of possible values $V_k = \{v_{k,m} | m = 1, 2, ..., M_k\}$. At each turn t ($t = 1, 2, 3, ...$), a target-based state tracker needs to maintain a distribution P_d^t over all targets based on the SLU output o_t of current turn and the target distribution P_d^{t-1} of previous turn.

$$P_d^t = TBST(o_t, P_d^{t-1}) \tag{1}$$

P_d^0 is the prior target distribution. The prior target distribution can be set as a uniform distribution when there is no prior knowledge. When the SDS collects enough dialogue data, we can reset the prior target distribution based on the collected data. Moreover, we can incorporate a user preference model to build the prior target distribution.

Generally, the SLU output o_t is an n-best list of dialogue act hypotheses A_{hyp_j} with corresponding probabilities p_j for $j = 0, ..., n-1$ and $\sum_{j=0}^{n-1} p_j \leq 1$. Only one dialogue act hypothesis in the n-best list can be the true dialogue act from the user, or none of the hypotheses are the true dialogue act. Each dialogue act hypothesis A_{hyp_j} is given by the dialogue act type A-type$_{hyp_j}$ (such as inform or affirm) and the set of slot bindings X_{hyp_j}.

3. Target-Based State Tracking Algorithms

In this section, we present two target-based state tracking algorithms. Before we go through the details of these two algorithms, let us review some basic mathematics. Let $P(X)$ denote the probability of the occurrence of an event X. There is an evidence E_1 supporting the occurrence of X with probability $P(E_1)$. When we receive the evidence E_1, the probability of the occurrence of X becomes:

$$\begin{aligned} P'(X) &= P(E_1) \cdot 1 + (1 - P(E_1)) \cdot P(X) \\ &= 1 - (1 - P(E_1)) \cdot (1 - P(X)) \end{aligned} \tag{2}$$

If there is an evidence E_2 opposing the occurrence of event X with probability $P(E_2)$. When we receive the evidence E_2, the probability of the occurrence of X becomes:

$$\begin{aligned} P'(X) &= P(E_2) \cdot 0 + (1 - P(E_2)) \cdot P(X) \\ &= (1 - P(E_2)) \cdot P(X) \end{aligned} \tag{3}$$

The Equation (2) and Equation (3) are two domain independent rules and are used to track slot marginal distributions in [10].

Inspired by the intuition of these two equations, we propose our TBST algorithms. For each turn t, the TBST algorithms consider the SLU output o_t as a list of evidences, each evidence e_j corresponds to a dialogue act hypotheses A_{hyp_j} with probability p_j in o_t. The algorithms track the target distribution P_d^t based on the n evidences and the previous distribution P_d^{t-1}. Besides, the algorithms are designed with two principles. First, no slot independent assumption should be made during the tracking process. We use the full slot bindings X_{hyp_j} of a dialogue act hypothesis A_{hyp_j} instead of factoring the slot bindings to slots. Second, the n evidences are observed at the same time, the tracking result should not be affected by the order of evidences' utilization.

Following our first principle, we first map each evidence e_j to a support set of targets S_j. We can get a set of targets S'_j by query the database with slot bindings X_{hyp_j}. If the dialogue act type A-type$_{hyp_j}$ is "inform", then the support set S_j is S'_j. If A-type$_{hyp_j}$ is "deny", then the support set S_j is the absolute complement of S'_j ($S_j = S'^C_j$) [2]. Both of the two TBST algorithms share the same mapping procedure.

3.1. The TBST1 algorithm

The TBST1 algorithm decomposes the evidences to each target, and then updates the probability of each target based on Equation (2) and Equation (3). After mapping each evidence e_j to a support set S_j, a support distribution C_d^t is constructed by reassigning the probability p_j of each evidence to the targets in their support set S_j according to the proportion of the previous target distribution.

