Learning non-cooperative dialogue policies to beat opponent models: "The good, the bad and the ugly"

Ioannis Efstathiou
Interaction Lab
Heriot-Watt University
ie24@hw.ac.uk

Oliver Lemon
Interaction Lab
Heriot-Watt University
o.lemon@hw.ac.uk

Abstract

Non-cooperative dialogue capabilities have been identified as important in a variety of application areas, including education, military operations, video games, police investigation and healthcare. In prior work, it was shown how agents can learn to use explicit manipulation moves in dialogue (e.g. "I really need wheat") to manipulate adversaries in a simple trading game. The adversaries had a very simple opponent model. In this paper we implement a more complex opponent model for adversaries, we now model *all* trading dialogue moves as affecting the adversary's opponent model, and we work in a more complex game setting: Catan. Here we show that (even in such a non-stationary environment) agents can learn to be legitimately persuasive ("the good") or deceitful ("the bad"). We achieve up to 11% higher success rates than a reasonable hand-crafted trading dialogue strategy ("the ugly"). We also present a novel way of encoding the state space for Reinforcement Learning of trading dialogues that reduces the state-space size to 0.005% of the original, and so reduces training times dramatically.

1 Previous work

Recently it has been demonstrated that when given the ability to perform both cooperative and non-cooperative / manipulative dialogue moves, a dialogue agent can learn to bluff and to lie during trading dialogues so as to win games more often, under various conditions such as risking penalties for being caught in deception – against a variety of adversaries (Efstathiou and Lemon, 2014b; Efstathiou and Lemon, 2014a). Some of the adversaries (which are computer programs, not humans) could detect manipulation (with increasing probability as more manipulation moves occurred), but only had a simple opponent model which would try to estimate the preferences of the player agent. Furthermore, only specific moves (e.g. "I really need sheep") affected the opponent model, and the setting was a simple 3-resource card-trading game. In this paper we model *all* trading dialogue moves as having effects on the adversary's opponent model (i.e. "I will give you sheep for wheat" means that the adversary believes that the player needs wheat and doesn't need sheep), and we work in the more complex setting of the Catan game (Afantenos et al., 2012).

2 Introduction

Work on automated conversational systems has been focused on cooperative dialogue, where a dialogue system's core goal is to assist humans in their tasks such as buying airline tickets (Walker et al., 2001) or finding a restaurant (Young et al., 2010). However, non-cooperative dialogues, where an agent may act to satisfy its own goals rather than those of other participants, are also of practical and theoretical interest (Georgila and Traum, 2011), and the game-theoretic underpinnings of non-Gricean behaviour have been investigated (Asher and Lascarides, 2008). For example, it may be useful for a dialogue agent not to be fully cooperative when trying to gather information from a human, or when trying to persuade, argue, or debate, or when trying to sell something, or when trying to detect illegal activity, or in the area of believable characters in video games and educational simulations (Georgila and Traum, 2011; Shim and Arkin, 2013). Another arena in which non-cooperative dialogue behaviour is desirable is in negotiation (Traum, 2008), where hiding information (and even outright lying) can be advantageous. Dennett (Dennett, 1997) argues that a deception capability is required for higher-order in-

tentionality in AI.

Machine learning methods have been used to automatically optimise *cooperative* dialogue management - i.e. the decision of what dialogue move to make next in a conversation, in order to maximise an agent's overall long-term expected utility, which is usually defined in terms of meeting a user's goals (Young et al., 2010; Rieser and Lemon, 2011). These approaches use Reinforcement Learning with reward functions that give positive feedback to the agent only when it meets the user's goals. This work has shown that robust and efficient dialogue management strategies can be learned, but until (Efstathiou and Lemon, 2014b), has only addressed the case of cooperative dialogue.

