TAM: Using trainable-action-mask to improve sample-efficiency in reinforcement learning for dialogue systems

Yen-Chen Wu
Cambridge University
ycw30@cam.ac.uk

Bo-Hsiang Tseng
Cambridge University
bht26@cam.ac.uk

Carl Edward Rasmussen
Cambridge University
cer54@cam.ac.uk

Abstract

By interacting with human and learning from reward signals, reinforcement learning is an ideal way to build conversational AI. Considering the expenses of real users' responses, improving sample-efficiency has been the key issue when applying reinforcement learning in real-world dialogue systems. Traditionally, in spoken dialogue systems (SDS), hand-crafted components are embedded with the data-driven parts to accelerate the training process. Action mask, one of these components defined by humans, is used to rule out impossible actions. However, handcrafted action mask can barely be generalized to unseen domains. In this paper, we propose trainable-action-mask (TAM) which learns from data automatically without complicated handcrafted rules. Experiments are conducted in the Cambridge Restaurant domain and TAM is considerably more sample-efficiency than the baseline.

1 Introduction

Model-free deep reinforcement learning (RL) algorithms have been applied in a range of challenging tasks from games [1, 2] to robotic control [3]. Often formulated as a RL problem [4], Spoken Dialogue Systems (SDS) also benefit from it [5]. In a typical modular SDS, user inputs are firstly processed by automatic speech recognition (ASR) and natural language understanding (NLU) modules and then aggregated by a belief state tracker to produce *belief states*, which is the estimate of the user's goal. After the system determines what to convey to the user, this semantic information is passed through natural language generation (NLG) and speech synthesis modules. Between these two ends is the core of the SDS: the dialogue manager.

With high-capacity function approximators of deep RL, the dialogue manager through exploration can learn how to interact with users. It estimates an appropriate mapping from the belief state to the policy or the state-action values Q. However, the widespread adoption of such methods in real-world dialogue systems has faced two major challenges. First, model-free deep RL methods are notoriously expensive in terms of their interactions required. Even relatively simple tasks can require thousands of labelled dialogues of data collection and complex behaviours with multi-domain application might need substantially more. Second, these methods are often brittle with respect to hyperparameters: learning rates, and exploration must be set carefully for different problems to achieve good results [5]. Both of these challenges severely limit the applicability of model-free deep RL to real-world dialogue systems [6, 7, 8].

One of the reasons for the above-mentioned limitations is the lack of knowledge by the agent of dynamics or so-called transition functions of an environment. Lacking the ability to predict future states, the agent wastes time on repeating similar action sequences to get accurate return estimations without generalization among the same transitions. This inefficient exploration not only worsens

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

sample efficiency but also introduces the uncertainty to model-free algorithms because agents cannot obtain positive signals from sparse-rewards environments such as dialogue systems.

To that end, we draw on model-based RL (MBRL) [9, 10, 11, 12, 13]. In MBRL, an *environment model* is used to model state transition; this model predicts future states or rewards given a present state and the next action. One way to utilize the environment model in SDS is Dyna-Q [14, 15] which generates training data for agents and keeps improving its environment model from real interactions between agents and users. This method has achieved some successes in training dialogue systems [16], [17]. Nevertheless, these approaches rely on accurate predictions of future states. Otherwise, the generated noisy data could adversely affect the experience replay buffer and result in convergence toward sub-optimal performance. This problem is even more critical in real-world tasks such as multi-domain dialogue systems where training an accurate environment model is challenging.

In this paper, we propose trainable-action-mask (TAM) to rule out bad actions by a simplified environment model. In experiments in Cambridge Restaurant domain, TAM only requires one third of training data to reach the same success rate (80%) compared with baseline.

2 Preliminaries

In this section, we establish the reinforcement learning notations used throughout this paper and briefly introduce the handcrafted action mask.

