The Negotiation Dialogue Game

Chapter · December 2017

DOI: 10.1007/978-981-10-2585-3_33

Romain Laroche
Microsoft Maluuba

The Negotiation Dialogue Game

Romain Laroche[1] and Aude Genevay[1]

Orange Labs, Châtillon, France,
romain.laroche@orange.com, aude.genevay@gmail.com

Abstract. This article presents the design of a generic negotiation dialogue game between two or more players. The goal is to reach an agreement, each player having his own preferences over a shared set of options. Several simulated users have been implemented. An MDP policy has been optimised individually with Fitted Q-Iteration for several user instances. Then, the learnt policies have been cross evaluated with other users. Results show strong disparity of inter-user performances. This illustrates the importance of user adaptation in negotiation-based dialogue systems.

Keywords: Spoken dialogue systems · Dialogue simulation · Reinforcement learning · Negotiation · Game theory

1 Introduction

Research on negotiation dialogue experiences a growth of interest. At first, Reinforcement Learning [1], the most popular framework for dialogue management in spoken dialogue systems [2–4], has been applied to negotiation with mitigated results [5, 6], because the non-stationary policy of the opposing player prevents those algorithms from converging consistently. Then, Multi-Agent Reinforcement Learning [7] was applied but still with convergence difficulties [8]. Finally, recently, Stochastic Games [9] were applied successfully [10], with convergence guarantees, but still only for zero-sum games, which is quite restrictive in a dialogue setting where noisy communication and miscommunication are the bases of the game.

In this article, the negotiation dialogue games in the literature ([5] considers sets of furniture, [11, 8] resource trading, and [12–15] appointment scheduling) have been abstracted as an agreement problem over a shared set of options. The goal for the players is to reach an agreement and select an option. This negotiation dialogue game can be parametrised to make it zero-sum, purely cooperative, or general sum.

In addition to the study of negotiation dialogue, we claim that this game can be used for user adaptation in dialogue systems [16, 17], which is not progressing as fast as it should because of lack of data. Indeed, while one used to need only a dataset to learn from, user adaptation requires as many datasets as users in order to learn and evaluate the algorithms. The negotiation game enables to introduce several handcrafted user simulators with a set of parameters. An MDP policy has been individually optimised for five user instances. Then, these policies have been cross evaluated on all users. Results show strong disparity of inter-user performance. This illustrates the importance of user adaptation in negotiation-based dialogue.

2 The Negotiation Dialogue Game

The goal for each participant is to reach an agreement. The game involves a set of m players $\mathscr{P} = \{p^i\}_{i \in [1,m]}$. n options (in resource trading, it is an exchange proposal, in appointment scheduling, it is a time-slot) are considered, and for each option τ, players have a cost $c^i_\tau \sim \delta^i$ randomly sampled from distribution $\delta^i \in \Delta_{\mathbb{R}^+}$ to agree on it. Players also have a utility $\omega^i \in \mathbb{R}^+$ for reaching an agreement. For each player, a parameter of cooperation with other players $\alpha^i \in \mathbb{R}$ is introduced. As a result, player p^i's immediate reward at the end of the dialogue is:

$$R^i(s^i_T) = \omega^i - c^i_\tau + \alpha^i \sum_{j \neq i} (\omega^j - c^j_\tau) \tag{1}$$

where s^i_T is the last state reached by player p^i at the end of the dialogue, τ is the agreed option. If players fail to agree, the final immediate rewards $R^i(s^i_T) = 0$ for all players p^i. If at least one player p^j misunderstands and agrees on a wrong option τ^j which was not the one proposed by the other players, this is even worse, since each player p^i gets the cost of selecting option τ^i without the reward of successfully reaching an agreement:

