Sub-domain Modelling for Dialogue Management with Hierarchical Reinforcement Learning

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić,
Tsung-Hsien Wen, Iñigo Casanueva, Lina Rojas-Barahona and Milica Gašić
Cambridge University, Engineering Department, Trumpington Street, Cambridge, UK
{pfb30,mg436}@cam.ac.uk

Abstract

Human conversation is inherently complex, often spanning many different topics/domains. This makes policy learning for dialogue systems very challenging. Standard flat reinforcement learning methods do not provide an efficient framework for modelling such dialogues. In this paper, we focus on the under-explored problem of multi-domain dialogue management. First, we propose a new method for hierarchical reinforcement learning using the *option* framework. Next, we show that the proposed architecture learns faster and arrives at a better policy than the existing flat ones do. Moreover, we show how pretrained policies can be adapted to more complex systems with an additional set of new actions. In doing that, we show that our approach has the potential to facilitate policy optimisation for more sophisticated multi-domain dialogue systems.

1 Introduction

The statistical approach to dialogue modelling has proven to be an effective way of building conversational agents capable of providing required information to the user (Williams and Young, 2007; Young et al., 2013). Spoken dialogue systems (SDS) usually consist of various statistical components, dialogue management being the central one. Optimising dialogue management can be seen as a planning problem and is normally tackled using reinforcement learning (RL). Many approaches to policy management over single domains have been proposed over the last years with ability to learn from scratch (Fatemi et al., 2016; Gašić and Young, 2014; Su et al., 2016; Williams and Zweig, 2016).

The goal of this work is to propose a coherent framework for a system capable of managing conversations over multiple dialogue domains. Recently, a number of frameworks were proposed for handling multi-domain dialogue as multiple independent single-domain sub-dialogues (Lison, 2011; Wang et al., 2014; Mrkšić et al., 2015; Gašić et al., 2015). Cuayáhuitl et al. (2016) proposed a network of deep Q-networks with an SVM classifier for *domain selection*. However, such frameworks do not scale to modelling complex conversations over large state/action spaces, as they do not facilitate conditional training over multiple domains. This inhibits their performance, as domains often share sub-tasks where decisions in one domain influence learning in the other ones.

In this paper, we apply *hierarchical reinforcement learning* (HRL) (Barto and Mahadevan, 2003) to dialogue management over complex dialogue domains. Our system learns how to handle complex dialogues by learning a multi-domain policy over different domains that operate on independent time-scales with temporally-extended actions.

HRL gives a principled way for learning policies over complex problems. It overcomes the curse of dimensionality which plagues the majority of complex tasks by reducing them to a sequence of sub-tasks. It also provides a learning framework for managing those sub-tasks at the same time (Dietterich, 2000; Sutton et al., 1999b; Bacon et al., 2017).

Even though the first work on HRL dates back to the 1970s, its usefulness for dialogue management is relatively under-explored. A notable exception is the work of Cuayáhuitl (2009; 2010), whose method is based on the MAXQ algorithm (Dietterich, 2000) making use of hierarchical abstract machines (Parr and Russell, 1998). The main limitation of this work comes from the tabular approach which prevents the efficient approximation of the state space and the objective function. This is crucial for scalability of spoken dia-

Proceedings of the SIGDIAL 2017 Conference, pages 86–92,
Saarbrücken, Germany, 15-17 August 2017.

logue systems to more complex scenarios. Parallel to our work, Peng et al. (2017) proposed another HRL approach, using deep Q-networks as an approximator. In separate work, we found deep Q-networks to be unstable (Su et al., 2017); in this work, we focus on more robust estimators.

The contributions of this paper are threefold. First, we adapt and validate the option framework (Sutton et al., 1999b) for a multi-domain dialogue system. Second, we demonstrate that hierarchical learning for dialogue systems works well with function approximation using the GPSARSA algorithm. We chose the Gaussian process as the function approximator as it provides uncertainty estimates which can be used to speed up learning and achieve more robust performance. Third, we show that independently pre-trained domains can be easily integrated into the system and adapted to handle more complex conversations.