2.1 Dialogue management through reinforcement learning

Dialogue management can be cast as a continuous MDP [4] composed of a continuous multivariate belief state space B, a finite set of actions A and a reward function $R(b_t, a_t)$. The belief state b is a probability distribution over all possible (discrete) states. At a given time t, the agent (policy) observes the belief state $b_t \in B$ and executes an action $a_t \in A$. The agent then receives a reward $r_t \in R$ drawn from $R(b_t, a_t)$. The policy π is defined as a function $\pi : B \times A \rightarrow [0, 1]$ that with probability $\pi(b, a)$ takes an action a in a state b. For any policy π and $b \in B$, the value function V_π corresponding to π is defined as:

$$V^\pi(b) = \mathbb{E}\{r_t + \gamma r_{t+1} + ...|b_t = b, \pi\} \tag{1}$$

where $0 \leq \gamma \leq 1$, is a discount factor and r_t is a one-step reward. The objective of reinforcement learning is to find an optimal policy π^*, i.e. a policy that maximizes the value function in each belief state. Equivalently, we can estimate the unique optimal value function V^* which corresponds to an optimal policy. In both cases, the goal is to find an optimal policy π^* that maximises the discounted total return

$$R = \sum_{t=0}^{T-1} \gamma^t r_t(b_t, a_t) \tag{2}$$

over a dialogue with T turns, where $r_t(b_t, a_t)$ is the reward when taking action a_t in dialogue state b_t at turn t and γ is the discount factor.

2.2 Hand-crafted action mask

Generally speaking, a dialogue system requires thousands of dialogues as training data to converge to satiable performance while doing reinforcement learning. In order to stabilize or accelerate the training process, dialogue systems usually incorporate several heuristic components. One of them is action mask, which filters out the improper action and avoid unnecessary exploration in reinforcement learning. Hand-crafted action mask requires tons of effort to design the rules. For example, we need to check whether each goal is completed ,if not, the action *goodbye* will be filtered. When there is a new domain with different action space, a new action mask is needed. Therefore, hand-crafted action mask is unscalable and impractical in real-world applications.

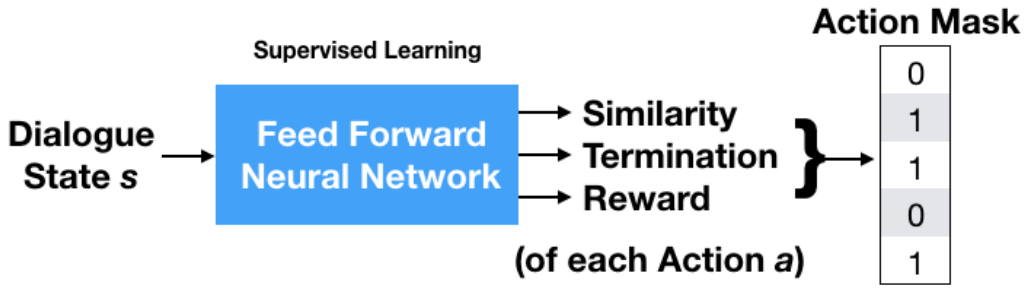

Figure 1: TAM architecture:

3 Trainable-action-mask (TAM)

Since the hand-crafted action mask is hard to generalize to other domain, we propose trainable-action-mask (TAM) to learn the action mask automatically from data. TAM learns to predict several attributes of future states and uses these attributes to decide which actions should be ruled out. TAM belongs to model-based reinforcement learning(MBRL), so we give a brief introduction and point out the difference from other MBRL approaches.

3.1 Model-based reinforcement learning

Model-based RL additionally learns an *environment model* to predict the transition between states, while model-free RL only learns the mapping of the value function. Traditionally, an *environment model* is used to generate pseudo training data or roll-out future steps. However, the imperfect environment model which predicts completely wrong future states causes devastating unstable training problem. TAM avoids this problem by using a simplified version of the environment model. TAM does not predict the exact next state but only predicts three signals instead. This makes the environment model relatively easy to train and debug[1]. (See Figure 1.)

These three signals are: similarity, termination, and reward.

