A Case Study on the Importance of Belief State Representation for Dialogue Policy Management

Margarita Kotti[1], *Vassilios Diakoloukas*[2], *Alexandros Papangelis*[1], *Michail G. Lagoudakis*[2], *Yannis Stylianou*[1,3]

[1]Speech Technology Group, Toshiba Research Cambridge, UK
[2]School of Electrical & Computer Engineering, Technical University of Crete, Greece
[3]Department of Computer Science, University of Crete, Greece

{margarita.kotti,alex.papangelis,yannis.stylianou}@crl.toshiba.co.uk,
vdiak@telecom.tuc.gr, lagoudakis@ece.tuc.gr

Abstract

A key component of task-oriented dialogue systems is the belief state representation, since it directly affects the policy learning efficiency. In this paper, we propose a novel, binary, compact, yet scalable belief state representation. We compare the standard verbose belief state representation (268 dimensions) with the domain-independent representation (57 dimensions) and the proposed representation (13 or 4 dimensions). To test those representations, the recently introduced Advantage Actor Critic (A2C) algorithm is exploited. The latter has not been tested before for any representation apart from the verbose one. We study the effect of the belief state representation within A2C under 0%, 15%, 30%, and 45% semantic error rate and conclude that the novel binary representation in general outperforms both the domain-independent and the verbose belief state representation. Further, the robustness of the binary representation is tested under more realistic scenarios with mismatched semantic error rates, within the A2C and DQN algorithms. The results indicate that the proposed compact, binary representation performs better or similarly to the other representations, being an efficient and promising alternative to the full belief.

Index Terms: dialogue systems, belief state, binary belief state representation, domain-independent parametrisation

1. Introduction

Conversational systems is a thriving research area with numerous real-world applications, such as call-centers [1], tourist information [2], car navigation [3], education [4], social robots [5], banking [6], health services[7], and games and entertainment [8]. Many commercial applications are also emerging, such as intelligent personal assistants, e.g. Microsoft's Cortana, Apple's Siri, and Amazon's Echo among others. Task-oriented information seeking is a common case for a statistical dialogue system, where a database is inquired for a specific item given a set of hard restrictions [9]. In the traditional case, the database covers one task, that is, a single domain, such as searching for a restaurant. Challenges derive from the fact that the number of belief state (BS) features and dialogue actions increases rapidly as more context and domains are taken into account [10].

1.1. Relation to prior work

Regarding representations, the standard belief state (BS) features have been widely used [11]. Their main advantage is that they have been used and optimised for over a decade now and are publicly available. Their main disadvantage is that they are domain-dependent, very sparse, and with a high degree of redundancy. The problem of operating across domains is resolved by the use of domain-independent (DIP) features [12]. DIP results in a fixed-dimensional space, so domains with different belief spaces are automatically mapped onto a common belief space base. In this work, we make use of the amended DIP representation, with four new features, such as the normalised turn number, referred simply as DIP features from now on.

A family of novel state representations based on binary features is firstly presented in this paper. This representation is domain-transferable and compact, yet robust. The key idea is that, in many domains, the knowledge of exact slot values may not be as important for dialogue action selection, as the knowledge of whether a slot value is known or not. Therefore, the proposed features mark the presence or absence of a slot value at each dialogue turn with a binary (0-absent or 1-present) value. These abstract features are combined linearly for all domain slots (BinLin) and may be augmented with auxiliary binary information about slot requests by the user (BinAux).

Regarding the policy manager part, traditional approaches are using either Markov Decision Processes (MDPs) [13] or Partially Observable Markov Decision Process (POMPDS) [14] to solve the sequential decision problem. The most common approach to solving the optimisation problem of sequential decision making is the use of reinforcement learning (RL) [15]. Most recently, the use of deep neural networks (NNs) to solve the optimisation problem is exploited. This may be attributed partially to the fact that deep architectures with several hidden layers can be efficiently used for complex tasks and environments. Promising results have been achieved for example with the Deep Q-Network (DQN) systems [16], leading to many variations, such as the NDQN [17]. Lately, a trend towards policy-gradient methods, such as Advantage Actor-Critic (A2C), appeared [18], which have proven to be efficient in Atari games, car simulators, and physics simulators [19]. Focusing on dialogue systems, the authors of [18] test a deep A2C algorithm, either initialised with supervised learning or not, on the restaurants domain, as in this paper. Their main finding is that the A2C algorithm converged faster than the DQN and the GP-SARSA algorithms. Furthermore, A2C approaches are presented in [20], where improvements are introduced either by the use of experience replay and the trust region policy optimisation method or by an alternative approximation of the advantage function. Experimental results prove the suitability of the A2C algorithms for on-line learning. It should be noted that any policy manager could serve for the aims of our work, since it is the different input representations to the policy manager that are