2.1 Corpus analysis

An example of the type of non-cooperative dialogue behaviour which we are generating in this work is given by our (dishonest) trading player agent A in the following dialogue:

A1: "I will give you a wheat and I need 2 clay"[A lies - it does not need clay but it needs wheat]
B1: "No"
A2: "I'll give you a rock and I need a clay"[A lies again and it actually needs rocks too, but it does not have any rocks to give]
B2: "No"
A3: "I'll give you a clay and I need a wheat
B3: "Yes"

Here, B is deceived into providing the wheat that A actually needs, because B believes that A needs clay (A asked for it twice) rather than wheat and rock (that it offered). Similar human behaviour can be observed in the Catan game corpus (Afantenos et al., 2012): a set of on-line trading dialogues between humans playing Settlers of Catan. We analysed a set of 32 logged and annotated games, which correspond to 2512 trading negotiation turns. We looked for explicit lies, of the form: *Player offers to give resource X (possibly for Y) but does not hold resource X* - such as in turn A2 in the above example.

11 turns out of 2512 were lies of this type. Since this corpus was not collected with expert players, we expect the number to be larger for more experienced negotiators. Other lies such as asking for a resource that is not really wanted, cannot be detected in the corpus, since the player's intention would need to be known.

2.2 Non-cooperative dialogues

Our trading dialogues are linguistically cooperative (according to the Cooperative Principle (Grice, 1975)) since their linguistic meaning is clear from both sides and successful information exchange occurs. Non-linguistically though they are non-cooperative, since they they aim for personal goals. Hence they violate Attardo's Perlocutionary Cooperative Principle (PCP) (Attardo, 1997). In the work below, the honest player agent proposes only sincere trades. It offers resources that are available and it asks for resources that it really needs. Hence it is learning to manipulate through legitimate persuasion (Dillard and Pfau, 2002; O'Keefe, 2002) and without any negative consequences. On the other hand, our dishonest player (see below) proposes false trades too, offers resources that are not available, and can ask for resources that it does not need. In other words, it can learn to manipulate based on lies and deception. We will show that both of the player agents can learn how to manipulate their adversaries through different but equally successful policies, by being cooperative on the locutionary level and non-cooperative on the perlocutionary level. In addition, we will present a hand-crafted naive agent who -like the honest player- is sincere, but does not learn how to use manipulation. In other words, it does not take into consideration at all the 'side effects' of its trading proposals, and we show that its performance is significantly lower than that of the two manipulative players.

2.3 Structure of the paper

We initially present the trading game "Catan" (section 3) and describe the version that we use for our experiments. All of the actions (trading proposals) that we use along with their manipulation mechanisms are presented and explained in detail. Section 4 presents the adversary and opponent model that we employ. We then propose a novel way of encoding (compressing) the state space for Reinforcement Learning (RL) with a tabular representation in Section 5, which reduces the training times dramatically. Then we present two Reinforcement Learning Agents (RLA) in Section 6 who -through honesty ("the good") and dishonesty ("the bad")- successfully learn how to use communicative manipulation (with every normal

trading proposal). In Section 6.3 we investigate players without manipulation. Section 7 presents our experiments and detailed results are presented in Section 8.

3 The Trading Game "Catan"

To investigate non-cooperative dialogues in a controlled setting we used a 2-player version of the board game "Catan", which is a complex, sequential, non-zero-sum game with imperfect information. We call the 2 players the "adversary" and the "Reinforcement learning agent" (RLA). We also created a "hand-crafted agent" (HCA) for comparison. We assume that the adversary (see section 4) is affected by all the trading proposals of the learning agents, in such as way that it tries to stop the learning agents from getting the resources that they say they need. Intuitively, this is a basic aspect of adversarial behaviour.

The RLA or the HCA proposes trades to the adversary sequentially and tries to reach a goal number of resources (in the case of a city: 3 rocks and 2 wheat). There are four different goals that can be achieved in the normal "Catan" game: to build a road, a city, a settlement or buy a development card. Our RLA has also learned how to successfully trade in order to achieve all those goals but this paper is based on the example case of the city. There are five different resources to trade and the adversary only responds by either saying "Yes" or "No" to accept or reject the trade respectively. Currently we assume that the adversary has all of the resources available to give so it is up to the RLA or the HCA to use a successful strategy that will allow it to reach its goal. The learning agents start the game with a random number of resources (up to 7 of each resource) and therefore there are cases where the initial number of resources is insufficient to eventually reach their goal. The agents still learn how to get as close to the goal as possible (due to the reward function, see section 6).