$$C_d^t(d_i) = \sum_{j=0}^{n-1} I(d_i \in S_j) \cdot \frac{P_d^{t-1}(d_i)}{\sum_{d_m \in S_j} P_d^{t-1}(d_m)} \cdot p_j \tag{4}$$

where $I()$ is an indicator function. At last, the probability for each target d_i in turn t can be updated by applying Equation (2) and Equation (3) for every target:

$$\begin{aligned} &P_d^t(d_i) = \eta(1 - (1 - P_d^{t-1}(d_i))(1 - C_d^t(d_i))), \quad if\ C_d^t(d_i) > 0; \\ &P_d^t(d_i) = \eta(1 - \sum_{j=0}^{n-1} p_j) P_d^{t-1}(d_i), \qquad\qquad if\ C_d^t(d_i) = 0; \end{aligned} \tag{5}$$

where η is a normalization constant to ensure the sum of the target distribution is 1. For any target d_i who is supported by any evidence from o_t ($C_d^t(d_i) > 0$), we use Equation (2) to update its probability. If a target d_i is not supported by any evidence from o_t ($C_d^t(d_i) = 0$), the probability will decrease based on Equation (3). This tracking algorithm is first presented in [8].

3.2. The TBST2 algorithm

If we look at the first line of Equation (2) and Equation (3), we will find that they are very similar to formulas of expectations. Following this intuition, the TBST2 algorithm is proposed as follows:

$$P_d^t(d_i) = P_d^t(d_i | e_N) \cdot p_N + \sum_{j=0}^{n-1} P_d^t(d_i | e_j) \cdot p_j \tag{6}$$

[2] If A-type$_{hyp_j}$ is "affirm" or "negate", we can easily change this dialogue act to a "inform" or "deny" with last system prompt.

where $P_d^t(d_i|e_j)$ represents the posterior probability when e_j is the only true evidence. $p_N = 1 - \sum_{j=0}^{n-1} p_j$ represents the residual probability of the n-best list. $P_d^t(d_i|e_N)$ represents the posterior probability when all the evidences are false. When e_j is the true evidence, the targets in its support set S_j are truly wanted by the user and the probabilities should follow the proportion of the previous target distribution. So $P_d^t(d_i|e_j)$ can be calculated as follows:

$$\begin{aligned} P_d^t(d_i|e_j) &= \frac{P_d^{t-1}(d_i)}{\sum_{d_m \in S_j} P_d^{t-1}(d_m)}, \quad if\ d_i \in S_j; \\ P_d^t(d_i|e_j) &= 0, \qquad\qquad\qquad\quad if\ d_i \notin S_j; \end{aligned} \tag{7}$$

When none of the evidences in o_t is true, then we have no idea about what target is wanted by the user. The posterior distribution should equal to the previous target distribution:

$$P_d^t(d_i|e_N) = P_d^{t-1}(d_i) \tag{8}$$

Both of the two TBST algorithms are derived from the intuition of Equation (2) and Equation (3). TBST1 applies the intuition in the target grain while TBST2 applies the intuition in the target set grain.

4. Experiments and Result Analysis

4.1. The Song-On-Demand Domain

In this study, the proposed target-based state tracking algorithms are tested in a Song-On-Demand domain. There is a back-end database which consists of 38,117 songs in total. Each song is associated with a set of 12 slots listed in Table 1. In the SoD task, a user try to find a song according to his preferences of these slots, and the system returns the related song based on the information elicited from the user during a dialogue. Although it seems like a simple task, it faces similar challenges in common with other task-oriented spoken dialogue tasks.

Table 1: *The 12 slots of a song in the database.*

ID	Slot	Description	Size
1	Singer	The name of the singer	3010
2	Gender	The gender of the singer	2
3	Region	The region of the singer	19
4	Album	The album of the song	10024
5	Company	The publisher of the song	1184
6	Language	The language of the song	10
7	Lyricist	The lyricist of the song	5633
8	Composer	The composer of the song	5582
9	Live	Live version or not	2
10	Time	Release year of the song	50
11	Style	The style of the song	15
12	Emotion	The emotion of the song	37

These data were primarily collected from the internet. The number of possible values for each slots is shown in the rightmost column of Table 1. The collection of 38,117 songs are published for recent 50 years. There are 3010 distinct singers issuing a total of 10024 albums. This database is used for a real-world application. The database entries and the possible values for slots is large enough to test the scalability of our algorithms.

4.2. Experiments Setup

In this paper, the proposed target-based state tracking algorithms are tested in an end-to-end spoken dialogue system. A large vocabulary continuous speech recognizer [11, 12] is used to transcribe the input speech. Based on the multiple-candidates recognition results, a rule-based SLU module [13] is used to generate the n-best SLU results[3]. Top-1 and Top-5 accuracies are used as the performance metrics for both ASR and SLU module, and the results are shown in Table 2. Since the performances of ASR and SLU modules are different for different slots, so the overall performances have a little difference between different test cases.