10.21437/Interspeech.2018-1293

studied here. We decided to resort to A2C methods, since they are quite new and have not been tested before for representations other than the full belief, thus adding to the novelty.

1.2. Contribution of this work

This paper presents: i) a novel binary belief state representation (BinLin, BinAux); ii) the first use of the A2C algorithm with compact representations (DIP, BinLin, BinAux); iii) a systematic comparison between the full and the compact representations in matched and miss-matched semantic error conditions; and iv) a confirmation that compact representations can attain high performance at considerably lower computational cost.

2. Method

2.1. The dialogue model

For the sake of completeness, a short introduction to the dialogue model is presented. The input to the model is the belief state (BS) $\mathbf{b}_t$ at each time t. The model's role is to find an optimal policy π that maximises the discounted total return:

$$R = \sum_{t=0}^{T-1} \gamma^t \, r_t(\mathbf{b}_t, a_t), \tag{1}$$

where t is the current turn number, T is the total number of dialogue turns, γ is a discount factor, and $r_t(\mathbf{b}_t, a_t)$ is the reward of taking action a_t when being at BS $\mathbf{b}_t$. For the A2C dialogue manager presented here, the output is a distribution of probabilities over the next action $\pi(a_t|\mathbf{b}_t)$ and also one extra scalar standing for the value function of the BS $\mathbf{b}$ given the policy:

$$V(\mathbf{b}) = \sum_{t=0}^{T-1} \gamma^t \, r_t\big(\mathbf{b}_t, a_t \bigm| \pi, \mathbf{b}_0 = \mathbf{b}\big). \tag{2}$$

2.2. The A2C algorithm

A2C is a policy-gradient RL method [20]. It approximates the policy directly in a model-free way. That is, the policy is represented as a network that maps from the BS (or equivalently any other feature set) to the action space in a probabilistic manner:

$$\pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t) = \pi(a_t|\mathbf{b}_t; \boldsymbol{\theta}) = \mathcal{P}(a_t|\mathbf{b}_t, \boldsymbol{\theta}_t = \boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\theta}$ is the weight vector of the policy network. The policy parameters are learned by gradient-based optimisation. In order to search for the optimal $\boldsymbol{\theta}$ parameters, we need to define the objective function $J(\boldsymbol{\theta})$, the expected reward over all possible dialogue trajectories given a starting state. Hence, the aim is to maximise $J(\boldsymbol{\theta})$ based on the Policy Gradient Theorem [21]:

Theorem 1. *For any differentiable policy $\pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t)$ and for the average reward or the start-state objective function, the policy gradient can be computed as*

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\big[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t) Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t, a_t)\big]. \tag{4}$$

This can be seen as having $\pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t)$ be the actor and $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t, a_t) = \mathbb{E}(r_t + \gamma r_{t+1} + \ldots|\mathbf{b}_t, a_t)$ be the critic [22]. Nevertheless, the direct use of Equation (4) has proven to be unstable due to high variance [23]. To reduce variability without changing the gradient, a baseline function is used. It holds that subtracting $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t)$ from $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t, a_t)$ does not change the gradient. Hence, the advantage function is introduced:

$$A^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_t, a_t) = Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t, a_t) - V^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t), \tag{5}$$