3.1 Actions (Trading Proposals)

Trade occurs through trading proposals that may lead to acceptance or rejection from the adversary, and have deterministic and stochastic effects. We will first discuss the action's stochastic effect, that is whether or not the trade will be successful. In an agent's proposal (turn) only one 'give 1-for-1' or 'give 1-for-2' trading proposal may occur, or nothing (41 actions in total for the case of the dishonest RLA):

1. I will do nothing
2. I will give you a wheat and I need a timber
3. I will give you a wheat and I need a rock

...

40. I will give you a brick and I need two rocks
41. I will give you a brick and I need two sheep

In contrast to the case of the dishonest RLA, the cases of the honest RLA and the naive HCA consist of 17 of the above actions because they ask only for goal resources (rock and wheat). The adversary responds by either saying "Yes" or "No" to accept or reject the learning agent's proposals. Each of these actions affects the adversary's opponent model as described below.

3.2 Manipulation through trading actions

We assume that all of the above trading proposals (apart from "I will do nothing") affect the opponent model of the adversary. Hence a trading proposal may or may not lead to a trade (the action's stochastic effect) as we saw, but it will definitely affect (action's deterministic effect) the adversary's belief model. Here we will discuss each action's deterministic effect. Each of the trading proposals consists of two parts: the offered resource and the wanted one(s). The adversary's opponent model is affected by both of these parts – for example the more often the agent insists on asking for wheat, the less the adversary will be eager to give it. Hence the agents need to learn how to appropriately use this effect in order to successfully manipulate the adversary and reach the goal number of resources.

4 The Adversary and its Opponent model

The adversary remains the same in all of our experiments. However other adversary and opponent models are clearly possible. We created this as a simple implementation of the intuition that a rational adversary will act so as to hinder other players in respect of their expressed preferences.

Opponent models (OM) with hindering abilities have previously been shown to be important in games such as the "Machiavelli" card game

(Bergsma, 2005). Hence our adversary is using an opponent model that is based on hindering the LA's preferences, as the LA expresses its preferences through trading proposals and this is the only information that the adversary receives. Since opponent modeling is focused on using knowledge about other agents to improve performance, the adversary therefore hinders the LA's announced preferences (trading proposals).

Our model is inspired by this approach to OM and uses knowledge (from the LA's announcements) in an effort to improve its performance. Unlike the OM (Carmel and Markovitch, 1993; Iida et al., 1993a; Iida et al., 1993b) or the PrOM search model of (Donkers et al., 2001) though, it does not explicitly predict the moves of the LA, but the history of those moves are used to direct the adversary's future responses.

The adversary therefore uses an opponent model which directs its responses to the other agent's (RLA or HCA) trading proposals. Every time that an agent utters a trading proposal, probabilities of the adversary giving resource types change accordingly (details below), and therefore the adversary becomes more or less eager to give some resources than others. It does this because it tries to hinder the other players from acquiring the resources that they ask for. For instance, if an agent insists on asking only for wheat then the probability that it will be given becomes very low (the adversary considers it now as valuable), but the relative probability that it will get one of the other four resources increases.

However, the adversary also takes into consideration what the agent offers to *give*, so the more an agent keeps offering a resource the more likely becomes for the adversary to give it too (it considers the resource as less valuable).

In detail, at the beginning of each trading phase the probabilities that represent the adversary's willingness to give each of the resource types start at 50%. When the agent asks for a resource then the probability to give that particular resource is reduced by either 8% or 12% (if it is a 'give 1-for-1' or 'give 1-for-2' trade proposal respectively), and the probability of giving the four other resource types increases accordingly. The probability of giving the offered resource also increases by 8%. We experimented with a variety of different increments, and very similar results were obtained to those presented below, so there is nothing particularly hinges on the 8% figure.

Due to this opponent model, it is possible to manipulate the adversary into eventually giving resources that are needed, if the right trading proposals are made.

