Online Learning of Attributed Bi-Automata for Dialogue Management in Spoken Dialogue Systems

Manex Serras[1], María Inés Torres[2](✉), and Arantza Del Pozo[1]

[1] Vicomtech-IK4 HSLT Department, Donostia, Spain
{mserras,adelpozo}@vicomtech.org
[2] Speech Interactive Research Group, Universidad del País Vasco, UPV/EHU, Leioa, Spain
manes.torres@ehu.es
http://www.vicomtech.org
http://www.ehu.es/en/web/speech-interactive/about-us

Abstract. Online learning of dialogue managers is a desirable but often costly property to obtain. Probabilistic Finite State Bi-Automata (PFSBA) have shown to provide a flexible and adaptive framework to achieve this goal. In this paper, an Attributed PFSBA (A-PSFBA) is implemented and experimentally compared with previous non-attributed PFSBA proposals. Then, a simple yet effective online learning algorithm that adapts the probabilistic structure of the Bi-Automata on the run is presented and evaluated. To this end, the User Model is also represented by an A-PFSBA and the impact of different user behaviors is tested. The proposed approaches are evaluated on the Let's Go corpus, showing significant improvements on the dialogue success rates reported in previous works.

Keywords: Spoken Dialogue Systems · Online learning · Attributed Bi-Automata · Dialogue management

1 Introduction

Spoken Dialogue Systems (SDS) enable people to interact with computers, using spoken language in a natural way [1]. A key task that every SDS has to carry out is controlling the logic structure of the interaction, usually done by the Dialogue Manager (DM). Several approaches have been proposed to model the DM statistically: Bayesian networks [3], Stochastic Finite-State models [4,10], Partially Observable Markov Decision Processes [8] and Deep Learning approaches which are capable of building end-to-end dialogue systems [5,18,19].

In this work, we deal with the Interactive Pattern Recognition (IPR) framework [13] that has also been proposed to represent SDS [11]; this formulation needs to estimate the joint probability distribution over the set of semantic units provided by the SU and the set of actions to be provided by the DM. In [10], such joint probability distribution was modeled by stochastic regular *bi-languages*. These languages had also been successfully proposed to deal with

L.A. Alexandre et al. (Eds.): IbPRIA 2017, LNCS 10255, pp. 22–31, 2017.
DOI: 10.1007/978-3-319-58838-4_3

Machine Translation [12]. To this end, a Probabilistic Finite State Bi-Automata (PFSBA) was defined in [10]. Because dialog management also requires keeping the values of all relevant internal variables that can be updated after each user turn, an attributed model that allows dealing with task attribute values was also proposed. So far, only the PFSBA has been experimentally validated in [11], thus, the potential of the A-PFSBA remains unexplored. On the other hand, a turn-by-turn online learning procedure was proposed in [6] aimed at adapting the PFSBA's structure and parameters at each new interaction with an user. Although the capability and flexibility of the PFSBA to learn new edges and nodes was demonstrated, there was no dialogue success rate improvement.

The first goal of this paper is to validate the A-PFSBA framework showing that attributes can contribute to a significant increase of the dialogue success rate. The second goal is to propose a novel online learning algorithm capable of improving the dialogue success rate by learning on a dialogue basis, exploiting a criterion similar to the reward functions used in reinforcement learning [15]. Because the learning procedure requires a user model to interact with the DM, an additional contribution is the proposal of a User Model that exploits the prior probabilities modeled under the A-PFSBA framework. The proposed approaches have been evaluated through various dialog generation tasks over the Let's Go corpus [7], allowing direct comparison with previous works and resulting in significant dialogue task success rate improvements.

The paper is structured as follows: Sect. 2 explains spoken dialogue interaction as an IPR framework and describes the PFSBA and A-PFSBA formulations, detailing how the Dialogue Manager and the User can be modeled. The proposed online dialogue learning procedure is then described in Sect. 3. Section 4 presents the evaluation experiments carried out and their results. Finally, the main conclusions are summarized in Sect. 5, where future guidelines are set.