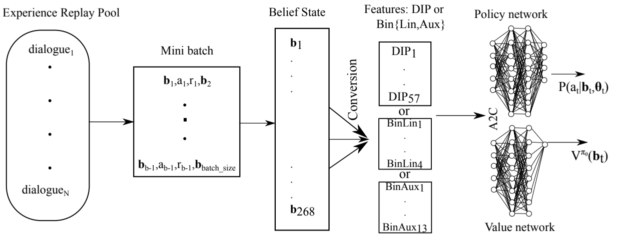

Figure 1: *A2C-{BS, DIP, BinLin, BinAux} system architecture: The experience replay pool contains several dialogues from which a mini batch of turns is selected. Then the* 268*-dimensional BS is transformed to the* 57*-dimensional DIP,* 4*-dimensional BinLin or* 13*-dimensional BinAux features that are given as input to the two NNs. The policy network has the* $\boldsymbol{\theta}$ *weight vector and the value network has the* $\mathbf{w}$ *weight vector. The output is the probability over actions* $\mathcal{P}(a_t|\mathbf{b}_t, \boldsymbol{\theta}_t)$ *and the value function of the specific belief state* $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t)$ *respectively. For A2C-BS, no conversion takes place, i.e. the belief state is given directly as input to the two NNs.*

with $\mathbf{w}$ being the weight vector of the value network. The insight of using $A^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}$ is to determine not just how good the actions are, but how much better they turn out to be than expected (advantage). Combining Equation (4) with Equation (5) we get:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\big[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t) A^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_t, a_t)\big] \tag{6}$$

Here, $\pi_{\boldsymbol{\theta}}(a_t|\mathbf{b}_t)$ is the actor and $A^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_t, a_t)$ is the critic. Finally, to reduce the number of network parameters, the advantage function is estimated using the temporal difference [24]:

$$A^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_t, a_t) \approx \delta_{\mathbf{w}}(t) = r_t + \gamma V^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_{t+1}) - V^{\pi_{\boldsymbol{\theta}}}_{\mathbf{w}}(\mathbf{b}_t). \tag{7}$$

A graphical representation of the system architecture described in this Section is depicted in Figure 1.

2.3. DIP feature representation

DIP is an alternative representation of the BS. DIP features are domain-independent, so they can facilitate learning a policy in an abstract way. Policies learned on that fixed-dimensional base space can be transfered to new domains. This is done by exploring the nature and commonness of the underlying tasks in different domains and parameterising different slots according to their relations and potential contributions. The original set [12] includes the number of values, their distribution in the database (DB), and how likely filling a slot will (or not) reduce the number of matching DB records below a threshold. This representation is further amended by the importance and priority of the slots [25]. Four additional features are added for this work:

- how many entities of the database (DB) satisfy the query given the current BS $\mathbf{b}$: $|DB(\mathbf{b})|$
- whether the system has already provided an entity to the user or not
- whether the system couldn't provide any information (i.e. can't help)
- the normalised turn number t/N_T, with t being the current turn number and N_T the maximum number of turns.

The overall number of DIP features is 57.

2.4. Binary feature representation

The proposed binary features consist an abstraction of the verbose full belief state (BS) representation proposed within the PyDial toolkit framework. This latter belief state is a high-dimensional vector, which is extremely sparse and has a high degree of redundancy. Our goal is to define compact, yet robust, forms of the state-vector representation, which could serve as a more informative alternative to the original belief. The reduced size of the proposed binary features leads to diminished computational cost, hence facilitating real-time systems, as well as systems that don't have access to large computational power.

The proposed binary abstraction is valid under the assumption that the most important information for a dialogue manager is whether a slot name is activated (its value is not NULL) at a specific dialogue turn, rather than knowing its exact slot value. This idea is also motivated and supported by the fact that the distributions over slot values within the belief vector are in almost all dialogue instances extremely sparse; the grand majority of the slot values have zero probability. Under the proposed abstraction, it is not necessary to know the detailed domain's ontology and all the possible values a slot can take. Likewise, it is not required to unnecessarily grow the feature vectors to accommodate discreteness for all possible slot values. Instead, we can obtain only binary values for the available slots in the belief vector (one binary feature per slot).

