Continuously Learning Neural Dialogue Management

Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona,
Stefan Ultes, David Vandyke, Tsung-Hsien Wen and Steve Young
Department of Engineering, University of Cambridge, Cambridge, UK
{phs26, mg436, nm480, lmr46, su259, djv27, thw28, sjy}@cam.ac.uk

Abstract

We describe a two-step approach for dialogue management in task-oriented spoken dialogue systems. A unified neural network framework is proposed to enable the system to first learn by supervision from a set of dialogue data and then continuously improve its behaviour via reinforcement learning, all using gradient-based algorithms on one single model. The experiments demonstrate the supervised model's effectiveness in the corpus-based evaluation, with user simulation, and with paid human subjects. The use of reinforcement learning further improves the model's performance in both interactive settings, especially under higher-noise conditions.

1 Introduction

Developing a robust Spoken Dialogue System (SDS) traditionally requires a substantial amount of hand-crafted rules combined with various statistical components. In a task-oriented SDS, teaching a system how to respond appropriately is non-trivial. More recently, this *dialogue management* task has been formulated as a reinforcement learning (RL) problem which can be automatically optimised through human interaction (Levin and Pieraccini, 1997; Roy et al., 2000; Williams and Young, 2007; Jurčíček et al., 2011; Young et al., 2013). In this framework, the system learns by a *trial and error* process governed by a potentially delayed learning objective, a *reward function*, that determines dialogue success (El Asri et al., 2014; Su et al., 2015; Vandyke et al., 2015; Su et al., 2016). To enable the system to be trained on-line, sample-efficient learning algorithms have been proposed (Gašić and Young, 2014; Daubigney et al., 2014) which can learn policies from a minimal number of dialogues. However, even with such methods, performance is still poor in the early training stages, and this can impact negatively on the user experience. For these and other reasons, most commercial systems still hand-craft the dialogue management to ensure its stability.

Supervised learning (SL) has also been used in dialogue research where a policy is trained to produce an example response given the dialogue state. Wizard-of-Oz (WoZ) methods (Kelley, 1984; Dahlbäck et al., 1993) have been widely used for collecting domain-specific training corpora. Recently an emerging line of research has focused on training a network-based dialogue model, mostly in text-input schemes (Vinyals and Le, 2015; Serban et al., 2015; Wen et al., 2016; Bordes and Weston, 2016). These systems were directly trained on past dialogues without detailed specification of the internal dialogue state. However, there are two key limitations of using the SL approach for SDS. Firstly, the effects of selecting an action on the future course of the dialogue are not considered. Secondly, there may be a very large number of dialogue states for which an appropriate response must be generated. Hence, the SL training set may lack sufficient coverage. Another issue is that there is no reason to suppose a human wizard is acting optimally, especially at high noise levels. These problems exacerbate in larger domains where multi-step planning is needed. Thus, learning to mimic a human wizard does not necessary lead to optimal behaviour.

To get the best of both SL- and RL-based dialogue

management, this paper describes a network-based model which is initially trained with a supervised spoken dialogue dataset. Since the training data may be mismatched to the deployment environment, the model is further improved by RL in interaction with a simulated user or human users. The advantage of the proposed framework is that a single model can be trained using both SL and RL without modifying the system architecture. This resembles the training process used in AlphaGo (Silver et al., 2016) for the game of Go. In addition, unlike most of the-state-of-the-art RL-based dialogue systems (Gašić and Young, 2014; Cuayáhuitl et al., 2015) which operate on a constrained set of *summary* actions to limit the policy space and minimise expensive training costs, our model operates on a full action set.

2 Neural Dialogue Management

The proposed framework addresses the dialogue management component in a modular SDS. As depicted in Figure 1, the input to the model is the belief state s which encodes the understood user intents along with the dialogue history (Henderson et al., 2014b; Mrkšić et al., 2015), and the output is the master dialogue action a that decides the semantic reply. This is subsequently passed to the natural language generator (Wen et al., 2015).

Dialogue management is represented as a **Policy Network**, a neural network with one hidden layer exploiting *tanh* non-linearities, an output layer consisting of two softmax partitions and six sigmoid partitions. For the softmax outputs, one is for predicting ***DiaAct***, a multi-class label over five dialogue acts: {`request`, `offer`, `confirm`, `select`, `bye`}, and the other for predicting ***Query***, containing four options for the search constraint: {`food`, `pricerange`, `area`, `none`}. ***Query*** options only matter if the dialogue act in {`request`, `confirm`, `select`} is used. The sigmoid partitions are for ***Offer***, each of which is used to determine a binary prediction when making system offer[1].