2 Attributed Probabilistic Finite State Bi-Automata as Dialogue Manager

This section describes spoken dialogue interaction in terms of the Interactive Pattern Recognition (IPR) framework and how Probabilistic Finite State Bi-Automata can be used to model these interactions. At the end, the definition of a Dialog Manager and a User Model over the structure of the Attributed PFSBA is presented.

2.1 Interactive Pattern Recognition Framework

Human-machine interaction can be seen as a pattern recognition process, where both interact under an unknown distribution of states in order to complete some objectives. Within the IPR framework, the user provides feedback signals f. As a response, the system will provide an hypothesis or system action a to disambiguate the user's intention through the dialogue.

Ignoring the user feedback except for the last interaction and assuming a classical minimum-error criterion, the Bayes decision rule is simplified to maximize the posterior $P(a_t \mid q_{t-1},\ f_{t-1})$ where a_t is the system action at current turn, f_{t-1} is the last user feedback and q_{t-1} the previous state. The interpretation of the decoding d of the user feedback $f \in F$ cannot be considered a deterministic process due to Automatic Speech Recognition (ASR) errors. Thus, the space of decoded feedback is the input to the SDS from the user, usually achieved by filtering the ASR output through some Spoken Language Understanding module. Then, the best hypothesis or system action $\hat{a}$ can be obtained as follows [11]:

$$\hat{a_t}\ = \arg\ \max_a\ P(a \mid q_{t-1},\ f_{t-1}) = \arg\ \max_a \sum_d P(a,\ d \mid q_{t-1},\ f_{t-1}\ )$$

As considering every possibility for the joint probability of the action and the decoding is computationally expensive, a sub-optimal approach can be performed:

$$\hat{d}_{t-1} = \arg\ \max_d P(f_{t-1} \mid d) P(d \mid q_{t-1}\ )$$

$$\hat{a_t} \approx \arg\ \max_a P(\ a \mid \hat{d}_{t-1},\ q_{t-1}\ )$$

Similarly, the user feedback f_t depends on the previous state q_{t-1} and system action, through an unknown distribution $P(f_t|q_{t-1},\ a_{t-1})$. In this case, as the feedback produced by the system is known, there is no noisy channel that corrupts the signal f_t and no decoding procedure is needed.

2.2 Probabilistic Finite State Bi-Automata

Probabilistic Finite State Bi-Automata are suitable to model both probabilities: $P(a_t|\ q_{t-1},\ f_{t-1})$ and $P(f_t|\ q_{t-1},\ a_{t-1})$. Their goal is to maximize the probability of model M to generate a given sample of dialogues Z, being $\mathbf{z}$ the dialogues that compose sample Z.

$$\hat{M} = \arg\ \max_M\ P_M(Z) = \arg\ \max_M \prod_{z \in Z} P_M(\mathbf{z})$$

As the model learns its structure by maximizing the likelihood to fit the samples, it can also generate dialogue samples, as done by end-to-end neural networks [20].

The PSFBA model can then be defined as $\hat{M} = (\Sigma, \Delta, \Gamma, \delta q_0, P_f, P)$ where

- Σ is the alphabet of user's decoded feedbacks, $d \in \Sigma$.
- Δ is the alphabet of system actions, $a \in \Delta$.
- Γ is an extended alphabet $\Gamma \subseteq (\Sigma^{\geq m} \times \Delta^{\geq n})$ that contains the combinations of user's decoded feedbacks and system actions.
- $Q = Q_S \cup Q_U$ is the set of states labeled by bi-strings: $(\tilde{d_i} : \tilde{a_i}) \in \Gamma$.
- $\delta\ \subseteq Q \times \Gamma \times Q$ is the union of two sets of transitions $\delta = \delta_S \cup \delta_U$ as follows:
  - $\delta_S \subseteq\ Q_S \times \Gamma \times Q_U$ is a set of system transitions of the form $(q, (\epsilon : \tilde{a_i}), q')$ where $q \in Q_S,\ q' \in Q_U$ and $(\epsilon : \tilde{a_i}) \in \Gamma$.