Specifically, the proposed BinLin representation comprises the binary abstract features for the domain slots combined linearly. More specifically, the slots are turned into a linear binary vector of size equal to the number of slots and we mark the presence or absence of a slot value at each dialogue turn with a binary (0-absent or 1-present) value. In this way, the abstract binary feature vector only consists of n components for n slots. Hence, in the *CamRestaurants* domain, BinLin will create a feature-vector of size 4, since there are only 4 domain slots: area (7 values), food (93 values), name (114 values), price range (5 values). This simple representation can be augmented by an auxiliary part that contains the information of whether a slot has been requested or not by the user, information which is otherwise discarded in BinLin, although it is contained in the verbose belief state. The addition of the auxiliary part to BinLin leads to the BinAux features and adds robustness to the pure binary linear (BinLin) representation. In the *CamRestaurants* domain, BinAux yields a feature-vector of size $4 + 9 = 13$, since, in addition to the 4 domain slots, there are 9 slots in the auxiliary part that the user may request, namely: address, area, description, food, name, phone, postcode, price range, and signature. Notice that the proposed representations (both BinLin and BinAux) scale linearly with the number of slots and therefore are applicable to large domains. Apart from the two representations described here, it is possible to design a family of similar binary feature choices, for example using separate and complementary on/off bit features for presence and absence of a slot value.

3. Experimental Results

Experiments are conducted using the software toolkit PyDial [11]. The *CamRestaurants* domain refers to restaurants in the Cambridge, UK area and consists of 113 restaurants, each with 7 slots (database attributes), of which 4 can be used by the system to constrain the search (food, area, name, price range) and 3 are system-informable properties (phone number, address, postcode) available once a database entity has been found. An agenda-based simulator is used, operating at dialogue act level.

A2C is realised as two fully connected feed-forward NNs with two hidden layers: the policy network and the value network. Both NNs have the same architecture, besides the output layer, and share the same input. Mini batches of dialogue experiences $(\mathbf{b}_j, a_j, r_j, \mathbf{b}_{j+1})$ are randomly sampled from a replay pool. The dialogue experiences can be represented either as BS, DIP, BinLin or BinAux features. That is, the input to the NNs can be either the 268-dimensional BS from PyDial, the 57-dimensional DIP, the 4-dimensional BinLin, or the 13-dimensional BinAux. The output size is $15 + 1 = 16$, corresponding to 15 dialogue actions determining the system intent at the semantic level plus one scalar value for the $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{b}_t)$, as can be seen in Figure 1. Weights are randomly initialised from a Gaussian function with a zero mean value and a standard deviation of 0.01. The value of the γ discount factor is set to 0.99. The total return R is computed as: $R = I_{\text{success}} - 0.05 \times T$, where I_{success} is the indicator function of success and T is the dialogue length that has a maximum value of 30. A dialogue is deemed successful, if the retrieved restaurant complies with the set of target preferences provided by the simulated user for the specific dialogue. I_{success} can also be seen as the task completion flag. Small reward values are used to avoid gradient instability and network training inconsistency. The Adam optimiser [26] is used with an initial learning rate of 0.001 for both the actor and the critic. During training, an ϵ-greedy policy is used, where ϵ is initially set to 0.5 and is annealed to 0.0 over 2000 dialogues.

For A2C-BS, the input to the NNs is a BS with 268 dimensions. The NNs have two hidden layers, the first with 130 neurons and the second with 50 neurons. The mini batch size is 64 and the capacity of the experience replay pool is 6000. For the A2C-DIP case, the input to the NNs has 57 dimensions. The NNs have two hidden layers, the first with 200 neurons and the second with 150 neurons. The mini batch size is 70 and the capacity of the experience replay pool is 100. For A2C-Bin{Lin,Aux}, the input to the NNs has 4 and 13 dimensions respectively. The NNs have two hidden layers, the first with 250 neurons and the second with 75 neurons. The mini batch size is 64 and the capacity of the experience replay pool is 1000. The difference in the size of the experience replay pool can be partially attributed to the different sizes and nature of the input.

3.1. Training and testing in matched conditions

The studied models are first evaluated under 0%, 15%, 30%, and 45% semantic error rate, as can be seen in Table 1. The number of training dialogues is 2000 and the number of testing dialogues is 300. To validate the quality and stability of the learned policies, accuracy is averaged over 5 independent runs of 2000 training dialogues / 300 evaluation dialogues cycles each. Two figures-of-merit are used: the objective success and the reward, the latter since this is the criterion the algorithm aims to optimise. Note that the optimal set of figures-of-merit for dialogue systems evaluation is still an open problem [27].

