Single-model Multi-domain Dialogue Management with Deep Learning

Alexandros Papangelis and Yannis Stylianou

Abstract We present a Deep Learning approach to dialogue management for multiple domains. Instead of training multiple models (e.g. one for each domain), we train a single domain-independent policy network that is applicable to virtually any information-seeking domain. We use the Deep Q-Network algorithm to train our dialogue policy, and evaluate against simulated and paid human users. The results show that our algorithm outperforms previous approaches while being more practical and scalable.

1 Introduction

With the proliferation of intelligent agents, assisting us on various daily tasks (customer support, information seeking, hotel booking, etc.), several challenges of Spoken Dialogue Systems (SDS) become increasingly important. Following the successful application of statistical methods for SDS - casting the dialogue problem as a Partially Observable Markov Decision Process (POMDP) and applying reinforcement learning (RL) algorithms - several approaches have been proposed to tackle these challenges as well as reduce the development effort. One such challenge is the ability of an SDS to converse about multiple topics or domains.

Statistical Dialogue Management. POMDPs have been preferred in dialogue management due to their ability to handle uncertainty, which is inherent in human communication. A POMDP Dialogue Manager (DM) typically receives an n-best list of language understanding hypotheses, which are used to update the belief state (reflecting an estimate of the user's goals). Using RL, the system selects a response that maximises the long-term return of the system. This response is typically selected from an abstract action space and has to be converted to text through language

Alexandros Papangelis, Toshiba Research Europe, Cambridge, UK,
Yannis Stylianou, Toshiba Research Europe, Cambridge, UK, and University of Crete, Greece
e-mail: {alex.papangelis,yannis.stylianou}@crl.toshiba.co.uk

generation. Concretely, a POMDP is defined as a tuple $\{S, A, T, O, \Omega, R, \gamma\}$, where S is the state space, A is the action space, $T : S \times A \rightarrow S$ is the transition function, $O : S \times A \rightarrow \Omega$ is the observation function, Ω is a set of observations, $R : S \times A \rightarrow \Re$ is the reward function and $\gamma \in [0,1]$ is a discount factor of the expected cumulative rewards $J = E[\sum_t \gamma^t R(s_t, a_t)]$. A policy $\pi : S \rightarrow A$ dictates which action to take from each state. An optimal policy $\pi^\star$ selects an action that maximises the expected returns of the POMDP, J. Learning in RL consists exactly of finding such optimal policies; however, due to state-action space dimensionality, approximation methods need to be used for practical applications. One such method is to perform learning on summary spaces [1, e.g.].

Relevant Work. Recently, Gašić et al. [2, 3] proposed the use of a hierarchical structure to train generic dialogue policies that can then be refined when in-domain data become available. A Bayesian Committee Machine (BCM) over multiple dialogue policies (each trained on one domain) decides which policy can better handle the user's utterance, and delegates control to that policy. Using this structure, the system can deal with multi-domain dialogues. In order to adapt to new speakers with dysarthria, Casanueva et al. [4] explore ways of transferring data from known speakers, to improve the cold-start performance of a SDS. In particular, when a new speaker interacts with the system, they propose a way to select data from speakers that are similar to the new speaker, and weigh them appropriately.

In [5], the authors use Deep RL to train a policy network which takes as input noisy text (thus bypassing Spoken Language Understanding) and outputs the system's action. The latter can either be simple (e.g. inform(·)) or composite (e.g. a sub-dialogue $\equiv$ domain). The internal structure of the model is a network of Deep-Q policy networks, each of which learns a dialogue policy for a given domain, plus one network for general dialogues (e.g. greetings etc.). The authors train Naive Bayes classifiers to identify valid actions from each dialogue state and they train a Support Vector Machine (SVM) to select the appropriate domain. They evaluate their system on a two-domain information seeking task, for hotels and restaurants. Our work is different in that we train a single Deep-Q network that is able to operate across domains, therefore making it much more scalable, as in Cuayahuitl et al.'s [5] work, it is necessary to add a network for each domain, and the actions in the output are domain-specific.