$$R^i(s^i_T) = -c^i_{\tau^i} - \alpha^i \sum_{j \neq i} c^j_{\tau^j} \tag{2}$$

The values of α^i give a description of the nature of the players, and therefore of the game as modelled in game theory [9]. If $\alpha^i < 0$, player p^i is said to be antagonist: he has an interest in making the other players lose. In particular, if $m = 2$ and $\alpha^1 = \alpha^2 = -1$, it is a zero-sum game. If $\alpha^i = 0$, player p^i is said to be self-centred: he does not care if the other player is winning or losing. Finally, if $\alpha^i > 0$, player p^i is said to be cooperative, and in particular, if $\forall i \in [1, m]$, $\alpha^i = 1$, the game is said to be fully cooperative because $\forall (i,j) \in [1,m]^2$, $R^i(s^i_T) = R^j(s^j_T)$.

From now on, and until the end of the article, we suppose that there are only $m = 2$ players: a system p_s and a user p_u. They act each one in turn, starting randomly by one or the other. They have four possible actions. Accept(τ) means that the user accepts the option τ (independently from the fact that τ has actually been proposed by the other player. If it has not, this induces the use of Equation 2 to determine the reward). This act ends the dialogue. RefProp(τ) means that the user refuses the proposed option and proposes instead option τ. Repeat means that the player asks the other player to repeat his proposition. And finally, EndDial denotes the fact that the player does not want to negotiate anymore, and terminates the dialogue.

Understanding through speech recognition of system $\mathscr{p}_s$ is assumed to be noisy with a sentence error rate SER_s^u after listening to a user $\mathscr{p}_u$: with probability SER_s^u, an error is made, and the system understands a random option instead of the one that was actually pronounced. In order to reflect human-machine dialogue reality, a simulated user always understands what the system says: $SER_u^s = 0$. We adopt the way [18] generates speech recognition confidence scores: $score_{reco} = \frac{1}{1+e^{-X}}$ where $X \sim \mathcal{N}(c, 0.2)$ given a user $\mathscr{p}_u$, two parameters $(c_\perp^u, c_\top^u)$ with $c_\perp^u < c_\top^u$ are defined such that if the player understood the right option, $c = c_\top^u$ otherwise $c = c_\perp^u$. The further apart the normal distribution centres are, the easier it will be for the system to know if it understood the right option, given the score.

3 The Inter-User Policy Experiment

This section intends to show that, in the negotiation game, a policy that is good or optimal against a given user might yield very poor performance against another user. First, it introduces two classes of handcrafted users. Then, it designs a linear parametrisation in order to use Fitted Q-Iteration [19, 20] for policy optimisation. And finally, it shows that policies that have been learnt and optimised on specific users are only marginally successfully reusable on other users.

3.1 User profiles

A straightforward characteristic of a user $\mathscr{p}_u$ is its intelligibility by the system $\mathscr{p}_s$, parametrised by its average sentence error rate SER_s^u. Another understanding characteristic consists in varying centres $(c_\perp^u, c_\top^u)$ for the speech recognition score. For distant $(c_\perp^u, c_\top^u)$ values, the system will easily know if it understood well.

In order to add more variability in our simulated users, two handcrafted classes of users have been implemented:

- The *Deterministic User* (parameter x) $\textsc{Accept}(\tau)$ if and only if $\tau \in \mathscr{T}_x$, where $\mathscr{T}_x$ is the set of its x preferred options. If $\tau \notin \mathscr{T}_x$, he $\textsc{RefProp}(\tau')$, $\tau' \in \mathscr{T}_x$ being his preferred options that was not proposed before. If all $\tau \in \mathscr{T}_x$ have been refused, or if the system insists by proposing the same option twice, he $\textsc{EndDial}$.
- The *Random User* (parameter p) $\textsc{Accept}(\tau)$ any option τ asked by the system, with probability p. With probability $1-p$, he $\textsc{RefProp}(\tau')$ an option τ' randomly. If he's asked to repeat, he'll make a new random proposition.