  - $\delta_U \subseteq\ Q_U \times \Gamma \times Q_S$ is a set of user transitions of the form $(q, (\tilde{d_i} : \epsilon), q')$ where $q \in Q_U,\ q' \in Q_S$ and $(\tilde{d_i} : \epsilon) \in \Gamma$.
- $q_0 \in Q_S$ is the unique initial state: $(\epsilon : \epsilon)$ where ϵ is the empty symbol.
- $P_f : Q \rightarrow [0, 1]$ is the final-state probability distribution.
- $P\ :\ \delta \rightarrow [0, 1]$ defines the transition probability distributions $P(q, b, q') \equiv P(q',\ b \mid q)$ $\forall b \in \Gamma$ and $q, q' \in Q$ such that:

$$P_f(q) + \sum_{b \in \Gamma, q' \in Q} P(q, b, q') = 1\ \forall q \in Q$$

where transition (q, b, q') is completely defined by the initial state q and the transition state b. Thus, $\forall q \in Q,\ \forall b \in \Gamma,\ |\{q' : \{(q, b, q')\}| \leq 1$.

Taking advantage of the structural flexibility provided by the PFSBA formulation presented above, dialogue attributes can be easily incorporated to represent the transcendent variables of the dialogue as discrete values that are kept from one dialogue turn to another (e.g. specified bus number, current departure place etc.) through the inclusion of an additional alphabet Ω, which includes the discrete valued dialogue attributes seen in the sample set Z. As a result, the elements of the state alphabet Q are enhanced to $[(\tilde{d_i} : \tilde{a_i}), \tilde{\omega_i}] \in \Gamma \times \Omega$.

2.3 Dealing with Unseen Situations

Field-deployed SDS have to deal with unseen situations, so each time the user gives feedback that leads to a state $q' \notin Q$ the system state q has to be approximated using a smoothing strategy [11] as shown in Fig. 1:

$$q = \begin{cases} q', & \text{if } q'\ \in Q \\ \min_{q \in Q}\ G(q', q), & \text{otherwise} \end{cases} \tag{1}$$

where G is some function that defines the distance between the nodes. This smoothing procedure ensures that the DM can estimate unseen states. The Distance Function (G) used in the paper is defined as follows:

$$G(q, q') = \text{dist}((\tilde{d_q} : \tilde{a_q}), (\tilde{d_{q'}} : \tilde{a_{q'}})) + \lambda(|\tilde{\omega_q} \cap \tilde{\omega_{q'}}| - |\tilde{\omega_q} \cup \tilde{\omega_{q'}}|)$$

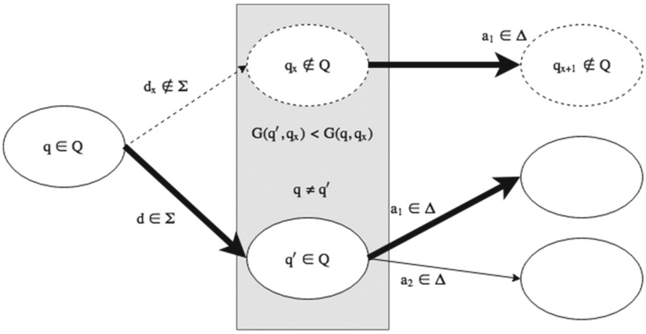

Fig. 1. Creation of candidate states and transitions through smoothing

where *dist* corresponds to the Levenshtein distance and λ is a parameter which penalizes the distance by the amount of attributes that both states differ.

Figure 1 shows how candidate states and transitions are created through the smoothing procedure. Unknown nodes such as $q_x \notin Q$ are generated through the user decoded feedback d_x and the most similar node q' is used to determine the system action $a_1 \in \varDelta$. In this process, two new states q_x and q_{x+1} are estimated.