5 The State Encoding Mechanism

To overcome issues related to long training times and high memory demands, we implemented a state encoding mechanism that automatically converts all of our trading game states to a significantly smaller number states in a compressed representation. The new state representation takes into consideration the distance from goal and the availability of the resource, as well as its quality (goal or non-goal resource) and uses 7 different characters. The agent's state consists of the numbers of the five resources that it currently has available. In the case of the city, it needs wheat and rocks. That means two out of five resources are goal resources and therefore they can be represented by 'G' (goal) when their number is equal to the goal amount, 'N' (null) when their number is 0, 'M' (more) when their number is more than the goal-quantity, and '1' or '2' when the distance from the goal quantity is 1 or 2 respectively. The 3 non-goal resources are represented by 'Z' (zero) when they are 0 and 'A' (available) when they are more than 0.

For example, the state $\langle 1, 4, 3, 0, 2 \rangle$ would be encoded to $\langle 1, A, G, Z, A \rangle$. The numeric state space of our problem has 8 x 8 x 8 x 8 x 8 (=32,768) states that are encoded to only 4 x 2 x 5 x 2 x 2 (=160) states. This is reduced to 0.005% of the original size of the state space. With this method and despite the fact that the representation still remains tabular, in all of our experiments 3 million training games required only around 10 minutes to finalize. The performances were very successful too as the logic is still based on the precision of the RL tabular representation.

6 The Reinforcement Learning Agents (RLA)

As we discussed earlier the game state is represented by the RLA's encoded set of resources (see section 5). The RLA plays the game and learns while perceiving only its own set of resources. It is aware of its winning condition in as much as it experiences a large final reward when reaching this state. It learns how to achieve the goal state

through trial-and-error exploration while playing repeated games. Each game consists of up to 7 trading proposals, but nothing particularly hinges upon this number – we have experimented with a number of different length constraints, and obtained similar results. The agent is modelled as a Markov Decision Process (Sutton and Barto, 1998): it observes states, selects actions according to a policy, transitions to a new state (due to the adversary's response), and receives rewards at the end of each game. This reward is then used to update the policy followed by the agent using the SARSA(λ) algorithm.

As we see in Figure 1, it learns to win 96.8% of the time (not 100% due to the cases with insufficient initial resources).

6.1 Reward function

The reward function used in all the experiments takes into consideration the number of trading proposals made and the distance from the goal, as well as trading success. In detail, the reward function that is used is: + 10,000 (if trading successful) $-(1,000*$ proposals) $-(1,000*$ distance).

6.2 Training parameters

The agents were trained using a custom SARSA(λ) learning method (Sutton and Barto, 1998) with an initial exploration rate of 0.2, which gradually decays to 0, and a learning rate α of 1, which also gradually decays to 0 by the end of the training phase. After experimenting with the learning parameters we found that with λ equal to 0.9 and γ equal to 0.9 we obtain the best results for our problem and therefore these values have been used in all of the experiments that follow.

6.3 Initial cases with no manipulation / cooperative adversary

Before we examine the cases with manipulation and the adversary's opponent model, we first explore the case of learning a trading policy for adversaries that do not have an opponent model and thus do not try to hinder the learning agent. This adversary always accepts an agent's trading proposal, and so this serves as an initial proof-of-concept of the extent to which the game is winnable by the learning agents if the adversary is being fully cooperative.

Here the RLA learned how to successfully trade in the full version of the "Catan" game for every goal case. These include building a road, a city, a settlement, or a development card. The different goals are different numbers and types of resources that the RLA needs to gather in order to win.

The RLA has located a successful policy for each one of those cases, showing that the cooperative version of the game is solvable as an MDP problem. It has identified and taken advantage of the power of the 'give 1-for-2' over the 'give 1-for-1' trades and therefore it uses them much more frequently (with a ratio of around 75% over 25% for the 'give 1-for-1'). The adversary that it plays against does not have an opponent model, the learning agent's trading proposals do not affect it, and the adversary always accepts them. Hence we initially show that RL is capable of successfully learning how to trade in this version of the game (with every different goal) while learning to also exploit the 'give 1-for-2' trading proposals.