3.2 Reinforcement Learning implementation

The system $\mathscr{p}_s$ learns the optimal policy with the Fitted Q-Iteration algorithm [19, 20], when playing against user $\mathscr{p}_u$. This subsection details the design of the Reinforcement Learning implementation.

The dialogue system is formalised as an MDP $\langle \mathcal{S}, \mathcal{A}, R, P, \gamma \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $R : \mathcal{S} \to \mathbb{R}$ is the immediate reward function, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the Markovian transition function and γ is the discount factor.

Least-squares Fitted Q-Iteration is used to learn the policy with a linear parametrisation of the Q-function. The optimal Q-function Q^* verifies Bellman's equation:

$$Q^*(s,a) = \mathbb{E}\left[R(s) + \gamma \max_{a'} Q^*(s', a')\right] \Leftrightarrow \quad Q^* = T^* Q^* \tag{3}$$

The optimal Q-function is thus the fixed point of Bellman's operator T^* and since it is a contraction ($\gamma < 1$), Banach's theorem ensures its uniqueness. Hence, the optimal Q-function is obtained by iteratively applying Bellman's operator.

When the state space is continuous (or very large) it is impossible to use Value-Iteration as such. The Q-function must be parametrised. A popular choice is the linear parametrisation of the Q-function [20]: $Q_a(s) = \theta_a^\top \Phi_a(s)$ where $\Phi = \{\Phi_a\}_{a \in \mathcal{A}}$ is the feature vector for linear state representation and $\theta = \{\theta_a\}_{a \in \mathcal{A}}$ are the parameters that have to be optimised. Each dimension of θ_a represents the influence of the corresponding feature in the Q_a-function.

In the experiment, the feature vector Φ_a is a 5-dimensional vector composed of the following features for each action: utility loss between the last proposed option and the next one, the square of the previous value, number of options which can still be proposed, length of the dialogue, speech recognition score. $\mathcal{A}$ is defined according to notations in Subsection 3.1 as follows: Accept(τ), RefInsist(τ_k) $\Leftrightarrow$ RefProp(τ_k), with τ_k equal to the last proposed option by the system, RefNewProp(τ_{k+1}) $\Leftrightarrow$ RefProp(τ_{k+1}), with τ_{k+1} the preferred one after τ_k, and Repeat.

3.3 Experiment Results

The experiment includes nine different users $\mathcal{p}_u^i$ whose characteristics are described in Table 1. The systems are fully cooperative ($\alpha_s^i = 1$) with discount factor $\gamma = 0.9$ and sentence error rate $SER^i = 0.3$. The immediate reward $\omega_s^i = \omega_u^i = 1$ for reaching an agreement is the same for all players. The cost distributions are set as the uniform distribution over $[0,1]$: $\delta^{\mathcal{p}_s^i} = \delta^{\mathcal{u}_s^i} = \mathcal{U}_{[0,1]}$. The costs are sampled independently at the beginning of each dialogue.

At first, learning is performed individually on the first five users $\mathcal{p}_u^i$ with Fitted Q-Iteration. The policy is updated every 500 dialogues for a total of 5000 dialogues to ensure convergence. An ϵ-greedy policy is used with $\epsilon = \frac{1}{2j}$ where j is the iteration index. Then, the policy at the end of the learning phase is saved into a player instance: system $\mathcal{p}_s^i$. Finally, systems $\mathcal{p}_s^i$ for $i \in [1,5]$ are evaluated against all nine users $\mathcal{p}_u^j$ for $j \in [1,8]$.

Table 1 reports all the results. Using a policy learnt with a user on another user can yield very low return if the users are too different. In particular, using a policy learnt with a random user on a deterministic user is highly inefficient, but the same statement can be made with users with more subtle differences such as $\mathcal{p}_s^2$ versus $\mathcal{p}_u^1$ with only a 0.38 average return.