Given the system's understanding of the user, the model's role is to determine *what* the intent of the system response should be and which *slot* to talk about. The exact *value* in each slot is decided by a separate database parser, where the query is the top prediction of each user-informable slot[2] from the dialogue state tracker and the output is a matched entity. This output forms the system's semantic reply, the *master dialogue action*.

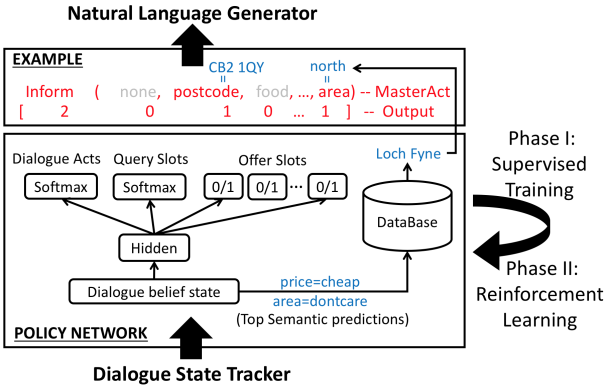

Figure 1: Network-based System Architecture.

2.1 Phase I: Supervised Learning

In the first phase, the policy network is trained on corpus data. This data may come from WoZ collection or from interactions between users and an already existing policy, such as a hand-crafted baseline. The objective here is to 'mimic' the response behaviour within the supervised data.

The training objective for each sample is to minimise a joint cross-entropy loss $\mathcal{L}(\theta)$ between model action labels y defined in §2 and predictions p:

$$\mathcal{L}(\theta) = \sum_{k \in \{d_a, q, O_s\}} H(y_k, p_k), \quad (1)$$

where DiaAct d_a and Query q outputs are categorical distributions, and the Offer set O_s contains six binary offer slots. θ are the network parameters.

2.2 Phase II: Reinforcement Learning

The policy trained in phase I on a fixed dataset may not generalise well. In spoken dialogue, the noise level may vary across conditions and thus can significantly affect performance. Hence, the second phase of the training pipeline aims at improving the SL trained policy network by further training using policy-gradient based RL. The model is given the freedom to select any combination of

[1] System-offer slots are slots the system can mention, such as area, phone number and postcode.

[2] User-informable slots are slots used by the user to constrain the search, such as area and price range.

master action. The training objective is to find a parametrised policy π_θ that maximises the expected reward $J(\theta)$ of a dialogue with T turns: $J(\theta) = E\left[\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \middle| \pi_\theta\right]$, where γ is the discount factor and $r(s_t, a_t)$ is the reward when taking master action a_t in dialogue state s_t. Note that the structure and initial weights of π_θ are fixed by the SL pre-training phase, since the RL training aims to improve the SL trained model.

Here a batch algorithm is adopted, and all transitions are sampled under the current policy. At each update iteration, N episodes were generated, where the ith episode consists of a set of transition tuples $\{(s_t^i, a_t^i, r_t^i)\}_{t=0}^{T_i}$. The estimated gradient is estimated using the likelihood ratio trick:

$$\nabla_\theta J(\theta) = \frac{1}{NT_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i} \nabla_\theta \log \pi(a_t^i | s_t^i; \theta) R_t^i, \quad (2)$$

where $R_t^i = \sum_{t'=t}^{T_i} \gamma^{t'-t} r_{t'}^i$ is the cumulative return from time-step t to T_i. Gradient descent is, however, slow and has poor convergence properties.

Natural gradient (Amari, 1998) improves upon the above 'vanilla' gradient method by computing an ascent direction that approximately ensures a small change in the policy distribution. This direction is $w = F(\theta)^{-1} \nabla_\theta J(\theta)$, where $F(\theta)$ is the Fisher information matrix (FIM). Based on this, Peters and Schaal (2006) developed the Natural Actor-Critic (NAC). In its episodic case (eNAC), the FIM does not need to be explicitly computed to obtain the natural gradient w. eNAC uses a least square method:

$$R_n = \left[\sum_{t=0}^{T} \nabla_\theta \log \pi(a_t^i | s_t^i; \theta)^T\right] \cdot w + C, \quad (3)$$

where C is a constant and $\forall n \in \{1, ..., N\}$ an analytical solution can be obtained. For larger models with more parameters, a truncated variant (Schulman et al., 2015) can also be used to practically calculate the natural gradient.

Experience replay (Lin, 1992) is utilised to randomly sample mini-batches of experiences from a reply pool $\mathcal{P}$. This increases data efficiency by reusing experience samples in multiple updates and reduces the data correlation. As the gradient is highly correlated with the return R, to ensure stable training, a unity-based *reward normalisation* is adopted to normalise the total return R_n between 0 and 1.