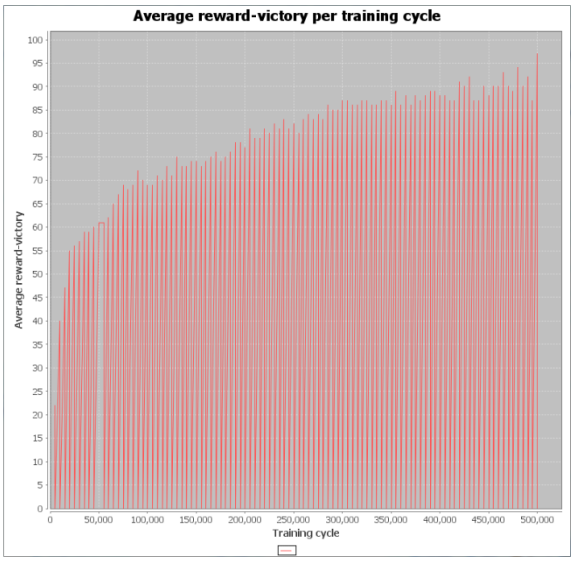

Figure 1: *Learning Agent's reward-victory graph in 500 thousand training games of Initial Experiment: building a city, cooperative adversary.*

6.4 The Honest Reinforcement Learning Agent - "The Good"

The honest RLA only asks for resources that it really needs (therefore it is restricted to 17 out of the 41 actions). It is a sincere RLA and it only proposes a trade after it has checked that the offered resource is indeed available. However, the fact that it still learns how to successfully manipulate (legitimately persuade) the adversary under those honest constraints, and in a continuous non-stationary MDP environment due to the ever-changing adversarial belief model (i.e. the envi-

ronment's dynamics can change after an action is selected), makes the outcome surprising. In the experiments that follow we will see that it locates a honest way of persuading its adversary.

6.5 The Dishonest Reinforcement Learning Agent - "The Bad"

The dishonest RLA can ask for resources that it does not need (therefore it uses all of the 41 actions). It can also propose trades without checking if the offered resource is available. If such a deceitful trading proposal gets accepted by the adversary, the RLA then refuses to actually make the trade. Thus its learning process is a harder Reinforcement Learning task than that of the honest RLA (since it has more actions). However, it still learns how to successfully manipulate (deceive) the adversary under those dishonest conditions, and in a continuous non-stationary MDP environment due to the ever-changing adversarial opponent model as above, resulting on a surprisingly equal performance with that of the honest RLA. As we will see in the experiments that follow, its strategy is based on the use of lies.

6.6 The Naive Hand-Crafted Learning Agent - "The Ugly"

This agent is not a learning agent but instead uses a hand-crafted naive strategy. In detail, it uses a reasonable way of proposing trades by checking the availability of the resources that it does not need and offers them for those that it needs in an equi-probable manner. The reason that we call it naive (as well as "ugly") is because it does not take into consideration the fact that its trading proposals affect the adversary's opponent model and -instead of learning that- it just keeps following the same naive rule-based strategy. This agent is a baseline case and despite the fact that its strategy is quite sensible, we show that it is significantly worse than that of the two manipulative RLAs.

7 Experiments

All agents are compared in respect of their win rates, which is the percentage of trading games in which they achieve their goal (in this case, to get the resources required to build a city). The y-axes of the graphs below represent this quantity (which we also refer to as "success rate" or "reward-victory").

7.1 Naive HCA vs. Adversary: Experiment 1 (Baseline)

The naive HCA played 3 million games against the Adversary in Experiment 1. This is our baseline case for comparison. The agent's trading proposals affect the opponent model of the adversary but the agent is unaware of that and therefore it does nothing about it. It just keeps playing the game based on the naive but reasonable strategy discussed in Section 6.6.

7.2 Honest RLA vs. Adversary: Experiment 2

In this experiment we trained the honest RLA against the adversary in 3 million games. The RLA's trading proposals affect the opponent model of the adversary and we show that, despite the honest constraints, the honest RLA can learn how to successfully manipulate the adversary. Ultimately we show that the performance is better than that of the baseline case in Experiment 1. The performance of the Honest RLA before training (i.e. random action selection) is about 21%.

7.3 Dishonest RLA vs. Adversary: Experiment 3

In this experiment we trained the dishonest RLA against the adversary in 3 million games. The RLA's trading proposals again affect the opponent model of the adversary and we show that the dishonest RLA can learn how to successfully manipulate it. As above, we show that the performance is better than that of the baseline case in Experiment 1. Furthermore, we explore how well this deceitful RLA performs compared to the previous honest one, who legitimately persuades. The performance of the Dishonest RLA before training (i.e. random action selection) is about 4%.