For the 0% semantic error rate, A2C-BinAux achieves significantly higher success and reward, while all other representations are quite close to each other in terms of performance. In addition, it is remarkable that even the extremely compact, 4-dimensional feature set, used in the A2C-BinLin method, is able to achieve comparable performance to the much higher-dimensional representations. It should be noted that all feature representations have similar standard deviation, which means that they exhibit similar robustness.

As expected, the performance for all methods deteriorates, when noise is introduced to the semantic input; however, the

Table 1: *Dialogue success rate and reward in the* CamRestaurants *domain for various semantic error rates under matched error rate conditions. The results are given in the form of mean value (standard deviation). Best performance is highlighted in boldface.*

Method	Semantic error rate (%)							
	0%		15%		30%		45%	
	success	reward	success	reward	success	reward	success	reward
A2C-BS	83.9 (6.2)	10.4 (1.1)	75.9 (7.6)	8.1 (1.5)	64.5 (7.5)	5.4 (1.3)	46.5 (11.3)	1.1 (1.8)
A2C-DIP	82.0 (8.3)	10.0 (1.6)	73.8 (9.8)	6.8 (2.0)	57.0 (10.1)	3.3 (1.8)	47.6 (8.8)	**3.0** (1.5)
A2C-BinLin	82.4 (8.2)	10.3 (1.5)	76.7 (9.3)	8.7 (1.9)	61.8 (12.6)	5.1 (2.3)	49.7 (8.8)	1.8 (1.5)
A2C-BinAux	**94.8** (7.6)	**12.9** (1.5)	**82.5** (5.9)	**9.6** (1.2)	**69.9** (11.2)	**6.5** (2.1)	**52.9** (12.9)	2.5 (2.3)

relative performance picture remains the same. Specifically, for the 15% semantic error rate, the A2C-BinAux exhibits a performance deterioration similar to all other methods, but still remains the highest performing one. The differences become less prominent for the 30% and the 45% semantic error rates, although A2C-BinAux still outperforms all other representation choices. Clearly, the addition of the auxiliary part in BinAux has a significant positive effect on the proposed binary representation. Overall, with respect to all error rates, A2C-BinAux is systematically better than all other representations for both success and reward, despite the 13-dimensional representation.

3.2. Training and testing in mismatched conditions

In real-world situations the semantic error is not zero, due to a multitude of factors, such as ASR mistakes (e.g. acoustic confusability), ambiguity of natural language, incomplete utterances, etc. In existent conversational systems the error rate of the top hypothesis is typically 20%-30%. To approach real-word scenarios and test for robustness under different training and testing semantic error rates, we applied the same experimental protocol, but with mismatched conditions. That is, we trained the policy with a 15% semantic error rate and tested with 30% and 45% semantic error rates. Since the 0% rate is unlikely to occur in a real world scenario, it is not studied here. The 30% and 45% semantic error rates are chosen, since making the system available to real users is expected to lead in an increase to the word error rate, as has already been proven [28][29].

The mismatched conditions results can be seen in Table 2. Once again, A2C-BinAux is clearly the best performing method, both in success and reward. It is also worth noting that for the mismatched conditions A2C-BinAux exhibits the lowest variation among all methods, even in the Train 15% – Test 45% case, where it comes second in reward performance. Once again, A2C-BinLin remains close to A2C-BS despite its simplicity and small dimension; similarly to the matched case, the results indicate that the addition of the user-requested slots (the auxiliary part) is important to achieve higher performance.

Table 2: *Dialogue success rate and reward in the* CamRestaurants *domain for various semantic error rates under mismatched conditions, shown as mean value (standard deviation).*

Method	Semantic error rate (%)			
	Train 15% - Test 30%		Train 15% - Test 45%	
	success	reward	success	reward
A2C-BS	68.3 (5.6)	5.8 (1.0)	54.9 (5.0)	2.6 (0.8)
A2C-DIP	64.8 (9.2)	5.1 (1.8)	54.2 (9.9)	**4.2** (2.2)
A2C-BinLin	66.3 (6.4)	5.9 (1.2)	55.7 (5.2)	3.4 (0.9)
A2C-BinAux	**72.1** (5.1)	**6.8** (1.0)	**58.9** (3.4)	3.5 (0.6)

3.3. Comparison to other methods

Although the policy manager per se is not the focus of this paper, for comparison purposes, the A2C algorithm was substituted with DQN [16] [30] and the representations were tested under the mismatched conditions of the previous section. Comparative results can be seen in Table 3.