3 Experimental Results

The target application is a live telephone-based SDS providing restaurant information for the Cambridge (UK) area. The domain consists of approximately 150 venues, each having 6 slots out of which 3 can be used by the system to constrain the search (food-type, area and price-range) and 3 are informable properties (phone-number, address and postcode) available once a database entity has been found.

The model was implemented using the Theano library (Theano Development Team, 2016). The size of the hidden layer was set to 32 and all the weights were randomly initialised between -0.1 and 0.1.

3.1 Supervised Learning on Corpus Data

A corpus consisting of 720 user dialogues in the Cambridge restaurant domain was split into 4:1:1 for training, validation and testing. This corpus was collected via the Amazon Mechanical Turk (AMT) service, where paid subjects interacted through speech with a well-behaved dialogue system as proposed in (Su et al., 2016). The raw data contains the top N speech recognition (ASR) results which were passed to a rule-based semantic decoder and the *focus* belief state tracker (Henderson et al., 2014a) to obtain the belief state that serves as the input feature to the proposed policy network. The turn-level labels were tagged according to §2. Adagrad (Duchi et al., 2011) per dialogue was used during backpropagation to train each model based on the objective in Equation 1. To prevent over-fitting, early stopping was applied based on the held-out validation set.

Table 1 shows the weighted F-1 scores computed on the test set for each label. We can clearly see that the model accurately determines the type of reply (DiaAct) and generally provides the right information (Offer). The hypothesised reason for the lower accuracy of ***Query*** is that the SL training data contains robust ASR results and thus the system examples contain more offers and less queries. This can be mitigated with a larger dataset covering more diverse situations, or improved via an RL approach.

Table 1: Model performance based on F-measure.

Output	DiaAct	Query	Offer
F-1	97.73	87.39	92.51

3.2 Policy Network in Simulation

The policy network was tested with a simulated user (Schatzmann et al., 2006) which provides the interaction at the semantic level. As shown in Figure 2, the first grid points labelled 'SL:0' represent the performance of the SL model under various semantic error rates (SER), averaged over 500 dialogues.

The SL model was then further trained using RL at different SERs. As the SL model is already performing well, the exploration parameter ϵ was set to 0.1. The size of the experience replay pool $\mathcal{L}$ was 2,000, and the mini-batch size was set to 32. For each update, natural gradient was calculated by eNAC to update the model weights of size $\sim$2600. The total return given to each dialogue was set to $20 \times \mathbb{1}(\mathcal{D}) - T$, where T is the dialogue turn number and $\mathbb{1}(\mathcal{D})$ is the success indicator for dialogue $\mathcal{D}$. Maximum dialogue length was set to 30 turns. Return normalisation was used to stabilise training.

The success rate of the SL model can be seen to increase for all SERs during 6,000 training dialogues, spreading between 1-8% improvement. Generally speaking, the greatest improvement occurs when the SER is most different to the SL training set, which are the higher SER conditions here. In this case, as the semantic hypotheses were more corrupted, the model learned to double-check more on what the user really wanted. This indicates the model's ability to refine its own behaviour via RL.

3.3 Policy Network with Real Users

Starting from the same SL policy network as in §3.2, the model was improved via RL using human subjects recruited via AMT. The policy network was plugged-in to a modular SDS, comprising the Microsoft's Bing speech recogniser [3], a rule-based semantic decoder, the *focus* belief state tracker, and a template-based natural language generator.

To ensure the dialogue quality, only those dialogues whose objective system check matched with the user rating were considered (Gašić et al., 2013). Based on this, two parallel policies were trained with 200 dialogues. To evaluate the resulting policies, policy learning was disabled and a further 110 dialogues were collected with both the SL only and SL+RL models. The AMT users were asked to rate

[3] www.microsoft.com/cognitive-services/en-us/speech-api.

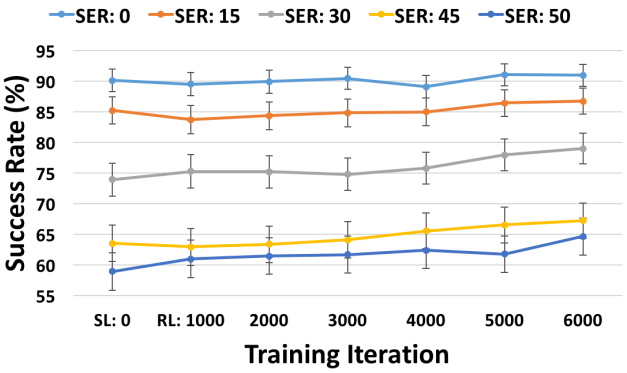

Figure 2: The success rate of the policy network in user simulation under various semantic error rates trained with SL and further improved via RL.

the dialogue quality by answering the question "*Do you think this dialogue was successful?*" on a 6-point Likert scale and also providing a binary rating on dialogue success. The average quality rating (scaled from 0 to 5) is shown in Table 2 with one standard error. The results indicate that the SL-model could work quite well with humans, but was improved by RL on the 200 training dialogues. This demonstrates that on-line RL is a viable approach to adapt a dialogue system to changing environmental conditions.