8 Results

The RLAs were trained on 3 million games against the Adversary. Their policies were then tested in 20,000 games. The HCA played 3 million games too against the same adversary. As there was no learning, no testing games were played because its performance remained stable throughout the 3 million games as we will see below.

8.1 Naive HCA: Experiment 1

The naive HCA has a win rate of only 25.3%. Its strategy focuses on 50% of the time asking for

wheat by offering each one of its available unwanted resources in turn, or 50% of the time asking for rocks using the same technique.

8.2 Honest RLA: Experiment 2

The honest RLA scored a winning performance of 35.8%, see Figure 2, starting from 21.1% (which is the performance of random action selection). Its strategy focuses on asking initially for either wheat, until it gathers rocks, or for rocks until it gathers wheat that needs to build a city (2 wheat and 3 rocks are required). It also mainly offers resources that it needs (goal ones) -and has available- instead of non-goal ones as it will become then easier to get them back. This honest persuasive strategy proved to be very effective against the adversarial hindering policy.

8.3 Dishonest RLA: Experiment 3

The dishonest RLA scored a winning performance of 36.2% after 3-million training games, and may improve with further training (see Figure 3), starting from only 4.2%. That clearly shows that its task was much harder than that of the honest RLA in Experiment 2, who started from 21.1%, as it has to understand how to effectively manipulate through all of the 41 actions (rather than the 17 honest actions which ask for goal resources only). Nevertheless its very effective learned strategy mainly focuses on the use of lies. It asks especially for resources that it does not need only for the sake of manipulation (deception) and it offers resources that it does not have for the same purpose. The type of the offered resources in this case are mainly goal ones again (as above) and the fact that this RLA can lie about their availability makes such offers even more frequent than before. This dishonest strategy proved to be equally effective with that of the honest RLA though.

Both of the RLAs (as we saw in Experiment 2 too) managed to learn successful strategies despite the fact that there are cases where the initial resources are insufficient to reach the goal within 7 proposals. They both realized again (as in our Initial Experiment, section 6.3) the power of the 'give 1-for-2' over the 'give 1 for-1' trades and they used them more often. Hence, in some cases they manage to approach their goals even with insufficient initial resources. By comparing the two manipulative cases to that of Experiment 1 we show that manipulation (through legitimate persuasion [Experiment 2] or deception [Experiment 3]) can be successfully learned by our RLAs and outperform by 11% a naive but reasonable strategy.

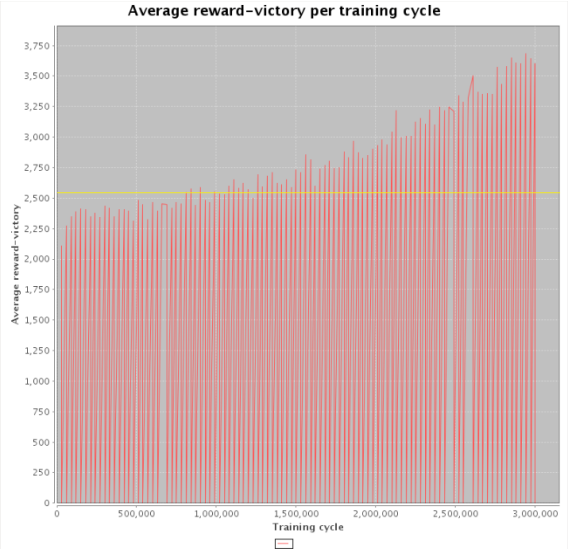

Figure 2: *Honest RLA's reward-victory graph in 3 million training games (experiment 2). Yellow horizontal line = Baseline performance.*