- **similarity**: a real value between 0 and 1, which is the cosine-similarity between the current state and the next state after taking an action a
- **termination**: a binary value, which indicates the dialogue terminates or not after taking an action a
- **reward**: a real value between 0 and 1, which is the normalized immediate reward after taking an action a

Based on these learned signals, we filter out the unwanted actions by defining two masks – termination action mask and useless action mask.

3.2 Termination action mask

In a state s, if an action a leads to the termination of the dialogue and dialogue fails, this action a should be filtered. For example, saying *goodbye* before task completion always leads to the termination and failure of the dialogue. In a state s, if an action a is predicted to terminate the dialogue with a negative immediate reward r, this action a would be ruled out.

3.3 Useless action mask

In a state s, if the next state s' does not change after taking an action a, the action a should be filtered. In other words, there is no information gain after taking action a. For example, if an agent already knows which *area* is requested by user, the agent should not ask the question regarding *area* again.

[1] All the loss functions are MSE-loss, and losses are back-propagated by Adam optimizer.

In our approach, we set a threshold δ to determine whether consecutive states remains unchanged. If the predicted similarity between two states is lager than δ[2], the action a would be classified as an *useless action* and ruled out. Another implementation is to directly predict a binary value which indicates the similarity is higher or lower than δ (the state s is changed or unchanged after taking action a). These two implementations are compared in Section 4.

4 Experiments and Results

Experiments are conducted on the Cambridge restaurant domain from the PyDial toolkit [18] with a goal-driven user simulator on the semantic level [19, 20]. The training process is broken down into 20 milestones, and each milestone contains 200 dialogues. For each milestone, the current policy network is tested on 500 dialogues. We run the evaluation 10 times and average the results to reduce variance arising from different random initialization.

User simulator A user simulator replicates user behaviour with sufficient accuracy to optimize model parameters to an acceptable level of performance [4] and is more cost-effective for development and evaluation purposes. We use an error model where confusions to the simulated user input are added. The error model outputs an N-best list of possible user responses. The input for all models is the full dialogue belief state b of size 268 and the output action space consists of 16 possible actions. The maximum dialogue length was set to 25 turns and γ was 0.99. The reward is defined as 20 for a successful dialogue minus the number of turns in the dialogue. In order to accommodate for ASR error, we include 15% semantic error rate (SER) in the user simulator.

Implementation Details For NN-based algorithms, the size of a minibatch, on which the training step is performed, is 64. ϵ-greedy exploration is used, with ϵ linearly reducing from 0.3 down to 0 over the training process. The Adam optimiser was used with an initial learning rate of 0.001 [21]. For algorithms employing experience replay, the replay memory has a capacity of 2000 interactions.

Baseline We use ACER as our model-free reinforcement learning algorithm. Hand-crafted mask serves as the oracle of having a carefully designed mask.

4.1 Comparison with baselines

In Figures 2, we compare the percentage of successful dialogues (the left figure) and the average number of turns in a dialogue (the right one).

Success rate TAM learns faster and converges to a higher success rate than the baseline while still not as good as the oracle. Hand-crafted rules are very complicated and it achieves around 85% of success rate with 200 training dialogues(while baseline uses four times of training data). This give us a great expectation of improving learned action mask.

Average turns per dialogue In terms of average turn per dialogue, TAM performs similar to baseline model and somewhat unstable. We think it is because termination mask prevent agent from stopping exploration, which produces lengthy dialogues. Oracle dialogues are quite short and efficient even in the beginning of the training process.

4.2 Mask ablation study

In Figures 3, we can see that the best result produced by using both two mask. When we use only termination mask, improvement is marginal. When we use only similarity mask, the performance is even worse than baseline. That is because when similarity mask blocks out most actions, the probability of taking action that leads to termination increase. The dialogues end soon without collecting useful training data. When we only use termination mask, the lengthy dialogues make reward credit assignment difficult.