To compare with our target-based state tracking algorithms, a frame-based state tracking algorithm is also implemented in the experiments. A simple handcrafted dialogue policy is used to control the dialogue process.

4.2.1. HWU baseline

The HWU tracker [10] is used as the baseline frame-based state tracker. The HWU tracker uses a selection of domain independent rules to update marginal distributions for each slot, and the joint distribution is calculated as a product of marginal distributions.

4.2.2. The Dialogue Policy

Since the main focus of this paper is to demonstrate the advantage of the target-based state representation and state tracking algorithms, the dialogue policy implemented in the experiments is quite simple and straightforward. The policy follows a predefined and fixed order, at each turn of a dialogue, it requests one slot from the user. Two orders are tested in the experiments, $Order1$ is the slot order shown in Table 1, $Order2$ is a descending slot order according to entropies of the marginal distributions of all 12 slots. The marginal distributions are calculated based on the target prior distribution. The target prior distribution is set as a uniform distribution in the experiments.

When the HWU tracker is used as the state tracker, both marginal distribution and the joint distribution can be generated based on the n-best SLU results at each turn. Then the top-m (m=1,3,5) joint state hypotheses are used to query the database to generate a target candidate set. The dialogue process keeps going until one of the following three conditions are met:
1) the candidate set is empty,
2) there is only one song in the candidate set,
3) all slots have been requested by the system.
For condition 1), the system return a "can't help" dialogue act to the user and finish the dialogue, for condition 2) and 3) the system offer a song to the user and finish the dialogue.

When the target-based state tracker is used, the target distribution is maintained by the tracker at each turn. The dialogue termination conditions is a little different:
1) one candidate song was dominant in probability, the top probability exceeds a predefined threshold t,
2) all slots have been requested by the system.

The system return the song with top probability to the user and finish the dialogue. When a dialogue is finished, the top-1 accuracy (whether the offered song is the right song), top-5 accuracy (whether the right song is included in top-5 candidate songs) and dialogue turns are evaluated as the metrics.

[3] n=5 in the entire experiments.

Table 2: *The experimental result for* $Order1$.

Tracker	ASR Accu	SLU Accu	Top 1 Accu	Top 5 Accu	Average Turns
HWU_t1[1]	0.894(0.948)[2]	0.891(0.946)[2]	0.678	0.712	8.975
HWU_t3[1]	0.893(0.942)	0.891(0.940)	0.780	0.831	8.767
HWU_t5[1]	0.891(0.942)	0.891(0.940)	0.831	0.881	9.083
TBST1	0.885(0.934)	0.877(0.932)	**0.933**	**0.983**	10.01
TBST2	0.886(0.934)	0.876(0.932)	0.9	**0.983**	9.317

[1]HWU_tx represents using top-x joint hypotheses to query database.
[2]Prediction accuracy for top 5 output candidates shown in parentheses.

Table 3: *The experimental result for* $Order2$.

Tracker	ASR Accu	SLU Accu	Top 1 Accu	Top 5 Accu	Average Turns
HWU_t1	0.835(0.885)	0.810(0.880)	0.729	0.763	4.333
HWU_t3	0.834(0.888)	0.811(0.876)	0.797	0.831	5.317
HWU_t5	0.839(0.891)	0.820(0.879)	0.814	0.847	6.367
TBST1	0.847(0.9)	0.836(0.894)	**0.967**	**1.0**	7
TBST2	0.842(0.891)	0.822(0.884)	0.917	0.983	6.05

4.3. Experimental Results

6 human subjects are involved in the experiments. Each human subject is assigned a task to get 10 songs. The human subjects are fully cooperated during the conversations. 5 trackers are tested with the 6 human subjects in two requested orders in the experiments. For one requested order, there are 60 test cases for each tracker. The probability threshold t for TBSTs is set as 0.9. The experimental results for different requested orders are shown in Table 2 and Table 3.

Comparing Table 2 with Table 3, we can see the average dialogue turns in $Order2$ are significantly less than the average dialogue turns in $Order1$. The dialogue efficiency can be improved when considering the database information.

