Safe transfer learning for dialogue applications
Nicolas Carrara, Romain Laroche, Jean-Léon Bouraoui, Tanguy Urvoy, Olivier Pietquin

▶ To cite this version:
Nicolas Carrara, Romain Laroche, Jean-Léon Bouraoui, Tanguy Urvoy, Olivier Pietquin. Safe transfer learning for dialogue applications. SLSP 2018 - 6th International Conference on Statistical Language and Speech Processing, Oct 2018, Mons, Belgium. 2018. <hal-01928102>

HAL Id: hal-01928102
https://hal.archives-ouvertes.fr/hal-01928102
Submitted on 20 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Safe transfer learning for dialogue applications.

Nicolas Carrara¹,², Romain Laroche³, Jean-Léon Bouraoui², Tanguy Urvoy², and Olivier Pietquin¹

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 handled in a very conservative way, for two reasons: avoid user dropout; gather more successful dialogues to speed up 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 loss 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: \( \epsilon \)-safe. It transfers a safe strategy which avoid any critical dialogue act to avoid the aforementioned problems.

2 \( \epsilon \)-safe

\( \epsilon \)-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].
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.

![Half-gaussian distribution of $p$ values.](image)

**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 below 0.5, otherwise ask_next, or summarize_inform after the last slot; The unsafe system uses repeat_numpad instead. We compare two $\epsilon$-safe agents. safe_on uses safe as transferred policy while unsafe_on transfers the unsafe policy.

![Graphs showing dialogue score and success frequency](image)

(4) 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 $\epsilon$-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