Other scholars tackle the problem of adapting to new or known users over time, or focus on different parts of the dialogue system (e.g. [6, 7, 8, 9, 10, 11]). Our approach, however, does not rely on complicated transfer learning methods [12, 13] but instead on modelling the generic class of information seeking dialogues by abstracting away from the specifics of each domain. In prior work, Wang et al. [14] proposed a domain-independent summary space (applicable to information-seeking dialogues) onto which a learning algorithm can operate. This allows policies trained on one domain to be transferred to other, unseen domains. In [15], we proposed to apply Wang et. al's domain transfer method to design a multi-domain dialogue manager. We here extend this work by applying Deep Q-Networks and show that this outperforms the previous, GP-SARSA-based multi-domain SDS.

2 Multi-Domain Dialogue Management

Domain Independent Parameterisation (DIP) [14] is a method that maps the (belief) state space into a feature space of size N, that is independent of the particular domain: $\Phi_{DIP}(s,l,a) : S \times L \times A \rightarrow \Re^N, s \in S, l \in L, a \in A$, where L is the set of slots (including a 'null' slot for actions such as *hello*). Φ_{DIP} therefore extracts features for each slot, given the current belief state, and depends on A in order to allow for different parameterisations for different actions. This allows us to define a fixed-size domain-independent space, and policies learned on this space can be used in various domains, in the context of information-seeking dialogues. As shown in [15], we can take advantage of DIP to design efficient multi-domain dialogue managers, the main benefit being that we learn a single, domain-independent policy model that can be applied to information-seeking dialogues. Aiming to further improve the efficiency and scalability of such dialogue managers, we here propose to use a variant of DQN [16] to optimise the multi-domain policy. To this end, we use a two-layer feed-forward network (FFN) to approximate the Q function.

Achieving Domain Independence. To be completely domain-independent, we need to define a generic action space $\mathscr{A}$ for information seeking problems. For our experiments, we include the following system actions: *hello, bye, inform, confirm, select, request, request_more, repeat*. The policy thus operates on the $\Phi_{DIP} \times \mathscr{A}$ space, instead of the original belief-action space. By operating in this parameter space and letting the policy decide which action to take next as well as which slot the action refers to, we achieve independence in terms of both slots and actions, as long as the actions of any domain can be represented as functions of $\mathscr{A} \times L$, where L are the domain's slots including the 'null' slot. The learnt policy therefore decides which action to take by maximising over both the action and the slots:

$$a_{t+1} = argmax_{l,a}\{Q[\Phi_{DIP}(s_t,l,a),a]\} \tag{1}$$

where $a_{t+1} \in \mathscr{A} \times L$ is the selected summary action, s_t is the belief state at time t, and $a \in \mathscr{A}$. To approximate the Q function, we use a 2-layered FFN with 60 and 40 hidden nodes, respectively. The input layer receives the DIP feature vector $\Phi_{DIP}(s,l,a)$ and the output layer is of size $\mathscr{A}$; each output dimension can be interpreted as $Q[\Phi_{DIP}(s,l,a),a]$:

$$\overrightarrow{Q}(\Phi_{DIP}(s_t,l,a)) \approx softmax(W_k^M x_k^{M-1} + b^M) \tag{2}$$

where $\overrightarrow{Q}(\Phi_{DIP}(s_t,l,a))$ is a vector of size $|\mathscr{A}|$, W^m are the weights of the m^{th} layer (out of M layers in total) for nodes k, x_k^m holds the activations of the m^{th} layer, where $x^0 = \Phi_{DIP}(s_t,l,a)$, and b^m are the biases of the m^{th} layer. To generate the summary system action, we simply combine the selected slot and action from equation (1).

The architecture of the system is shown in figure 1. We train the model with DQN with experience replay [16], standardising the input vector of each minibatch to ensure that it follows the same distribution over time. However, we introduce a small bias in the minibatch sampling, towards datapoints with infrequent rewards.