Table 1: Simulated users with the average return $\gamma^T R^i(s^i_T)$ obtained by the systems that were previously learnt with other simulated users.

name	user characteristics			average return w. policy π^i_s learnt w. π^i_u				
	type	x/p	centres	π^1_s	π^2_s	π^3_s	π^4_s	π^5_s
π^1_u	deterministic	3	(0,0)	**0.94**	0.38	0.55	0.33	0.35
π^2_u	deterministic	3	(-5,5)	1.04	**1.23**	0.95	0.50	0.52
π^3_u	deterministic	6	(-5,5)	1.06	**1.23**	**1.23**	0.61	0.65
π^4_u	random	0.3	(-5,5)	0.79	0.92	0.94	**1.02**	0.98
π^5_u	random	0.5	(-5,5)	0.83	0.97	1.02	1.08	**1.10**
π^6_u	deterministic	6	(-1,1)	1.02	0.95	**1.08**	0.54	0.54
π^7_u	deterministic	6	(0,0)	**0.91**	0.46	0.64	0.47	0.46
π^8_u	random	0.3	(-1,1)	0.76	0.95	0.86	**1.02**	**1.01**

4 Towards real users profiling

It is planned to develop a web client enabling any human user to play the negotiation game with a simulated user or another human. For the sake of simplicity (it is easier to develop such a web client without handling the speech and natural language understanding and generation), efficiency (it is faster to generate a lot of data with a click-based navigation) and generality (the experiments and results will not be dependent on a specific implementation), the vocal interaction will remain simulated, meaning that instead of interacting naturally, the users will be asked to click on the action they want to perform. Nevertheless, their actions will be corrupted with noise later in the same way as in the simulation.

If we suppose that the human users are rational, different human user behaviours might be induced by the setting of four parameters:

- Discount factor γ: the lower γ is, the more impatient the user will be.
- Reward for reaching an agreement ω^i: the lower ω^i is, the less inclined the user will be to make efforts to find an agreement.
- Cost distribution δ^i: the higher the mean of δ^i, the more difficult it will be for the user to find a suitable option. The higher the variance of δ^i, the more stubborn the user will be.
- Cooperation parameter α^i: the lower the cooperation parameter α^i, the less empathic the user will be.

A setting of these parameters are called a role. For instance a boss should have a standard discount factor, a low reward for reaching an agreement, a high-mean and high-variance cost distribution, and a low cooperation parameter. Thus, a human can be assigned to any role in a given situation. Data will be gathered from a set of ξ humans adopting a set of ρ roles, which will allow the learning of $\xi \cdot \rho$ user models. Human models can be learnt through imitation learning or inverse reinforcement learning [21, 22], and be used for further studies.

5 Conclusion

This article presented the model of the negotiation dialogue game in order to generate artificial dialogue datasets that can be used to train and test data-driven methods later on. Several handcrafted heterogeneous users are developed and policies that are learnt with Fitted Q-Iteration individually on each of them are shown to be inefficient against other users. This game intends to be useful for experimenting data driven algorithms for negotiation and/or user adaptation.

For the near future, we plan to use the negotiation dialogue game to study Knowledge Transfer for Reinforcement Learning [23, 24] applied to dialogue systems [25, 17]. We also project to use this game to generalise the work in [10] for general-sum games. Finally, co-adaptation [16] in dialogue will be tackled.