2 Hierarchical Reinforcement Learning

Dialogue management can be seen as a control problem: it estimates a distribution over possible user requests – *belief states*, and chooses what to say back to the user, i.e. which *actions* to take to maximise positive user feedback – the *reward*.

Reinforcement Learning The framework described above can be analyzed from the perspective of the *Markov Decision Process* (MDP). We can apply RL to our problem where we parametrize an optimal policy $\pi : \mathcal{B} \times \mathcal{A} \rightarrow [0, 1]$. The learning procedure can either directly look for the optimal policy (Sutton et al., 1999a) or model the Q-value function (Sutton and Barto, 1999):

$$Q^{\pi}(\mathbf{b}, a) = \mathrm{E}_{\pi}\{\sum_{k=0}^{T-t} \gamma^{k} r_{t+k} | \mathbf{b}_t = \mathbf{b}, a_t = a\},$$

where r_t is the reward at time t and $0 < \gamma \leq 1$ is the discount factor. Both approaches proved to be an effective and robust way of training dialogue systems online in interaction with real users (Gašić et al., 2011; Williams and Zweig, 2016).

Gaussian Processes in RL Gaussian Process RL (GPRL) is one of the state-of-the-art RL algorithms for dialogue modelling (Gašić and Young, 2014) where the Q-value function is approximated using Gaussian processes with a zero mean and chosen kernel function $k(\cdot, \cdot)$, i.e.

$$Q(\mathbf{b}, a) \sim \mathcal{GP}\left(0, k((\mathbf{b}, a), (\mathbf{b}, a))\right).$$

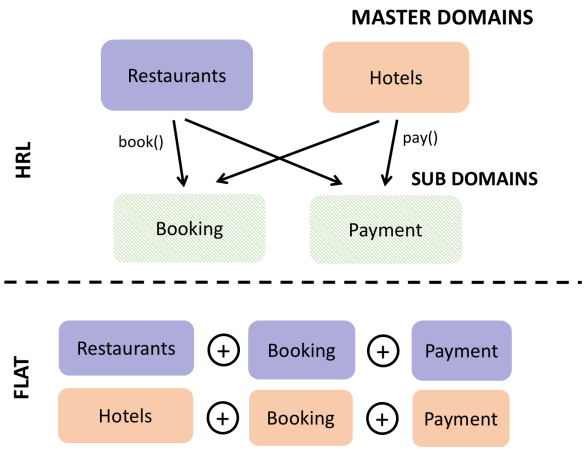

Figure 1: Comparison of two analysed architectures.

Gaussian processes follow a pure Bayesian framework, which allows one to obtain the posterior given a new collected pair $(\mathbf{b}, a)$. The trade-off between exploration and exploitation is handled naturally as given belief state $\mathbf{b}$ at the time t we can sample from posterior $Q(\mathbf{b}, a)$ over set of available actions $\mathcal{A}$ to choose the action with the highest sampled Q-value.

Hierarchical Policy Standard *flat* models where a single Markov Decision Process is responsible for solving multi-task problems have proven to be inefficient. These models have trouble overcoming the cold start problem and/or suffer from the curse of dimensionality (Barto and Mahadevan, 2003). This pattern was also observed with state-of-the-art models proposed recently (Mnih et al., 2013; Duan et al., 2016).

To overcome this issue, many frameworks have been proposed in the literature (Fikes et al., 1972; Laird et al., 1986; Parr and Russell, 1998). They make use of hierarchical control architectures and learning algorithms whereby specifying a hierarchy of tasks and reusing parts of the state space across many sub-tasks can greatly improve both learning speed and agent performance.

The key idea is the notion of *temporal abstraction* (Sutton et al., 1999b) where decisions at the given level are not required at each step but can call temporally-extended sub-tasks with their own policies.