The use of the well-studied DQN gives a significant performance gain to the BS and DIP representations. Interestingly, DQN-DIP moves to the top, whereas DQN-BS and DQN-BinAux perform similarly. DQN-BinLin exhibits lower performance compared to DQN-BinAux, which is consistent with the A2C results. In any case, the choice of A2C or DQN has minimal influence on the performance of BinLin and BinAux.

4. Conclusion and Future Work

This paper introduces the first use of the A2C algorithm along with the BinLin, BinAux, as well as the DIP, features. It was found that the reduction of dimensions obtained by the conversion of the BS to DIP or Bin{Lin,Aux} has noteworthy advantages, both when the training and testing of the system is done under the same semantic error rate conditions, as well as when the testing semantic error rate is substantially higher than the training one. This may be partially attributed to the redundancy and sparseness of the standard BS representation. In all experimental conditions within A2C, the proposed BinAux representation is consistently better than the BS and DIP ones. The newly proposed BinAux features are i) domain-transferable, ii) low-dimensional, offering an order of magnitude reduction compared to the full BS, iii) scalable to domains with many slots and many slot values, and iv) computationally efficient. In the future we aim to test BinAux on larger and multiple domains to assess if performance and scalability findings generalize.

5. Acknowledgements

We thank Pawel Budzianowski from Dialogue Systems Group, CUED for the fruitful conversations and advice.

Table 3: *Dialogue success rate and reward in the* CamRestaurants *domain for various semantic error rates under mismatched conditions with a DQN policy (instead of A2C).*

Method	Semantic error rate (%)			
	Train 15% - Test 30%		Train 15% - Test 45%	
	success	reward	success	reward
DQN-BS	72.7 (5.6)	7.2 (1.1)	59.9 (4.9)	3.9 (1.0)
DQN-DIP	**76.8** (5.2)	**9.0** (1.0)	**67.2** (2.6)	**6.4** (0.5)
DQN-BinLin	66.7 (8.9)	6.1 (1.6)	56.5 (7.4)	3.5 (1.1)
DQN-BinAux	70.6 (7.2)	6.8 (1.3)	59.2 (6.7)	3.8 (0.9)

6. References

[1] H. Hardy, T. Strzalkowski, and M. Wu, “Dialogue management for an automated multilingual call center,” in *Proc. HLT-NAACL 2003 Workshop Research Directions in Dialogue Processing*, May-June 2003, pp. 10–12.

[2] P. Budzianowski, S. Ultes, P.-H. Su, N. Mrksic, T.-H. Wen, I. Casanueva, L. M. Rojas-Barahona, and M. Gasic, “Sub-domain modelling for dialogue management with hierarchical reinforcement learning,” in *Proc. SIGdial Workshop Discourse and Dialogue*, August 2017, pp. 86–92.

[3] K. Kim, C. Lee, S. Jung, and G. G. Lee, “A frame-based probabilistic framework for spoken dialog management using dialog examples,” in *Proc. SIGdial Workshop Discourse and Dialogue*, June 2008, pp. 120–127.

[4] P.-H. Su, Y.-B. Wang, T.-H. Yu, and L.-S. Lee, “A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, May 2013, pp. 8213–8217.

[5] R. Jiang, Y. K. Tan, D. K. Limbu, T. A. Dung, and H. Li, “Component pluggable dialogue framework and its application to social robots,” in *Proc. Int. Workshop Spoken Dialogue Systems*, November 2014, pp. 225–237.

[6] H. Melin, A. Sandell, and M. Ihse, “Ctt-bank: A speech controlled telephone banking system-an initial evaluation,” *Speech, Music and Hearing Quarterly Progress and Status Report (TMH-QPSR)*, vol. 1, pp. 1–27, 2001.

[7] A. Papangelis, R. Gatchel, V. Metsis, and F. Makedon, “An adaptive dialogue system for assessing post traumatic stress disorder,” in *Proc. Int. Conf. Pervasive Technologies Related to Assistive Environments*, May 2013, pp. 49:1–49:4.