Due to the errors of ASR and SLU results, there is no guarantee of the correctness of the top-1 joint hypothesis generated by the HWU tracker. In many test dialogues, the top-1 joint hypothesis went wrong and the dialogue finished because there was no candidate songs for the top-1 joint hypothesis. This leaded to the lowest accuracies and shortest average dialogue turns for "HWU_t1".

In both requested orders, the TBST algorithms have higher dialogue accuracies than the HWU trackers. The TBST1 tracker gets the highest top-1 accuracy and top-5 accuracy, while the average dialogue turn is a little higher than other trackers. The TBST2 tracker gets the second good accuracies while keeping a high dialogue efficiency.

For further comparison between the two TBST tracker, we evaluate the top-1 accuracy and the average dialogue turns in different termination thresholds t (t changes from 0.6 to 0.95). The results are shown in Figure 1. We can see the different properties of the two TBST tracker. TBST1 can reach a high dialogue accuracy while it needs more dialogue turns. TBST2 can reach a good dialogue accuracy in short dialogue turns. Since there is a tradeoff between the dialogue success rate and the dialogue efficiency, TBST1 is more suitable for tasks seeking for higher dialogue success rate while TBST2 is more suitable for tasks seeking for higher dialogue efficiency.

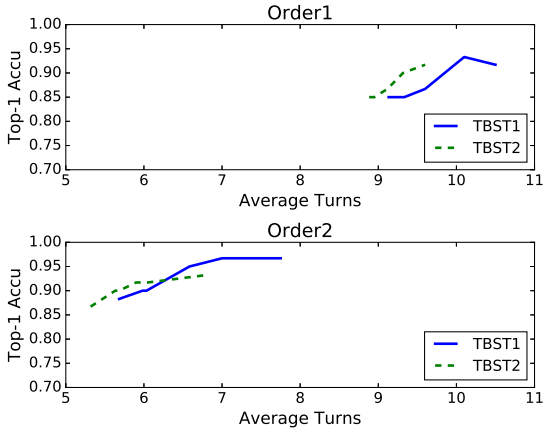

Figure 1: The performance of the two TBST trackers in different termination thresholds.

5. Conclusion and Future work

In this paper, a distribution over targets in a domain database is introduced to represent dialogue state. Two target-based state tracking algorithms are proposed to track the target distribution at each turn of a dialogue. The proposed target-based state trackers are compared with a frame-based state tracker in an end-to-end spoken dialogue system. The experimental results show that the target-based state trackers achieve a higher dialogue success rate than the frame-based state tracker.

We believe that the target-based state representation can easily incorporate the existing techniques in spoken dialogue systems. The target distribution can provide rich features for policy designing or reinforcement learning. When incorporate a user-preference model to the target prior distribution, the dialogue system can provide more personalized services to the user based on the current information presentation techniques. We leave these as our future works.

6. References

[1] J. D. Williams, "Incremental partition recombination for efficient tracking of multiple dialog states," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5382–5385.

[2] B. Thomson, J. Schatzmann, and S. Young, "Bayesian update of dialogue state for robust dialogue systems," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4937–4940.

[3] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference*, 2013, pp. 404–413.

[4] J. D. Williams and S. Young, "Scaling POMDPs for spoken dialog management," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2116–2129, 2007.

[5] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for POMDP-based spoken dialogue management," *Computer Speech & Language*, vol. 24, no. 2, pp. 150–174, 2010.

[6] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech & Language*, vol. 24, no. 4, pp. 562–588, 2010.

[7] M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Gaussian processes for fast policy optimisation of POMDP-based dialogue managers," in *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010, pp. 201–204.

[8] J. Wu, M. Li, and C.-H. Lee, "A Probabilistic Framework for Representing Dialog Systems and Entropy-Based Dialog Management Through Dynamic Stochastic State Evolution," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 2026–2035, Nov 2015.

[9] ——, "An Entropy Minimization Framework for Goal-Driven Dialogue Management," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[10] Z. Wang and O. Lemon, "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*, 2013, pp. 423–432.

[11] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 215–219.

[12] Z. Chen, T. Zhang, and J. Wu, "Subword scheme for keyword search," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 483–488.

[13] W. Ward and S. Issar, "Recent improvements in the CMU spoken language understanding system," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 213–216.