Table 2: User evaluation on the policies. Quality: 6-point Likert scale, Success: binary rating.

policy	SL	SL+RL
Quality (0-5)	3.97 ± 0.12	4.04 ± 0.12
Success (%)	94.5 ± 2.2	98.2 ± 1.2

4 Conclusion

This paper has presented a two-step development for the dialogue management in SDS using a unified neural network framework, where the model can be trained on a fixed dialogue dataset using SL and subsequently improved via RL through simulated or spoken interactions. The experiments demonstrated the efficiency of the proposed model with only a few hundred supervised dialogue examples. The model was further tested in simulated and real user settings. In a mismatched environment, the model was capable of modifying its behaviour via a delayed reward signal and achieved better success rate.

Acknowledgments

Pei-Hao Su is supported by Cambridge Trust and the Ministry of Education, Taiwan.

References

[Amari1998] Shun-Ichi Amari. 1998. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

[Bordes and Weston2016] Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint: 1605.07683*, May.

[Cuayáhuitl et al.2015] Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. 2015. Strategic dialogue management via deep reinforcement learning. *arXiv preprint arXiv:1511.08099*.

[Dahlbäck et al.1993] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of oz studies: why and how. In *Proc of Intelligent user interfaces*.

[Daubigney et al.2014] Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, and Olivier Pietquin. 2014. A comprehensive reinforcement learning framework for dialogue management optimisation. *Journal of Selected Topics in Signal Processing*, 6(8).

[Duchi et al.2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

[El Asri et al.2014] Layla El Asri, Romain Laroche, and Olivier Pietquin. 2014. Task completion transfer learning for reward inference. In *Proc of MLIS*.

[Gašić and Young2014] Milica Gašić and Steve Young. 2014. Gaussian processes for pomdp-based dialogue manager optimization. *TASLP*, 22(1):28–40.

[Gašić et al.2013] Milica Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve J. Young. 2013. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *Proc of ICASSP*.

[Henderson et al.2014a] M. Henderson, B. Thomson, and J. Williams. 2014a. The Second Dialog State Tracking Challenge. In *Proc of SIGdial*.

[Henderson et al.2014b] M. Henderson, B. Thomson, and S. J. Young. 2014b. Word-based Dialog State Tracking with Recurrent Neural Networks. In *Proc of SIGdial*.

[Jurčíček et al.2011] Filip Jurčíček, Blaise Thomson, and Steve Young. 2011. Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as pomdps. *ACM TSLP*, 7(3):6.

[Kelley1984] John F. Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transaction on Information Systems*.

[Levin and Pieraccini1997] Esther Levin and Roberto Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogue strategies. *Eurospeech*.

[Lin1992] Long-Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321.

[Mrkšić et al.2015] 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 *Proc of ACL*.

[Peters and Schaal2006] Jan Peters and Stefan Schaal. 2006. Policy gradient methods for robotics. In *IEEE RSJ*.

[Roy et al.2000] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. 2000. Spoken dialogue management using probabilistic reasoning. In *Proc of SigDial*.

[Schatzmann et al.2006] 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.

[Schulman et al.2015] John Schulman, Sergey Levine, Philipp Moritz, Michael I Jordan, and Pieter Abbeel. 2015. Trust region policy optimization. *Proc of ICML*.

[Serban et al.2015] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*.

[Silver et al.2016] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

[Su et al.2015] Pei-Hao Su, David Vandyke, Milica Gašić, Dongho Kim, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Learning from real users: Rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems. In *Proc of Interspeech*.

[Su et al.2016] Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proc of ACL*.

[Theano Development Team2016] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[Vandyke et al.2015] David Vandyke, Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialogue success classifiers for policy training. In *ASRU*.

[Vinyals and Le2015] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

[Wen et al.2015] Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, September.

[Wen et al.2016] Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint: 1604.04562*, April.

[Williams and Young2007] Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

[Young et al.2013] Steve Young, Milica Gašic, Blaise Thomson, and Jason Williams. 2013. Pomdp-based statistical spoken dialogue systems: a review. In *Proc of IEEE*, volume 99, pages 1–20.