Multi-Agent Deep Reinforcement Learning for Negotiation Bots

Simon Zhuang
University of California, Berkeley
Email: simonzhuang@berkeley.edu

Abstract—With the advancement of deep reinforcement learning applications, there has recently been increasing efforts to examine environments in which multiple deep reinforcement learning agents interact. In this paper, a negotiation environment presented in the Lewis et al. 2017 paper Deal or No Deal? End-to-End Learning for Negotiation Dialogues [6] is used as the platform for multi-agent deep reinforcement learning. The Lewis et al. 2017 paper trains agents with single agent reinforcement learning, which this paper shows will yield suboptimal agents. Instead, this paper adapts multi-agent reinforcement learning to this environment and shows that it yields more robust agents. Using the language model from the Lewis et al. 2017 paper, both agents are independently outfitted with deep reinforcement learning algorithms based on policy gradient methods. Several adjustments to the method were made, the most significant of which was to interweave the policy gradient steps with supervised learning steps using the original language model. This update was instrumental in ensuring that both reinforcement learning agents converged upon a policy. The agents were analyzed throughout and after the training process. Metrics indicated that the single agent reinforcement learning agent adopted a strategy of exploitation against the imitation learning agent that was significantly suboptimal against other agents. Analysis of the behavior of the bot confirmed this, indicating an opponent-specific strategy. On the other hand, the multi-agent reinforcement learning bots yielded relatively strong rewards under each circumstance, and was an empirically reasonable strategy. Since the local optimal strategy in a multi-agent optimization game is a Nash equilibrium, the results are likely to be approximately optimal relative to each other.
Our code is available at github.com/simonzhuang/end-to-end-negotiator.

1 Introduction

Within the past several years, deep reinforcement learning has been applied to a new, wide range of tasks. Already, deep reinforcement learning agents have had super-human performance in games such as Go [1] and Atari games [2]. As deep reinforcement learning research progresses and is applied in the real world, deep reinforcement learning agents will have to increasingly interact with outside agents, including other AI agents. In order to maximize reward, such agents will have to take into account the actions of other agents, which may or may not be trying to achieve a related goal.

The inevitability of reinforcement learning agents having to interact has lead to a rise in the past several years in papers addressing multi-agent reinforcement learning, where deep reinforcement learning methods are applied to more than one agent within an environment. One main challenge of multi-agent reinforcement learning is that the environment for a given agent is non-stationary. Since other agents change behavior during the process, for a given agent, the optimization problem changes throughout the process. Despite this, recent literature has shown that traditional deep reinforcement learning architectures, such as deep Q-learning [3] and actor-critic [4], have been effective in multi-agent reinforcement learning environments. Additionally, there is also active research in algorithms specifically designed for multi-agent environments [5].

One natural environment in which deep reinforcement learning agents could interact is in the context of resource allocation. In 2017, Lewis et al. from Facebook AI Research released a paper entitled "Deal or No Deal? End-to-End Learning for Negotiation Dialogues"

[6]. In the paper, using natural language processing methods, agents were taught to negotiate to divide resources. However, only single agent reinforcement learning was done, which presents issues to be addressed in this paper.

In this paper, we apply multi-agent reinforcement learning to the environment presented in the Lewis et al. [6] paper. While the Lewis et al. paper contains significant novelty on language generation, for this study, we blackbox the natural language processing architecture and focus on reinforcement learning. Algorithms for multi-agent reinforcement learning are explored and implemented, and the resulting complexity is compared with single agent reinforcement learning implemented similar to the Lewis et al. paper.

Most significantly, we show that multi-agent reinforcement learning yields qualitatively and quantitatively superior agents. Single-agent reinforcement learning against a static opponent yields an agent that is exploitative against their specific opponent, but does not generalize well. On the other hand, multi-agent reinforcement learning yields robust agents that both approach optimal behavior.