The Option Framework One of the most natural generalisations of flat RL methods to com-

plex tasks and easily interchangeable with primitive actions is the *option* model (Sutton et al., 1999b). The option is a generalisation of a single-step action that might span across more than one time-step and can be used as a standard action.From mathematical perspective option is a tuple $\langle \pi, \beta, \mathcal{I} \rangle$ that consists of policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ which conducts the option, stochastic termination condition $\beta : \mathcal{S} \rightarrow [0, 1]$ and an input set $\mathcal{I} \subseteq \mathcal{S}$ which specifies when the option is available.

As we consider hierarchical architectures with temporally extended activities, we have to generalise the MDP to the semi-Markov Decision Process (SMDP) (Parr and Russell, 1998) where actions can take a variable amount of time to complete. This creates a division between *primitive* actions that span over only one action (and can be seen as a classic reinforcement learning approach) and *composite* actions (options) that involve an execution of a sequence of primitive actions. This introduces a policy μ over options that selects option o in state s with probability $\mu(s, o)$, o's policy might in turn select other options until o terminates and so on. The value function for option policies can be defined in terms of the value functions of the semi-Markov flat policies (Sutton et al., 1999b). Define the value function under a semi-Markov flat policy as:

$$V^{\pi}(s) = \mathbb{E}\{r_{t+1} + \gamma r_{t+2} + ... | E(\pi, s, t)\},$$

where $E(\pi, s, t)$ is the event of π being initiated at time t in s. The value function for the policy over options μ can be defined as the value function for corresponding flat policy. This means we can apply off-the-shelf RL methods in HRL using different time-scales.

3 Hierarchical Policy Management

We propose a multi-domain dialogue system with a pre-imposed hierarchy that uses the option framework for learning an optimal policy. The user starts a conversation in one of the *master* domains and switches to the other domains (having satisfied his/her goal) that are seen by the model as *sub-domains*. To model individual policies, we can use any RL algorithm. In separate work, we found deep RL models performing worse in noisy environment (Su et al., 2017). Thus, we employ the GPSARSA model from section 2 which proves to handle efficiently noise in the environment. The system is trained from scratch where the system has to learn appropriate policy using both primitive and temporally extended actions.

Algorithm 1 Hierarchical GPRL

Initialize dictionary sets $\mathcal{D}_{\mathcal{M}}, \mathcal{D}_{\mathcal{S}}$ and policies $\pi_{\mathcal{M}}, \pi_{\mathcal{S}}$ for master and sub-domains accordingly
for episode=1:N **do**
  Start dialogue and obtain initial state $\mathbf{b}$
  while $\mathbf{b}$ is not terminal **do**
    Choose action a according to π_m
    if a is primitive **then**
      Execute a and obtain next state $\mathbf{b}'$
      Obtain extrinsic reward r_e
    else
      Switch to chosen sub-domain
      while $\mathbf{b}$ is not terminal **or** a terminates **do**
        Choose action a according to π_s
        Obtain next state $\mathbf{b}'$
        Obtain intrinsic reward r_i
        Store transition in $\mathcal{D}_s$
        $\mathbf{b} \leftarrow \mathbf{b}'$
    Store transition in $\mathcal{D}_m$
    $\mathbf{b} \leftarrow \mathbf{b}'$
  Update parameters with $\mathcal{D}_m, \mathcal{D}_s$

We consider two task-oriented master domains providing restaurant and hotel information for the Cambridge (UK) area. Having found the desired entity, the user can then book it for a specified amount of time or pay for it. The two domains have a set of primitive actions (such as *request, confirm* or *inform* (Ultes et al., 2017)) and a set of composite actions (e.g., *book*, *pay*) which call sub-domains shared between them.

The `Booking` and `Payment` domains were created in a similar fashion: the user wants to reserve a table in a restaurant or a room in a hotel for a specific amount of money or duration of time. The system's role is to determine whether it is possible to make the requested booking. The sub-domains operates only on primitive actions and it's learnt following standard RL framework.