2.4 Modeling the Dialogue Manager

Given the A-PFSBA model $\hat{M}$, a DM can be defined as a function whose goal is to return the best system action given an user feedback decoding and the state at the current turn under a policy $\varPi_{DM}$ and a smoothing strategy with a distance function G:

$$DM_{\varPi} : Q \times \varSigma \rightarrow \varDelta \times Q$$

$$\varPi_{DM}(q_t, d_t, \hat{M}, G) \rightarrow a_{t+1},\ q_{t+1}$$

where the policy $\varPi_{DM}$ can be any function that decides which action to perform. The policy used for the DM in the experimental section of the paper is the Maximum Probability (MP); in which the system action to be done is the one that maximizes $P(a_t \mid f_{t-1},\ q_{t-1})$. This is equivalent to choosing the edge of the current state with the highest transition probability at each system turn:

$$\tilde{a_t} = arg\ max_{a_{t-1,j} \in \varDelta(q_{t-1})}\ P(q_{t-1}, (\ \epsilon : a_{t-1,\ j}), q'_j)$$

2.5 Modeling the User

As data gathering and evaluation is very expensive, the most common approach to train and evaluate stochastic dialogue managers is to model a simulated user from the available data. These User Models UM interact with the DM generating synthetic dialogues, for evaluation purposes [16,17].

Since the A-PFSBA paradigm is a generative model, it captures user behavior over the intercourse of the dialogue. Thus, an structure similar to that of the Dialog Manager can be used to design an User Model, whose goal is to return some user feedback given a system hypothesis and the current state under a certain policy $\varPi_{UM}$:

$$UM_{\varPi} : Q \times \varDelta \rightarrow F \times Q$$

$$\varPi_{UM}(q_t, a_t, \hat{U}, G) \rightarrow f_{t+1},\ q_{t+1}$$

where $\hat{U}$ is the A-PFSBA used to model the user dialogue samples and G is the distance function used in the smoothing procedure for the UM. The main advantage is that its implementation is straightforward, as the same methodology can be applied both for the DM and the UM. It is important to note that the UM's policy $\varPi_{UM}$ has to be non-deterministic, in order to achieve the highest possible variance while keeping the sensibility induced to the UM using the priors defined by the bi-automaton. In order to do so, the policy employed in the current implementation is an α-weighted Random Sampling (RS), in which the action to perform is sampled from the distribution of the hypotheses seen in the current state. These priors are modified using an $\alpha \in [0, 1]$ structural constraint parameter. Being $\delta_{i,j}$ the transition probability from the state q_i to q_j it can be re-scored by α as follows:

$$\alpha(\delta_{i,j}) = \frac{\delta_{i,j}^{\alpha}}{\sum_k \delta_{i,k}^{\alpha}}$$

So, the higher α, the more constrained the user variability under the priors modeled by $\hat{U}$. Note that using $\alpha = 0$ is the same as randomly picking a possible user feedback available from the current state q_t.

3 Online Learning

The ability to adapt and learn from unseen situations on the run is a powerful property of the PFSBA formulation. [6] showed that it is flexible enough to adapt to unseen situations using smoothing techniques and controlled structural learning. The learning process was done turn-by-turn and so, there was no quality check of the learned content and every new state and transition, both good and bad, were learned. The online learning algorithm presented in this section fixes this problem, learning only useful dialogues when they are finished through the exploitation of a quality metric QM that discriminates whether a dialogue is valid or not, in a way similar to the reward function used in reinforcement learning.

Being z' the A-PFSBA structure that models an unseen dialogue sample generated by some user and the DM with dialogue model $\hat{M} = (\Sigma, \Delta, \Gamma, \Omega, \delta, q_0, P_f, P)$, the online learning method consists on merging the new states and transitions estimated during the smoothing procedure employed to deal with unseen situations by the A-PFSBA framework in z', as described in Sect. 2.3, with the dialogue model $\hat{M}$ only if the quality metric QM decides that z' is a valid dialogue. Thus, for generated unseen dialogues rendered valid by QM, the new states $q_x \notin Q$ and the corresponding set of new transitions $\delta[q_x]$ shown in Fig. 1 are learned by the DM dialogue model $\hat{M}$, so they no longer need to be estimated by the smoothing procedure. The update pseudo-algorithm is defined as follows:

4 Setup and Experiments

This section describes the experiments made on the Let's Go Corpus [7] to test the presented approaches. The main goal is to show the improvements obtained by the inclusion of attributes in the PFSBA implementation, the proposed online learning procedure and to evaluate the impact of the User Model in the learning process.