2 Environment

The reinforcement learning environment used in this study is the negotiation platform described in detail in Lewis et al. [6]. In a given scenario, there are two agents, Alice and Bob. There exists three different objects to divide between them: hats, books, and balls. Each scenario is set up such that Alice and Bob each have a different reward function that is a linear function of the number of each object they are allocated, with non-negative weights for each object. The generation is set up such that the total value for a user of all items is 10, each item has non-zero value to at least one user, some items have non-zero value to both users, and there are between 5 and 7 total items in the pool.

Either Alice or Bob starts negotiation by sending and receiving a series of messages. When both agents agree to an allocation (or agree to stop), each agent outputs their belief of the allocation to which they agreed. If the two agents' beliefs coincide, their agreed allocation is used to tabulate their respective scores for the round. Otherwise, both agents receive a score of zero.

The negotiation game is interesting for several reasons. Firstly, it is a positive sum game, since each object has different value to each player. Thus, in expectation, there are strategies in which each agent receives more utility than a uniformly random allocation. Hence there is an incentive to play and to negotiate. However, there remains a competitive element to the game, since there is at least one object to which both agents assign positive value. The use of English as a negotiation protocol allows the advantage that the agents' behavior can be analyzed and understood, perhaps at the cost of some efficiency for the negotiating agents.

3 Methods

3.1 Setup

This study uses a dataset of 5808 sample dialogues generated by humans, the collection process explained in the Lewis et al. [6]. Additionally, the same recurrent network architecture using gated recurrent units is adopted to encode and decode language. Using the process described in detail Lewis et al., a baseline imitation learning model was trained, which parses and generates dialogue based on learning from the sample dialogues. Given knowledge of the environment and previous dialogue, the model trains to minimize the negative log likelihood of token sequences from the sample dialogues.

3.2 Single Agent Reinforcement Learning

The single agent reinforcement learning is, likewise, implemented based on the Lewis et al. [6] paper. In this case, negotiation simulation starts with both Alice and Bob set as the pretrained imitation model. Throughout the process the parameters of Bob's model are kept constant, while the parameters for Alice are updated through reinforcement learning. Standard policy gradient methods are used [7], with the reward function described as follows.

$$R(x_t) = \sum_{x_t \in X^A} \gamma^{T-t}(\text{score}_A - \mu) \qquad (1)$$

Here, for a given complete dialogue r^A is the final score based on resource allocation, T is the total length of the dialogue, and γ is a discount factor that results in utterances later being weighted more important. μ is a running average of complete dialogue rewards, and used as a baseline. Additionally, every k iterations, a supervised update occurs in the same way as in the imitation learning scenario.

3.3 Multi-Agent Reinforcement Learning

The multi-agent reinforcement learning learning process similarly begins with two independent pretrained agents from imitation learning. However, in this case, both agents' parameters are concurrently updated through every dialogue, based on their rewards. Like the single agent case, the learning algorithm is based on policy gradient; however, several modifications to the process were made to ensure convergence. First the reward function was modified so that agents terminate conversation within a specific number of dialogue steps.

$$R(x_t) = \begin{cases} \sum_{x_t \in X^A} \gamma^{T-t}(\text{score}_A - \mu) & t < 10 \\ 0 & t \geq 10 \end{cases} \quad (2)$$

Additionally, the aforementioned supervised updates were performed for every iteration of policy gradient. This algorithm of reinforcement learning interwoven with supervised learning ensures the overall model, especially the architecture for language generation, stays within bounds of the original dialogue samples. Both of these modifications were crucial in preventing the models from diverging from sensible language and ensuring that the agents were able to establish any sort of dialogue.

3.4 Metrics Collected

Throughout the training process, two main metrics were noted. Let $v_{X,\text{item}}$ be the value of a specific item to agent X, and let $n_{X,\text{item}}$ be how many of that item is allotted to agent X. Additionally, denote n_{item} be the total number of that item to be allotted. Thus, $n_{\text{item}} = n_{A,\text{item}} + n_{B,\text{item}}$.

First, each agent's score was collected. For example, the score for Alice is calculated as the following.

$$\text{score}_A = \sum_{\text{item} \in \{\text{ball,book,hat}\}} v_{A,\text{item}} n_{A,\text{item}} \quad (3)$$

Bob's score is calculated similarly.