The right part is the comparison between different similarity output implementation. We can see that

[2]In experiment setting, δ is 0.97

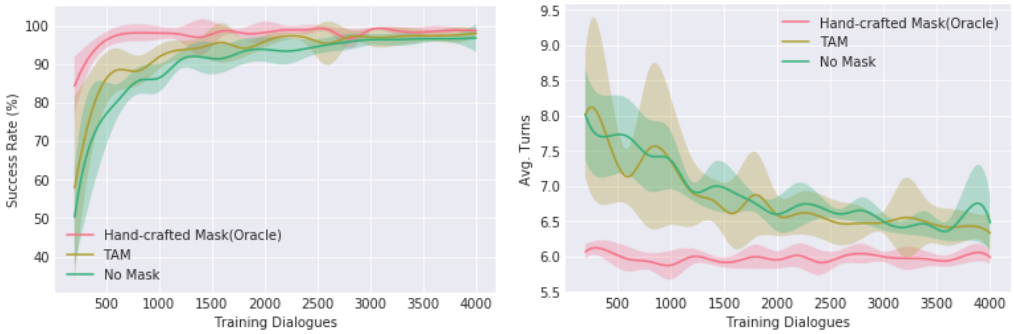

Figure 2: *Left:* Comparison between different update algorithms. *Right:* Experiment on robustness of different architectures to imperfect model.

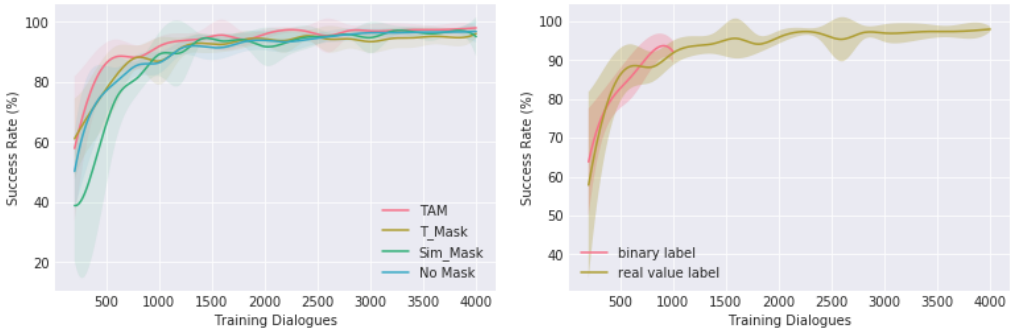

Figure 3: *Left:* Ablation study of different masks. T-Mask uses only termination mask and Sim-Mask uses only similarity mask. *Right:* Different output types of similarity. The red line is the training curve using binary similarity output while the brown line predicts continuous one.

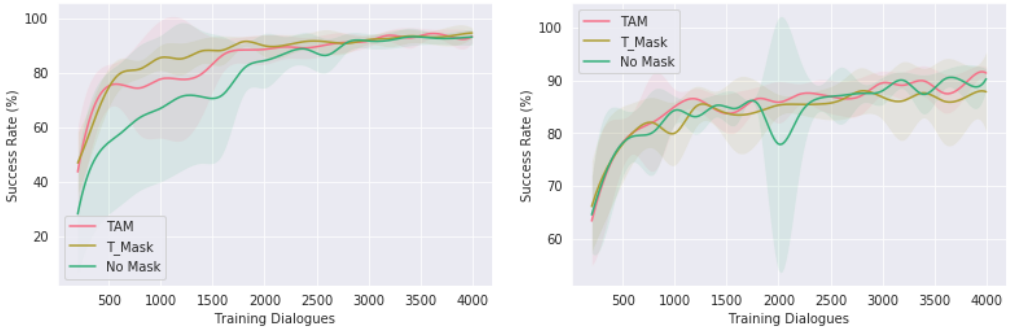

Figure 4: *Left:* Ablation study 15% SER in the US. *Right:* Ablation study 30% SER in the US.