Algorithm 1. Online Learning

```
procedure A-PFSBAUpdate
    $\hat{M} \leftarrow$ DM's A-PFSBA model
    $z' \leftarrow$ Unseen Dialogue's A-PFSBA model
    if $QM(z')$ is True then
        for $q_z \in Q_{z'}$ do:
            $\hat{M} \leftarrow merge(\hat{M}, q_z, \delta[q_z])$
            $\hat{M} \leftarrow update\_edge\_count(\hat{M})$
    return $\hat{M}$
```

4.1 Corpus Description

The Let's Go SDS developed by Carnegie Mellon University (CMU) exploits the Olympus architecture using RavenClaw [2] as DM to provide schedule and route information about the city of Pittsburgh bus service to the general public. The corpus linked to such SDS was collected from real user interactions during 2005, so events like unexpected dialogue closing, spontaneous talking, sudden noise etc. are observed. Some of the corpus statistics are shown in Table 1. In the corpus, every feedback decoding is done with the CMU Phoenix Parser [14], so each user state Q_U and system state Q_S is represented by a string. The attributes are discrete values related to bus schedule information. Table 2 shows some dialogue formatting examples. The corpus was split in half to model two A-PFSBA, $\hat{M}$ to be used as the DM and $\hat{U}$ as the UM.

Table 1. Main features of the Let's Go Corpus

Let's Go Corpus statistics					
Dialogues	1840	System turns	28141	System dialogue acts	49
Attributes	14	User turns	28071	User dialogue acts	138

4.2 Impact of the Attributes and the User Model

The inclusion of attributes changes the structural behavior of the PFSBA. To evaluate this change, a total of 25 000 dialogues have been generated between the DM and the UM, using both the PFSBA and A-PFSBA formulations under the same DM/UM policy and dialogue partitions as in [6]. The employed evaluation metrics are the Task Completion (TC) rate and the Average Dialogue Length (ADL). The task is rendered complete when the DM does a coherent query to the database and retrieves the information asked by the user. Note that this metric is more constrained than the one used in previous works [6] due to the inclusion of attributes. These metrics have also been calculated for the Let's Go SDS, that uses an agenda based DM.

Table 3 shows that the inclusion of attributes increases the number of unique nodes and edges in the graph. Also, it is clear that this structural complexity is

Table 2. Let's Go dialogue formatting example

$q = [(\tilde{d_i} : \tilde{a_i}), \tilde{\omega}_i]$	System actions and user Feedbacks
$q_0 = [(\epsilon : \epsilon), \epsilon] \in Q_S$	S: Welcome to the CMU Let's Go bus information system. To get help... $\tilde{a_1}$ = inform_welcome,inform_get_help,request_query_departure_place
$q_1 = [(\tilde{a_1} : \epsilon), \epsilon] \in Q_U$	U: I'm leaving from CMU. $\tilde{d_1}$ = inform_departure_place, PlaceInformation_registered_stop $\tilde{\omega_0} = \{\}$
$q_2 = [(\tilde{a_1} : \tilde{d_1}), \tilde{\omega_0}] \in Q_S$	S: Departing from <query.departureplace CMU>. Did I get that right? $\tilde{a_2}$ = Explicit_confirm, request_query_departure_place $\tilde{\omega_0} = \{\}$
$q_3 = [(\tilde{a_2} : \tilde{d_1}), \tilde{\omega_0}] \in Q_U$	U: Yes. $\tilde{d_2}$ = Generic_yes $\tilde{\omega_1} = \{< query.departure.place >\}$

needed to create a more sensible representation of the dialogues, as the TC rate increases from 20% to 31,5%.