Figure 1 shows the analysed architecture: the `Booking` and `Payment` tasks/sub-domains are shared between two master domains. This means we can train general policies for those sub-tasks that adapt to the current dialogue given the information passed to them by the master domains.

Learning proceeds on two different time-scales. Following (Dietterich, 2000; Kulkarni et al., 2016), we use pseudo-rewards to train sub-domains using an internal critic which assesses whether the sub-goal has been reached.

The master domains are trained using the reward signal from the environment. If a one-step option (i.e., a primitive action) is chosen, we ob-

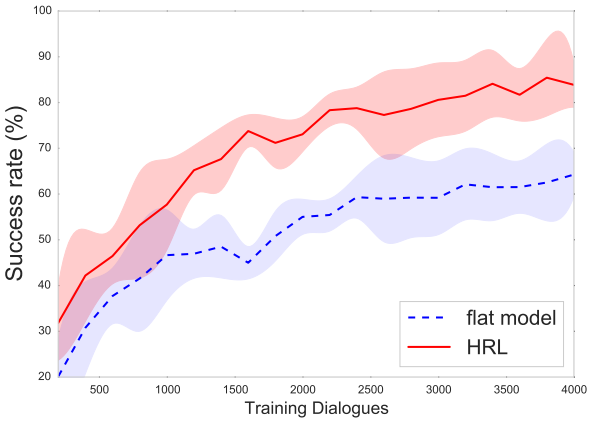

Figure 2: Learning curves for flat and the hierarchical reinforcement learning models.

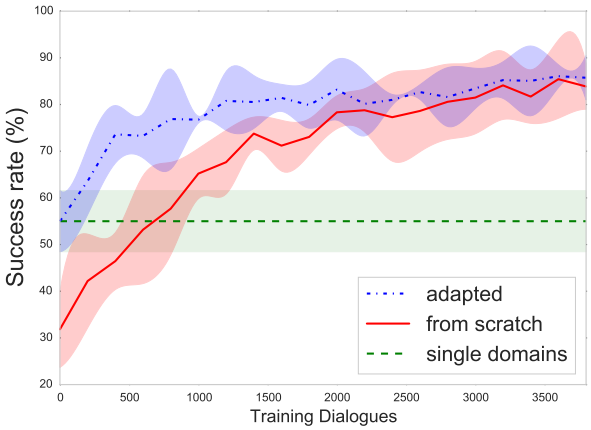

Figure 3: Comparison of policies learnt from scratch and those pre-trained on master domains.

tain immediate extrinsic reward while for the composite actions the master domain waits until the sub-domain terminates and the cumulative reward information is passed back to the master domain. The pseudo-code for the learning algorithm is given in Algorithm 1.

4 Experiments

The PyDial dialogue modelling tool-kit (Ultes et al., 2017) was used to evaluate the proposed architecture. The restaurant domain consists of approximately 100 venues with 3 search constraint slots while the hotel domain has 33 entities with 5 properties. There are 5 slots in the booking domain that the system can ask for while the payment domain has 3 search constraints slots.

In the case of the flat approach, each master domain was combined with the sub-domains, resulting in 11 and 13 requestable slots for the restaurants and hotel domains, respectively.

The input for all models was the full belief state $\mathbf{b}$, which expresses the distribution over the user intents and the requestable slots. The belief state has size 311, 156, 431 and 174 for the restaurants, hotels, booking and payment domains in the hierarchical approach. The flat models have input spaces of sizes 490 and 333 for the restaurant and hotel domains accordingly.

The proposed models were evaluated with an agenda-based simulated user (Schatzmann et al., 2006) where the user intent was perfectly captured in the dialogue belief state. For both intrinsic and extrinsic evaluation, the total return of each dialogue was set to $\mathbb{1}(\mathcal{D}) * 20 - T$, where T is the dialogue length and $\mathbb{1}(\mathcal{D})$ is the success indicator for dialogue $\mathcal{D}$. Maximum dialogue length was set to 30 in both hierarchical and flat model scenarios with $\gamma = 0.99$.