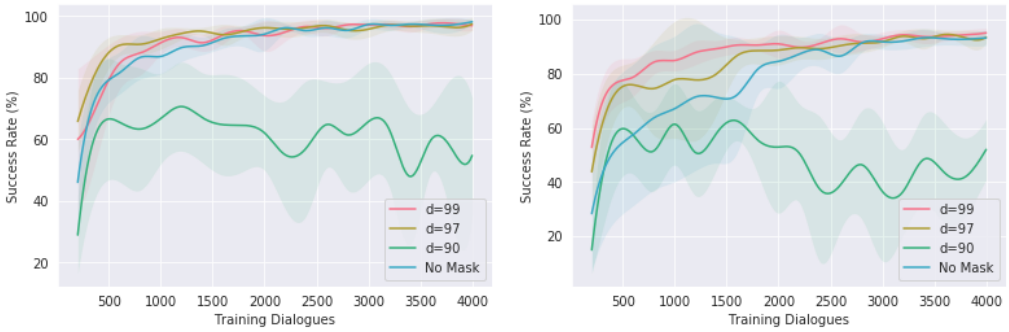

Figure 5: *Left:* Comparison between different similarity threshold δ with 0% semantic error rate(SER) in the user simulator(US). *Right:* Comparison between different similarity threshold δ with 15% SER in the US.

4.3 ablation study: noisy environment

In Figures 4, we compared TAM with baseline and using termination mask only(T-Mask) in two noisy environments. Interestingly, in the environment with 15% semantic error rate (SER), the performance of TAM become a bit unstable than using only termination mask. That is because similarity mask is hard to learn in the noisy environment and the inaccurate prediction of similarity could produce unstable or wrong action mask. Producing unstable action mask is quite harmful. The masked actions are explored less and thus have inaccurate Q-values. When an unstable mask does not filter these action, the wrong value function would ruin the whole policy.

In the right part is a more noisy environment with 30% SER. All of the model suffer from inaccurate state prediction and there inaccurate mask and value. They all performance with limited success rates.

4.4 Comparison of different δ

In Figures 5, we compare different similarity threshold δ $(0.9, 0.97, 0.99)$ in two environments. First, we can notice that when $\delta = 0.9$, it has poor performance and even much worse than baseline in both environments. That is because it blocks too many actions and sometimes blocks the correct or useful one, which also makes the training process very unstable.

Secondly, though both models with $\delta = 0.99$ and $\delta = 0.97$ perform better than baseline. The have different strength in different environments. In the environment without noise, $\delta = 0.97$ learns more efficiently by blocking more actions. In the environment with 15% SER, $\delta = 0.99$ is a better choice since it acts more conservatively and only blocks the action with high confidence. That is because the prediction of similarity become more inaccurate in the noisy environment.

5 Conclusion

TAM make the dialogue agent not only learns more efficiently but also more explainable. Our contributions are:

- Realizing the idea that the action mask can be automatically learned from data, and achieve considerably high sample-efficiency(using only one third of the training data to converge.)
- Proposing a new model-based reinforcement learning algorithm to use simplified environment model that are relatively ease to train. And this algorithm can be generalized to other applications apart from dialogue systems.

We aim at improving both sample-efficiency and stability of policy manager in dialogue systems until it is suitable for online learning: to train an agent safely within an affordable number of interactions with real users.

References

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

[2] 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. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[3] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

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

[5] Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. A benchmarking environment for reinforcement learning based task oriented dialogue management. *arXiv preprint arXiv:1711.11023*, 2017.

[6] Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu, and Steve Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 312–317. IEEE, 2011.

[7] Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016.

[8] Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*, 2016.

[9] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, pages 2154–2162, 2016.

[10] David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. The predictron: End-to-end learning and planning. *arXiv preprint arXiv:1612.08810*, 2016.

[11] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838, 2016.

[12] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[13] Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In *Advances in Neural Information Processing Systems*, pages 6118–6128, 2017.

[14] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pages 216–224. Elsevier, 1990.

[15] Richard S Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael P Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*, 2012.

[16] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*, 2018.

[17] Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442*, 2018.

[18] Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. Pydial: A multi-domain statistical dialogue system toolkit. *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, 2017.

[19] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics, 2007.

[20] Jost Schatzmann and Steve Young. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747, 2009.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.