Results also show the impact of the UM, as better performance is achieved when the UM behavior is constrained with the priors seen in the data. A rough 31% is achieved when the α parameter is set to 0 and the UM chooses its actions from the action set at random. However, when $\alpha = 1$ the TC metric goes up to 60,02%. Because the frequent actions seen in the data are more likely to appear, this constraint results in a more sensible UM and manages to improve the RavenClaw baseline.

4.3 Online Learning Procedure

As described before, the online learning procedure is done in a dialogue basis, merging those unseen dialogues that are rendered valid by a quality metric QM in the DM model $\hat{M}$. In order to test the performance of the algorithm, 400000 dialogues were generated using the RS $\alpha = 1$ policy for both UM and DM and using the TC metric as QM. Note that the UM's Bi-Automata $\hat{U}$ never learns during this process. Results in Table 3 show that the proposed algorithm is capable of changing the shape and structure of the A-PFSBA model on the run, adapting its internal parameters to increase the dialogue task completion rate from 60,02% to 69,39%.

In addition, the evolution of the TC mean over the amount of generated dialogues and the impact of the α constraint of the UM have also been analysed. For such purpose, the TC mean was evaluated after each run of 100 generated

Table 3. Attribute and Online Learning (OL) impact on $PFSBA$.

	Nodes-DM	Edges-DM	Nodes-UM	Edges-UM	TC (%)	ADL (%)
CMU RavenClaw	—	—	—	—	54.0	32.33 ± 1.2
PFSBA $\alpha = 0$	4030	7781	4044	7652	20.08 ± 0.51	29.23 ± 0.28
A-PFSBA $\alpha = 0$	11005	14737	11058	14988	31.58 ± 1.54	31.39 ± 0.722
A-PFSBA $\alpha = 1$	11005	14737	11058	14988	$\mathbf{60.02 \pm 1.36}$	30.98 ± 0.94
A-PFSBA OL $\alpha = 1$	14700	21952	11058	14988	$\mathbf{69.39 \pm 1.34}$	31.46 ± 0.69

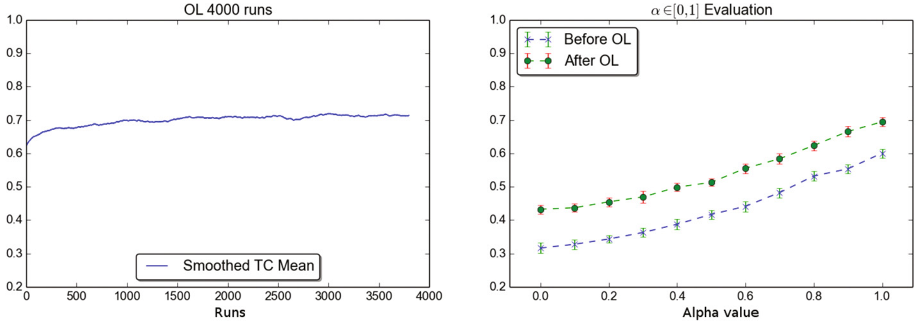

Fig. 2. Online learning impact on task completion

dialogues. The left graph of Fig. 2 depicts the TC smoothed mean over the 4000 runs. The right graph shows the mean TC score for differently constrained UM ($\alpha \in [0, 1]$) before and after the online learning process. Results demonstrate how the learning strategy converges quite rapidly after around 40 000 dialogues and that the procedure is valid for User Models with different structural constraints, where a 8–10% TC rate improvement is achieved in average.

5 Conclusions and Future Work

Throughout this paper various improvements have been proposed to previous implementations of the PFSBA. First of all, the inclusion of discrete dialogue attributes has been tested. As a consequence, the Task Completion rate has significantly increased making the generated dialogues more coherent. In addition, the inclusion of a quality metric to discriminate between successful and failed dialogues for online learning purposes has demonstrated to be a cheap-yet-effective way of controlling the learning process, as the overall improvement on the dialogue strategy is also significant. Finally, the inclusion of prior probability distributions in the User Model has shown to very significantly improve it's sensibility, demonstrating its capability to capture the user's behavior and creating a simple model to test Dialogue Managers.