[8] J. He, J. Chen, X. He, J. Gao, L. Li, L. Deng, and M. Ostendorf, “Deep reinforcement learning with a natural language action space,” in *Proc. Annual Meeting of the Association for Computational Linguistics*, August 2016, pp. 1621–1630.

[9] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” in *Proc. European Chapter of the Association for Computational Linguistics*, April 2007, pp. 438–449.

[10] H. Cuayáhuitl, S. Yu, A. Williamson, and J. Carse, “Scaling up deep reinforcement learning for multi-domain dialogue systems,” in *Proc. Int. Joint Conf. Neural Networks*, May 2017, pp. 3339–3346.

[11] S. Ultes, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.-H. Wen, M. Gasic, and S. Young, “PyDial: A Multi-domain Statistical Dialogue System Toolkit,” in *Proc. Annual Meeting of the Association for Computational Linguistics*, July 2017, pp. 73–78.

[12] Z. Wang, T.-H. Wen, P.-H. Su, and Y. Stylianou, “Learning domain-independent dialogue policies via ontology parameterisation.” in *Proc. SIGdial Workshop Discourse and Dialogue*, September 2015, pp. 412–416.

[13] E. Levin, R. Pieraccini, and W. Eckert, “Learning dialogue strategies within the markov decision process framework,” in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, December 1997, pp. 72–79.

[14] M. Gašić and S. Young, “Gaussian processes for POMDP-based dialogue manager optimization,” *IEEE/ACM Trans. Audio, Speech and Language Processing*, vol. 22, no. 1, pp. 28–40, Jan. 2014.

[15] P. Shah, D. Hakkani-Tür, and L. Heck, “Interactive reinforcement learning for task-oriented dialogue management,” in *Proc. Workshop on Deep Learning for Action and Interaction, Neural Information Processing Systems*, December 2016.

[16] A. Papangelis and Y. Stylianou, “Single-model multi-domain dialogue management with deep learning,” in *Proc. Int. Workshop Spoken Dialogue Systems Technology*, June 2017.

[17] H. Cuayahuitl and S. Yu, “Deep reinforcement learning of dialogue policies with less weight updates,” in *Proc. Int. Conf. Speech Communication Association (InterSpeech)*, August 2017, pp. 2511–2515.

[18] M. Fatemi, L. El Asri, H. Schulz, J. He, and K. Suleman, “Policy networks with two-stage training for dialogue systems,” in *Proc. SIGdial Workshop Discourse and Dialogue*, September 2016, pp. 101–110.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *Proc. Int. Conf. Machine Learning*, June 2016, pp. 1928–1937.

[20] P.-H. Su, P. Budzianowski, S. Ultes, M. Gašić, and S. Young, “Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management,” in *Proc. SIGdial Workshop Discourse and Dialogue*, August 2017, pp. 147–157.

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, Cambridge, 1998.

[22] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, “Natural actor-critic algorithms,” *Automatica*, vol. 45, no. 11, pp. 2471 – 2482, November 2009.

[23] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992.

[24] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” in *Proc. Int. Conf. Learning Representations*, May 2016.

[25] A. Papangelis and Y. Stylianou, “Multi-domain spoken dialogue systems using domain-independent parametrisation,” in *Proc. Int. Workshop Domain Adaptation for Dialog Agents*, September 2016.

[26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[27] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, “How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation,” in *Proc. Conf. Empirical Methods in Natural Language Processing*, November 2016, pp. 2122–2132.

[28] J. Glass, J. Polifroni, S. Seneff, and V. Zue, “Data collection and performance evaluation of spoken dialogue systems: The MIT experience,” in *Proc. Int. Conf. Spoken Language Processing*, October 2000, pp. 1–4.

[29] J. Schatzmann, B. Thomson, and S. Young, “Error simulation for training statistical dialogue systems,” in *Proc. IEEE Workshop Automatic Speech Recognition Understanding*, December 2007, pp. 526–531.

[30] I. Casanueva, P. Budzianowski, P. Su, N. Mrkšic, T. Wen, S. Ultes, L. Rojas-Barahona, S. Young, and M. Gašic, “A benchmarking environment for reinforcement learning based task oriented dialogue management,” in *Proc. Deep Reinforcement Learning Symposium, Neural Information Processing Systems*, December 2017.