At the beginning of each dialogue, the master domain is chosen randomly and the user is given a goal which consists of finding an entity and either booking it (for a specific date) or paying for it. The user was allowed to change the goal with a small probability and could not proceed with the sub-domains before achieving the master domain goal.

4.1 Hierarchical versus the Flat Approach

Following (Dietterich, 2000; Kulkarni et al., 2016), we apply a more exploratory policy in the case of master domains, allowing greater flexibility in managing primitive and composite actions during the initial learning stages. Figure 2 presents the results with 4000 training dialogues, where the policy was evaluated after each 200 dialogues.

The results validate the option framework: it learns faster and leads to a better final policy than the flat approach. The flat model did overcome the cold start problem but it could not match the performance of the hierarchical model. The policies learnt for sub-tasks with the flat approach perform only 10% worse (on average) than in the hierarchical case. However, providing the entity in both master domains has around 20% lower success rate compared to HRL.

Moreover, the flat model was not able to match the performance of the HRL approach even with more training dialogues. We let it run for another 6000 dialogues and did not observe any improvements in success rate (not reported here). This confirms the findings from other RL tasks - the flat approach is not able to remember successful strategies across different tasks (Peng et al., 2017;

Duan et al., 2016). An example of two successful dialogues for both models is presented in the Figure 4.

4.2 Adaptation of Pretrained Policies

Following the idea of curriculum learning (Bengio et al., 2009), we test the adaptation capabilities of pre-trained policies to more complex situations. Adaptation has proven to be an effective way of reusing existing dialogue policies in new domains (Gašić et al., 2014). Since the kernel function is factored into the kernel for the belief state space and the action space, we can consider them separately. Following (Gašić et al., 2014) the action kernel function is defined only on actions that appear both in original and extended sets and defined 0 otherwise. The kernel for the belief state space is not changed as we operate on the same belief space.

We first train both master domains (without subgoals) until robust policies are learned. Subsequently, both master domains are re-trained in a hierarchical manner for 4000 dialogues (testing after each 200). Figure 3 shows the results compared to the policy learnt from scratch. Both policies trained on independent domains were able to adapt to more complicated tasks very quickly using the hierarchical framework with new options. This confirms that our approach can substantially speed up learning time by training a policy in a supervised way with the available data and then adapting it to more complex multi-task conversations.

5 Conclusion and Future Work

This paper introduced a hierarchical policy management model for learning dialogue policies which operate over composite tasks. The proposed model uses hierarchical reinforcement learning with the Gaussian Process as the function approximator. Our evaluation showed that our model learns substantially faster and achieves better performance than standard (flat) RL models. The natural next step towards the generalisation of this approach is to deepen the hierarchy and apply to more complex tasks.

Acknowledgments

Paweł Budzianowski is supported by EPSRC Council and Toshiba Research Europe Ltd, Cambridge Research Laboratory.

constraint	slot
pricerange	moderate
kind	guesthouse
stars	don't care
hasparking	no
hour	4 pm
peopleNo	4
durationDays	4
day	don't care
requestable slots	
name of hotel	
price of hotel	

S:	Hello, welcome to the Cambridge Multi-Domain dialogue system. How may I help you? **[master domain]**
U:	Hello.
S:	Do you want a hotel or a guesthouse?
U:	I'm looking for a guesthouse in a moderate pricerange. I don't care about stars and don't need a parking access.
S:	A and B guest house is in a moderate pricerange and has 4 stars.
U:	How much does it cost?
S:	Cheapest single room is 50 pounds and a cheapest double room is 70 pounds .
U:	Ok. Can you book it for 4 people?
S:	book() **[option action]**
S:	What time are you planning to check in? **[sub-domain]**
U:	4 pm.
S:	How long do you want to stay?
U:	4 days anytime soon.
S:	You want to book it for 4 people, is that right?
U:	Yes.
S:	I confirm a reservation for 4 days on Saturday for 4 people. Reservation number is 8.
U:	Bye.
S:	Goodbye. **[master domain]**

