Safe transfer learning for dialogue applications

Nicolas Carrara, Romain Laroche, Jean-Léon Bouraoui, Tanguy Urvoy, Olivier Pietquin

Safe transfer learning for dialogue applications.

Nicolas Carrara[1,2], Romain Laroche[3], Jean-Léon Bouraoui[2], Tanguy Urvoy[2], and Olivier Pietquin[1]

[1] Univ. Lille, CNRS, Centrale Lille, INRIA UMR 9189 - CRIStAL, Lille, France
name.surname@inria.fr
[2] Orange Labs, Lannion, France
name.surname@orange.com
[3] Microsoft Maluuba, Montral, Canada
name.surname@microsoft.com

Abstract. In this paper, we formulate the hypothesis that the first dialogues with a new user should be handle in a very conservative way, for two reasons : avoid user dropout; gather more successful dialogues to speedup the learning of the asymptotic strategy. To this extend, we propose to transfer a safe strategy to initiate the first dialogues.

Keywords: Transfer Learning · Dialogue · Safety

1 Introduction

During its early steps of learning, a Reinforcement Learning (RL) [8] based dialogue agent does a lot of exploration that may lead to penalizing behavior. However, the first interactions between a user and a dialogue system are crucial to gain his trust. To improve jump-start performance of an RL agent, one can transfer a strategy [9, 7]. In dialogue, the strategy focuses on the success of the dialogue while minimizing its length [3, 5, 2, 4]. Rushing the dialogue may be problematic with some users and induce premature dialogue hangups. In one hand, it could lead to the lose of this user once for all. In an other hand, the lack of succeeding dialogues may affect the learning speed of the RL agent. To this extend, we introduce a novel algorithm : ϵ-safe. It transfers a safe strategy which avoid any critical dialogue act to avoid the aforementioned problems.

2 ϵ-safe

ϵ-safe (Fig 1) is a Q-learning algorithm [6] where each action is decided by a randomly chosen policy among the greedy policy, an exploratory policy and the transferred safe policy. It may be an handcrafted policy, an RL policy (with large reward penalty on the catastrophic event) or even a safe-RL policy [1].

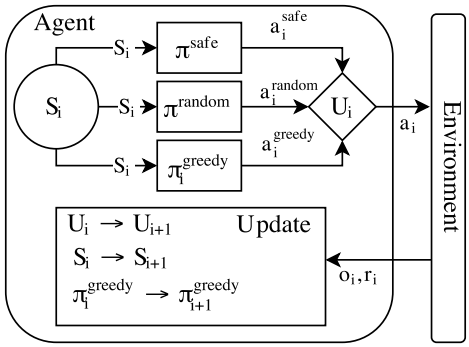


Fig. 1: ϵ-safe algorithm.

3 Experiment

We test our algorithm on a simple slot-filling application. The agent ask for slot values, slot by slot in a fix order. Several acts are available :

- `ask_next` : ask next slot (with NLU errors).
- `repeat_oral` : repeat current slot (with NLU errors).
- `repeat_numpad`: repeat using numeric pad (without NLU errors).
- `summarize_inform`: summarize slots values and return the form result. If values are correct, the dialogue ends successfully, if not, the slot values are reset and the dialogue continues from the first slot.

`repeat_numpad` is an unsafe action : the user hangups with probability p. For each new user, p is randomly generated. An histogram of p values is displayed Fig 2.

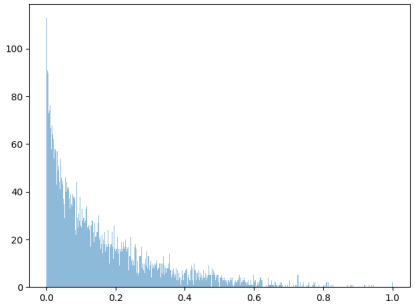
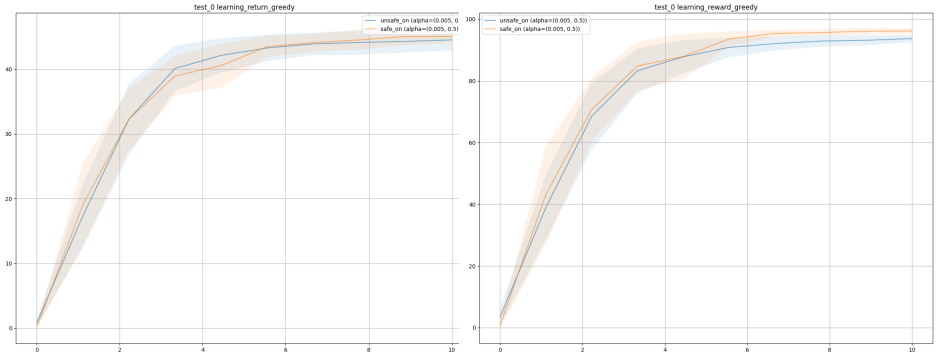


Fig. 2: Half-gaussian distribution of p values.

We define 2 handcrafted dialogue systems. The `safe` system uses `repeat_oral` if the recognition score is bellow 0.5, otherwise `ask_next`, or `summarize_inform` after the last slot; The `unsafe` system uses `repeat_numpad` instead. We compare two ϵ-safe agents. `safe_on` uses `safe` as transferred policy while `unsafe_on` transfers the `unsafe` policy.



(a) Dialogue score (b) Success frequency

Fig. 3: Performance of the greedy policies.

We test `safe_on` and `unsafe_on` with 10 randomly generated users, for 1000 dialogues. We repeat the experiment 10 times. We display the performance of the greedy policies. The dialogue score (reward penalized by dialogue length) is plotted on Fig 3a while the dialogue success (reward only) is plotted on Fig 3b. We see that despite a slightly difference on the dialogue success, the dialogue score is the same. That means ϵ-safe doesn't improve the learning speed of the greedy policy even if it is conservative enough to keep the user in the dialogue by avoiding catastrophic acts.

4 Future work

We believe that the hangup-model is too simple, so we plan to design the hangup-model with a Poisson distribution. We also want to replace the handcrafted policies by actual RL policies learned on source users. Finally, a real application on DSTC2 may be considered.

References

1. Carrara, N., Laroche, R., Bouraoui, J., Urvoy, T., Pietquin, O.: A Fitted-Q Algorithm for Budgeted MDPs. Workshop on Safety, Risk and Uncertainty in Reinforcement Learning (UAI2018) (2018)
2. Carrara, N., Laroche, R., Pietquin, O.: Online learning and transfer for user adaptation in dialogue systems. In: SIGDIAL/SEMDIAL on negotiation dialog (2017)
3. Casanueva, N., Hain, T., Christensen, H., Marxer, R., Green, P.: Knowledge transfer between speakers for personalised dialogue management (2015)
4. Chandramohan, S., Geist, M., Pietquin, O.: Optimizing Spoken Dialogue Management from Data Corpora with Fitted Value Iteration. Proceedings of the International Conference on Speech Communication and Technologies (2010)
5. Genevay, A., Laroche, R.: Transfer learning for user adaptation in spoken dialogue systems. In: AAMAS (2016)
6. J. C. H. Watkins, C., Dayan, P.: Q-learning. In: Machine Learning. pp. 279–292 (1992)
7. Lazaric, A.: Transfer in Reinforcement Learning: a Framework and a Survey (2012)
8. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press Cambridge (1998)
9. Taylor, M., Stone, P.: Transfer Learning for Reinforcement Learning Domains : A Survey. JMLR (2009)