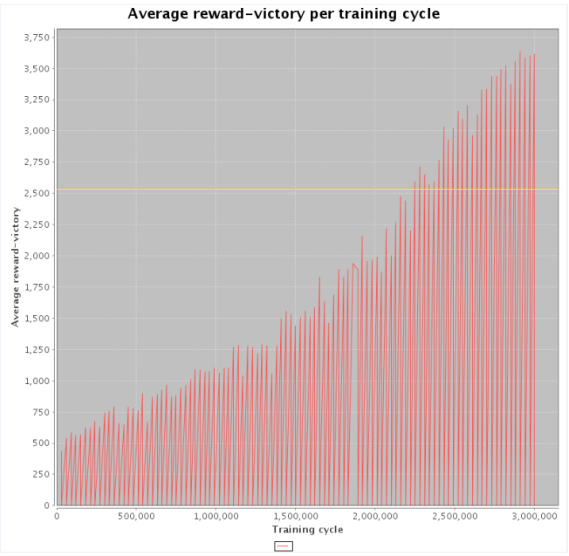

Figure 3: *Dishonest RLA's reward-victory graph in 3 million training games (experiment 3). Yellow horizontal line = Baseline performance.*

9 Discussion: a Non-Stationary MDP problem

Our Experiments 2 and 3 also show that RL is capable of learning successful policies even in the case where the environment's dynamics change (maximum of 7 times per game) and each action (trading proposal) has a stochastic effect (that of

Exp.	**Learning Agent policy**	**Adversary policy**	**Agent's wins**
Initial	SARSA + Honest actions	Accepts every trade	96.8%
	Random Honest actions	Hinders agent's preferences	21%
	Random Dishonest actions	Hinders agent's preferences	4%
1	Hand-Crafted Naive Honest (Baseline)	Hinders agent's preferences	25.3%
2	SARSA + Honest actions	Hinders agent's preferences	35.8%*
3	SARSA + Dishonest actions	Hinders agent's preferences	36.2%*

Table 1: *Performance (% wins) in 20 K testing games, after training. (*= significant improvement over baseline,* $p < 0.05$*)*

a possible trade) and a deterministic effect (that on adversary's opponent model). Every time the honest or dishonest RLA proposes a trade, the opponent model of the adversary changes as we have seen. That means the environment changes too (as the adversary is a part of it according to the RLA's perspective) and therefore makes our problem a non-stationary MDP (da Silva et al., 2006). Despite the fact that only the RLA's actions are responsible for those changes and so the problem may be solved by recasting it into a stationary one through state augmentation (Choi et al., 2001), our case is more complex. This is because our RLA's actions affect the environment in two different ways (through their stochastic and deterministic effects). Furthermore, the environment (adversary) responds to trading proposals based on the history of the deterministic effects of the actions (trading proposals' effect on adversary's belief) up to that point. In other words, the same action (trade) may have different effects due to the deterministic effects on the environment (changes of the adversary's opponent model) of the actions that preceded it. There are successful combinations between these two different kinds of effects that the RLA has managed to identify and learn how to effectively use, originating from the multi-dimensions (manipulative dimensions) of the problem. It is therefore an interesting multi-dimensional non-stationary MDP case that we have shown to be solvable by RL, which suggests that trading proposals in dialogue evoke non-stationary beliefs in our everyday negotiations. We demonstrated that phenomenon with the realistic assumption that the adversary's opponent model is affected by all normal trading actions.

10 Discussion: Discourse Studies

Our results also bring an important argument of Van Dijk (van Dijk, 2006) to light, according to which there is an everyday conventional inference of dishonesty from manipulative acts. That negative effect cannot be taken for granted though as manipulation according to Dillard and Pfau, as well as O'Keefe (Dillard and Pfau, 2002; O'Keefe, 2002) also occurs through legitimate persuasion. This is what our RL work suggests too. Hence we emphasize the significance of Attardo's perlocutionary cooperation as before.

11 Conclusion & Future Work

In this paper we implemented an opponent model for adaptive adversaries, and modelled *all* trading dialogue moves as affecting the adversary's opponent model. We worked in the complex game setting of Catan and we showed that agents can learn to be legitimately persuasive ("the good") or deceitful ("the bad"). We achieve up to 11% higher success-rates than a reasonable hand-crafted trading dialogue strategy ("the ugly").

We also presented a novel way of encoding the state space for Reinforcement Learning of trading dialogues that reduces the state-space size to 0.005% of the original, and so reduces training times dramatically.