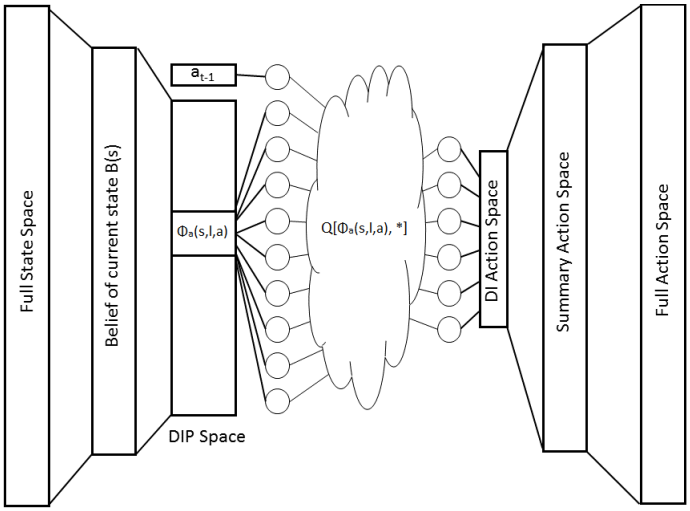

Fig. 1 The architecture of our DNN-based multi-domain dialogue manager. It should be noted that any policy learning algorithm can be used in place of DQN.

In particular, we sample datapoints d from an exponential-like distribution with a small λ value, taking into account the probability of a datapoint to occur in the experience pool, given a reward r, similarly to [17]. If done efficiently, this only introduces a linear factor in the algorithm's time complexity while considerably improving performance and robustness. We used a pool of 1,000 datapoints and a minibatch of 100.

3 Evaluation

Using the PyDial system [18], we trained the DQN-based DM in simulation on four domains: Cambridge Attractions (CA), Cambridge Shops (CS), Cambridge Hotels (CH), and Cambridge Restaurants (CR) using a topic tracker to switch between them, for SLU and NLG purposes. We allowed 1,000 training dialogues (training for one epoch after every ten dialogues) and 1,000 evaluation dialogues while varying the semantic error rate. To assess performance, we compare against a DIP-based DM trained with GPS and a baseline of policies trained with GPS in a single-domain setup (GPS-IND). For the DIP-based algorithms we allow 10 dialogue turns per active domain; this means that the multi-domain dialogues are longer than the single-domain ones, as the active domains at each dialogue and can range from 2 to 4. However, in the multi-domain condition, each domain is active on average in less than 750 dialogues. We therefore train the single-domain policies allowing 40 turns per dialogue, for 750 dialogues.

We then evaluate the multi-domain DMs against a BCM-based DM trained under the same conditions, by conducting a small human user trial, as due to certain restrictions we were not able to conduct crowd-sourced experiments.

Results. Table 1 shows the results of the experiments in simulation, where we varied the semantic error rate, averaged over 10 training/testing runs. We can see that the DQN-based DM outperforms GPS-DIP on multiple domains, as it is more robust to higher error rates (e.g. at the CH domain). Both DIP DMs outperform the baseline in the no-noise condition as they are able to learn more general policies and mitigate effects of harder-to-train domains (e.g. CH). In the presence of noise, GPS-DIP does not seem to cope very well, contrary to DQN-DIP which seems to fare much better in deteriorating conditions even though both algorithms use the same input.

	DQN-DIP				**GPS-DIP**				**GPS-IND**			
Error	0%	15%	30%	45%	0%	15%	30%	45%	0%	15%	30%	45%
CA	95.65	94.2	88.41	79.17	95.5	94.44	77.78	28.17	87.1	78.9	68.7	59.2
CS	95.45	92.96	90.48	76.47	92.59	92	84.62	54.55	89.6	87.2	80.4	71.9
CH	81.25	71.62	64.47	45	88.89	26.92	16.67	10.61	64.6	47.9	35	21.8
CR	92.86	90.62	87.88	71.23	82.61	77.78	61.9	36.11	86.5	78.4	74	59.2
AVG	91.30	87.35	82.81	67.97	89.90	72.79	60.24	32.36	81.95	73.1	64.53	53.03

Table 1 Dialogue Success rates for the three dialogue managers under evaluation.

We conducted a small user trial (30 interactions) comparing the DIP DMs and a BCM DM trained under the same conditions for 1,000 multi-domain dialogues. We asked participants to engage with the SDS in multi-domain dialogues (2 to 4 domains simultaneously active). Success was computed by comparing the retrieved item with the participant's goals for that session. In the end, participants were asked how they would rate the dialogue overall, and had to provide an answer from 1 (very bad) to 5 (excellent). Table 2 shows the results, where we can see that DQN performs very closely to BCM. Even though we can't draw strong conclusions from 30 interactions, it is evident that DQN using a single policy model performs at least as well as BCM, which uses one policy model for each domain. More trials will be conducted in the near future, including the DM proposed in [5].