S:	Hello, welcome to the Cambridge Multi-Domain dialogue system. How may I help you?
U:	I am looking for a guesthouse.
S:	How many stars should the place have? From 0 to 4.
U:	I don't need a parking access.
S:	How many stars should the place have? From 0 to 4.
U:	I don't care but the price should be moderate.
S:	A and B guest house is in a moderate pricerange and has 4 stars.
U:	How much does it cost?
S:	Cheapest single room is 50 pounds and a cheapest double room is 70 pounds.
U:	Ok. Can you book it for 4 people anytime soon?
S:	It doesn't have to have a parking, right? **[wrong domain]**
U:	Yes.
S:	For what day should I make a reservation?
U:	I don't care.
S:	You are looking for a 4 stars hotel right? **[wrong domain]**
U:	Yes.
S:	Would you like the place to have a parking? **[wrong domain]**
U:	No.
S:	How many days do you want to stay there?
U:	4.
S:	I confirm a reservation for 4 days on Saturday. Reservation number is 10.
U:	Bye.
S:	Goodbye.

Figure 4: An example dialogue with the same user goal (top) handled by HRL (middle) and flat (bottom) models.

References

P.-L. Bacon, J. Harb, and D. Precup. 2017. The Option-Critic Architecture. *31AAAI Conference On Artificial Intelligence* .

Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems* 13(4):341–379.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, pages 41–48.

Heriberto Cuayáhuitl. 2009. Hierarchical reinforcement learning for spoken dialogue systems. *PhD Thesis, University of Edinburgh* .

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2010. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech & Language* 24(2):395–429.

Heriberto Cuayáhuitl, Seunghak Yu, Ashley Williamson, and Jacob Carse. 2016. Deep reinforcement learning for multi-domain dialogue systems. *NIPS Workkshop* .

Thomas G Dietterich. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)* 13:227–303.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of The 33rd International Conference on Machine Learning*. pages 1329–1338.

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *Proc of SigDial* .

Richard E Fikes, Peter E Hart, and Nils J Nilsson. 1972. Learning and executing generalized robot plans. *Artificial intelligence* 3:251–288.

Milica Gašić, Filip Jurcicek, Blaise. Thomson, Kai Yu, and Steve Young. 2011. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *IEEE ASRU*.

Milica Gašić, Dongho Kim, Pirros Tsiakoulis, Catherine Breslin, Matthew Henderson, Martin Szummer, Blaise Thomson, and Steve Young. 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains .

Milica Gašić, Nikola Mrkšić, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Policy committee for adaptation in multi-domain spoken dialogue systems. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, pages 806–812.

Milica Gašić and Steve Young. 2014. Gaussian processes for pomdp-based dialogue manager optimization. *TASLP* 22(1):28–40.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*. pages 3675–3683.

John E Laird, Paul S Rosenbloom, and Allen Newell. 1986. Chunking in soar: The anatomy of a general learning mechanism. *Machine learning* 1(1):11–46.

Pierre Lison. 2011. Multi-policy dialogue management. In *Proceedings of the SIGDIAL 2011 Conference*. Association for Computational Linguistics, pages 294–300.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain Dialog State Tracking using Recurrent Neural Networks. In *Proceedings of ACL*.

Ronald Parr and Stuart J Russell. 1998. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems*. pages 1043–1049.

B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, and K.-F. Wong. 2017. Composite Task-Completion Dialogue System via Hierarchical Deep Reinforcement Learning. *ArXiv e-prints* .

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review* 21(02):97–126.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gašić, and Steve J. Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the SIGDIAL 2017 Conference*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689* .

Richard S. Sutton and Andrew G. Barto. 1999. *Reinforcement Learning: An Introduction*. MIT Press.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 1999a. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*. volume 99.

Richard S Sutton, Doina Precup, and Satinder Singh. 1999b. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve J. Young. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *ACL Demo*. Association of Computational Linguistics.

Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. 2014. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. pages 57–67.

Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language* 21(2):393–422.

Jason D Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269* .

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