In future work we will further investigate complex non-cooperative situations, and evaluate the performance of such learned policies in games with humans, by integrating this work with jSettlers (Thomas and Hammond, 2002).

Acknowledgements

This work is partially funded by the European Research Council, grant no. 269427 (STAC project). Thanks to Heriberto Cuayáhuitl for help with the corpus analysis.

References

S Afantenos, N. Asher, F. Benamara, A. Cadilhac, C Dégremont, P Denis, M Guhe, S Keizer, A. Lascarides, O Lemon, P. Muller, S. Paul, V. Popescu, V. Rieser, and L. Vieu. 2012. Modelling strategic conversation: model, annotation design and corpus. In *Proc. 16th Workshop on the Semantics and Pragmatics of Dialogue (Seinedial)*.

N. Asher and A. Lascarides. 2008. Commitments, beliefs and intentions in dialogue. In *Proc. of SemDial*, pages 35–42.

S. Attardo. 1997. Locutionary and perlocutionary cooperation: The perlocutionary cooperative principle. *Journal of Pragmatics*, 27(6):753–779.

M.H.J. Bergsma. 2005. Opponent Modeling in Machiavelli. B.s. thesis, Maastricht University, the Netherlands.

D. Carmel and S. Markovitch. 1993. Learning models of opponent's strategies in game playing. In *Proceedings AAAI Fall Symposium on Games: Planning and Learning*, pages 140–147. The AAAI Press.

Samuel P.M. Choi, Dit-Yan Yeung, and Nevin L. Zhang, 2001. *Sequence Learning - Paradigms, Algorithms, and Applications*, chapter Hidden-Mode Markov Decision Processes for Nonstationary Sequential Decision Making. Springer-Verlag.

Bruno C. da Silva, Eduardo W. Basso, Ana L.C. Bazzan, and Paulo M. Engel. 2006. Dealing with Non-Stationary Environments using Context Detection. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*.

Daniel Dennett. 1997. When Hal Kills, Who's to Blame? Computer Ethics. In *Hal's Legacy:2001's Computer as Dream and Reality*.

James Price Dillard and Michael Pfau. 2002. *The Persuasion Handbook: Developments in Theory and Practice*. SAGE Publications, Inc.

H. H. L. M. Donkers, H. J. Van Den Herik, and J. W. H. M. Uiterwijk. 2001. Probabilistic opponent-model search. *Information Sciences*, 135:123–149.

Ioannis Efstathiou and Oliver Lemon. 2014a. Learning to manage risk in non-cooperative dialogues. In *Proc. SEMDIAL*.

Ioannis Efstathiou and Oliver Lemon. 2014b. Learning non-cooperative dialogue behaviours. In *SIGDIAL*.

Kallirroi Georgila and David Traum. 2011. Reinforcement learning of argumentation dialogue policies in negotiation. In *Proc. INTERSPEECH*.

Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3.

H. Iida, J.W.H.M. Uiterwijk, H.J. van den Herik, and I.S. Herschberg. 1993a. Opponent-model search. Technical report cs 93-03, Universiteit Maastricht.

H. Iida, J.W.H.M. Uiterwijk, H.J. van den Herik, and I.S. Herschberg. 1993b. Potential applications of opponent-model search. part 1: The domain of applicability. *ICCA Journal*, 16(4):201–208.

Daniel O'Keefe. 2002. *Persuasion: Theory and research (2nd Edition)*. SAGE Publications, Inc.

Verena Rieser and Oliver Lemon. 2011. *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing. Springer.

J. Shim and R.C. Arkin. 2013. A Taxonomy of Robot Deception and its Benefits in HRI. In *Proc. IEEE Systems, Man, and Cybernetics Conference*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.

R. Thomas and K. Hammond. 2002. Java settlers: a research environment for studying multi-agent negotiation. In *Proc. of IUI '02*, pages 240–240.

David Traum. 2008. Extended abstract: Computational models of non-cooperative dialogue. In *Proc. of SIGdial Workshop on Discourse and Dialogue*.

Teun A. van Dijk. 2006. Discourse and manipulation. *Discourse & Society*, 17(2):359–383.

M. Walker, R. Passonneau, and J. Boland. 2001. Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Steve Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.