	DQN-DIP	**GPSARSA-DIP**	**GPSARSA-BCM**
Success	78.6%	59.3%	75%
Rating	3.78	2.67	3.5
Turns/Task	3.25	5.04	4.17

Table 2 Objective dialogue success and subjective dialogue quality rating by participants.

4 Conclusion

We have presented a novel approach to training multi-domain dialogue managers using DNNs. The core idea in this method is to train a single, domain-independent policy network that can be applied to information-seeking dialogues. This is achieved through DIP [14] and as we have shown in this paper, DNNs trained with DQN [16] perform very well. We are currently exploring different DNN architectures and

techniques that extend the present work and can handle large action spaces and multi-modal state spaces. Last, we plan to combine our DM with belief state tracking methods such as [19] or [20], in an effort to move towards end-to-end learning.

References

1. 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.
2. M. Gašić, D. Kim, P. Tsiakoulis, and S. Young, "Distributed dialogue policies for multi-domain statistical dialogue management," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5371–5375.
3. M Gašić, N Mrkšić, PH Su, D Vandyke, TH Wen, and S Young, "Policy committee for adaptation in multi-domain spoken dialogue systems," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 806–812.
4. I Casanueva, T Hain, H Christensen, R Marxer, and P Green, "Knowledge transfer between speakers for personalised dialogue management," in *16th SIGDial*, 2015, p. 12.
5. H Cuayáhuitl, S Yu, A Williamson, and J Carse, "Deep reinforcement learning for multi-domain dialogue systems," *arXiv preprint arXiv:1611.08675*, 2016.
6. S Chandramohan, M Geist, F Lefevre, and O Pietquin, "Co-adaptation in spoken dialogue systems," in *Natural Interaction with Robots, Knowbots and Smartphones*, pp. 343–353. 2014.
7. S Zhu, L Chen, K Sun, D Zheng, and K Yu, "Semantic parser enhancement for dialogue domain extension with little data," in *SLT Workshop*. IEEE, 2014, pp. 336–341.
8. O Lemon, K Georgila, and J Henderson, "Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the talk towninfo evaluation," in *2006 IEEE Spoken Language Technology Workshop*. IEEE, 2006, pp. 178–181.
9. A Margolis, K Livescu, and M Ostendorf, "Domain adaptation with unlabeled data for dialog act tagging," in *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*. ACL, 2010, pp. 45–52.
10. G Tur, U Guz, and D Hakkani-Tur, "Model adaptation for dialog act tagging," in *2006 IEEE Spoken Language Technology Workshop*. IEEE, 2006, pp. 94–97.
11. MA Walker, A Stent, F Mairesse, and R Prasad, "Individual and domain adaptation in sentence planning for dialogue," *Journal of Artificial Intelligence Research*, vol. 30, pp. 413–456, 2007.
12. RS Sutton and AG Barto, *Reinforcement learning: An introduction*, MIT press, 1998.
13. M E Taylor and P Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633–1685, 2009.
14. Z. Wang, T.H. Wen, P.H. Su, and Y. Stylianou, "Learning domain-independent dialogue policies via ontology parameterisation," in *16th Annual Meeting of the SIGDial*, 2015, p. 412.
15. A Papangelis and Y Stylianou, "Multi-domain spoken dialogue systems using domain-independent parameterisation," in *Domain Adaptation for Dialogue Agents*, 2016.
16. V Mnih, K Kavukcuoglu, D Silver, AA Rusu, J Veness, MG Bellemare, A Graves, M Riedmiller, A K Fidjeland, G Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
17. Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.
18. S. Ultes, L. Rojas-Barahona, P.H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.H. Wen, M. Gašić, and S. Young, "Pydial: A multi-domain statistical dialogue system toolkit," in *ACL 2017 Demo, Vancouver*. ACL.
19. T Zhao and M Eskenazi, "Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning," *arXiv preprint arXiv:1606.02560*, 2016.
20. S Lee and A Stent, "Task lineages: Dialog state tracking for flexible interaction," in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, p. 11.