Two improvements of the game are already considered: we will implement a web client for human data collection ; we will eventually use a more accurate model for the option proposition: often, in negotiation games, options are not monolithic, they have a complex structure, which implies two things: they cannot always be expressed and understood in a single dialogue turn, and they are not necessarily proposed by a single player, but are rather co-built.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
2. Levin, E., Pieraccini, R.: A stochastic model of computer-human interaction for learning dialogue strategies. In: Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech) (1997)
3. Laroche, R., Putois, G., Bretier, P., Bouchon-Meunier, B.: Hybridisation of expertise and reinforcement learning in dialogue systems. In: Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech). pp. 2479–2482 (2009)
4. Lemon, O., Pietquin, O.: Data-Driven Methods for Adaptive Spoken Dialogue Systems: Computational Learning for Conversational Interfaces . Springer (2012)
5. English, M.S., Heeman, P.A.: Learning mixed initiative dialogue strategies by using reinforcement learning on both conversants. In: Proceedings of the conference on Human Language Technology (HLT) (2005)
6. Georgila, K., Traum, D.R.: Reinforcement learning of argumentation dialogue policies in negotiation. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech). pp. 2073–2076 (2011)
7. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136(2), 215–250 (2002)
8. Georgila, K., Nelson, C., Traum, D.: Single-agent vs. multi-agent techniques for concurrent reinforcement learning of negotiation dialogue policies. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL) (2014)
9. Shapley, L.S.: Stochastic games. Proceedings of the National Academy of Sciences of the United States of America 39(10), 1095 (1953)

10. Barlier, M., Perolat, J., Laroche, R., Pietquin, O.: Human-machine dialogue as a stochastic game. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (Sigdial) (2015)
11. Efstathiou, I., Lemon, O.: Learning non-cooperative dialogue behaviours. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (Sigdial)
12. Putois, G., Laroche, R., Bretier, P.: Online reinforcement learning for spoken dialogue systems: The story of a commercial deployment success. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 185–192. Citeseer (2010)
13. Laroche, R., Putois, G., Bretier, P., Aranguren, M., Velkovska, J., Hastie, H., Keizer, S., Yu, K., Jurcicek, F., Lemon, O., Young, S.: D6.4: Final evaluation of classic towninfo and appointment scheduling systems. Report D6 4 (2011)
14. El Asri, L., Lemonnier, R., Laroche, R., Pietquin, O., Khouzaimi, H.: Nastia: Negotiating appointment setting interface. In: Proceedings of the 9th Edition of Language Resources and Evaluation Conference (LREC) (2014)
15. Genevay, A., Laroche, R.: Transfer learning for user adaptation in spoken dialogue systems. In: Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). International Foundation for Autonomous Agents and Multiagent Systems (2016)
16. Chandramohan, S., Geist, M., Lefèvre, F., Pietquin, O.: Co-adaptation in Spoken Dialogue Systems. In: Proceedings of the 4th International Workshop on Spoken Dialogue Systems (IWSDS). p. 1. Paris, France (Nov 2012)
17. Casanueva, I., Hain, T., Christensen, H., Marxer, R., Green, P.: Knowledge transfer between speakers for personalised dialogue management. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (Sigdial) (2015)
18. Khouzaimi, H., Laroche, R., Lefevre, F.: Optimising turn-taking strategies with reinforcement learning. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (Sigdial) (2015)
19. Gordon, G.J.: Stable function approximation in dynamic programming. In: Proceedings of the 12th International Conference on Machine Learning (ICML) (1995)
20. Chandramohan, S., Geist, M., Pietquin, O.: Optimizing spoken dialogue management with fitted value iteration. In: Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech) (2010)
21. Ng, A.Y., Russell, S.: Algorithms for inverse reinforcement learning. In: Proceedings of the 17th International Conference on Machine Learning (ICML). pp. 663–670. Morgan Kaufmann (2000)
22. El Asri, L., Piot, B., Geist, M., Laroche, R., Pietquin, O.: Score-based inverse reinforcement learning. In: Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). International Foundation for Autonomous Agents and Multiagent Systems (2016)
23. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research 10, 1633–1685 (2009)
24. Lazaric, A.: Transfer in reinforcement learning: a framework and a survey. In: Reinforcement Learning, pp. 143–173. Springer (2012)
25. Gašic, M., Breslin, C., Henderson, M., Kim, D., Szummer, M., Thomson, B., Tsiakoulis, P., Young, S.: Pomdp-based dialogue manager adaptation to extended domains. In: Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (Sigdial) (2013)