Despite the promising results, the A-PFSBA is a recently proposed framework that still requires thorough experimentation and testing. As future work, we plan to explore more complex policies for the Dialog Manager, additional online learning procedures and other User Modeling techniques. The implementation of the A-PFSBA in other dialogue databases is also intended, as it is the development of an end-to-end Spoken Dialogue System.

References

1. Gorin, A.L., Riccardi, G., Wright, J.H.: How may i help you? Speech Commun. **23**(1–2), 113–127 (1997)
2. Bohus, D., Rudnicky, A.I.: The RavenClaw dialog management framework: architecture and systems. Comput. Speech Lang. **23**, 332–361 (2009)

3. Thomson, B., Yu, K., Keizer, S., Gasic, M., Jurcicek, F., Mairesse, F., Young, S.: Bayesian dialogue system for the let's go spoken dialogue challenge. In: Spoken Language Technology Workshop (SLT), pp. 460–465. IEEE (2010)
4. Hurtado, L.F., Planells, J., Segarra, E., Sanchis, E., Griol, D.: A stochastic finite-state transducer approach to spoken dialog management. In: INTERSPEECH, pp. 3002–3005 (2010)
5. Vinyals, O., Le, Q.: A Neural Conversational Model. abs/1506.05869 CoRR (2015)
6. Orozko, O.R., Torres, M.I.: Online learning of stochastic bi-automaton to model dialogues. In: Paredes, R., Cardoso, J.S., Pardo, X.M. (eds.) IbPRIA 2015. LNCS, vol. 9117, pp. 441–451. Springer, Cham (2015). doi:10.1007/978-3-319-19390-8_50
7. Raux, A., Langner, B., Bohus, D., Black, A.W., Eskenazi, M.: Let's go public! Taking a spoken dialog system to the real world. In: Proceedings of Interspeech (2005)
8. Young, S., Gasic, M., Thomson, B., Williams, D.J.: POMDP-based statistical spoken dialog systems: a review. Proc. IEEE **101**(5), 1160–1179 (2013)
9. Jurcıcek, F., Thomson, B., Young, S.: Reinforcement learning for parameter estimation in statistical spoken dialogue systems. Comput. Speech Lang. **26**(3), 168–192 (2012)
10. Torres, M.I.: Stochastic bi-languages to model dialogs. In: Finite State Methods and Natural Language Processing, p. 9 (2013)
11. Torres, M.I., Benedí, J.M., Justo, R., Ghigi, F.: Modeling spoken dialog systems under the interactive pattern recognition framework. In: Gimel'farb, G., et al. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 519–528. Springer, Heidelberg (2012)
12. Torres, M.I., Casacuberta, F.: Stochastic k-TSS bi-languages for machine transla tion. In: Proceedings of the 9th International Workshop on Finite State Models for Natural Language Processing (FSMNLP), pp. 98–106. Association for Computational Linguistics, Blois (2011)
13. Toselli, A.H., Vidal, E., Casacuberta, F. (eds.): Multimodal Interactive Pattern Recognition and Applications. Springer, Heidelberg (2011)
14. Ward, W., Issar, S.: The CMU ATIS system. In: Proceedings of ARPA Workshop on Spoken Language Technology, pp. 249–251 (1995)
15. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, vol. 1, No. 1. MIT press, Cambridge (1998)
16. Schatzmann, J., Young, S.: The hidden agenda user simulation model. IEEE Trans. Audio Speech Lang. Process. **17**(4), 733–747 (2009)
17. Schatzmann, J., Georgila, K., Young, S.: Quantitative evaluation of user simulation techniques for spoken dialogue systems. In: Proceedings of 6th SIGDIAL, pp. 45–54 (2005)
18. Williams, J.D., Zweig, G.: End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. CoRR abs/1606.01269 (2016)
19. Zhao, T., Eskenazi, M.: Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In: Proceedings of 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 1–10 (2016)
20. Serban, I.V., et al.: Building end-to-end dialogue systems using generative hierarchical neural network. In: Proceedings of 30th conference of AAAI (2016)