Additionally, the present study introduces a new term called optimality, which represents how well the agents are collectively doing. Specifically, optimality is defined as the following.

$$\frac{\text{score}_A + \text{score}_B}{\sum_{\text{item} \in \{\text{ball,book,hat}\}} (\max\{v_{A,\text{item}}, v_{B,\text{item}}\}) n_{\text{item}}} \quad (4)$$

. This essentially measures how close a given allocation is to the best possible allocation in terms of total score.

For each, for the purposes of plotting and analysis, an exponentially decaying moving average (EWMA) is used. Start with given series of variables x_t which have a different value at each iteration t, we use the following formula to calculate the EWMA z_t for every 100 iteration.

$$z_t = \lambda \frac{\sum_{j=t-99}^{t} x_j}{100} + (1 - \lambda) z_{t-100} \quad (5)$$

In this paper, λ is set to be 0.1. This has the effect of reducing the noise in graphs while putting more weight on recent iterations that more accurately represent the model at a given timestep.

Algorithm 1 Multi-Agent Policy Gradient for Negotiation

```
Train π_S using imitation learning
Initialize π_A = π_S and π_B = π_S
loop
    Generate context ({r_{X,item}}, {n_item})
    Run π_A and π_B in context, generate scores
    Use scores to update π_A and π_B with PG
    Supervised update for π_A and π_B
end loop
```

4 Results

4.1 Agent Scores

During the training process, most significant and directly observable is the score of each

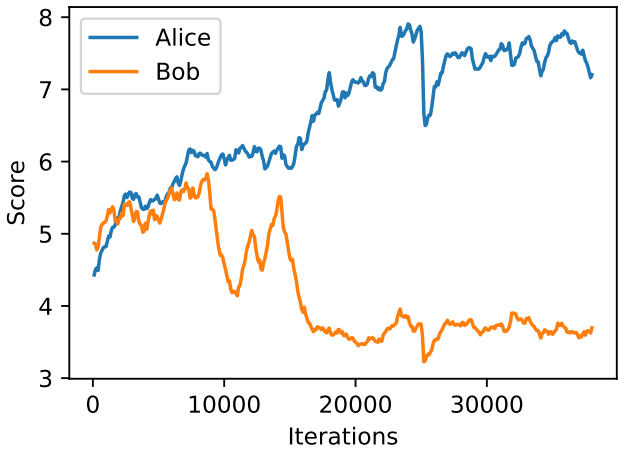

Fig. 1. EWMA of each agents's score throughout the single agent training process

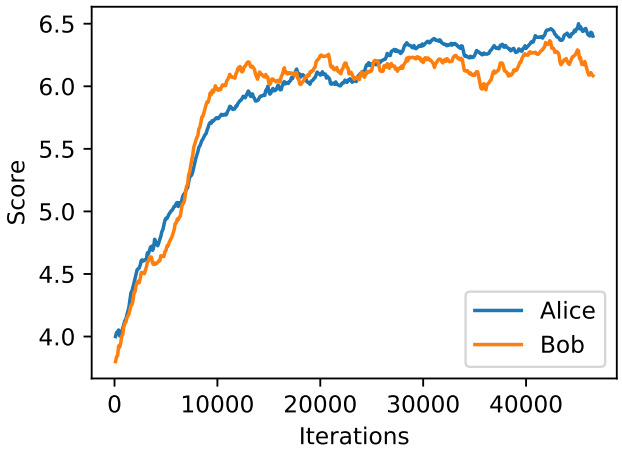

Fig. 2. EWMA of each agents's score throughout the multi agent training process

respective agent. In each scenario, the imitation learning agents start with an average score of around 4.6 when played against each other. In the single agent reinforcement learning case, initially both scores increase; however, after about 9,000 iterations, the score between the static and non-static agents diverge, as shown in Figure 1. This is an indication that the reinforcement learning agent begins to learn exploitative tactics that benefit the reinforcement learning agent at the detriment of the static agent. After 30,000 iteration, the scores level off at around 7.2 and 3.4 for the reinforcement learning and static agents, respectively.

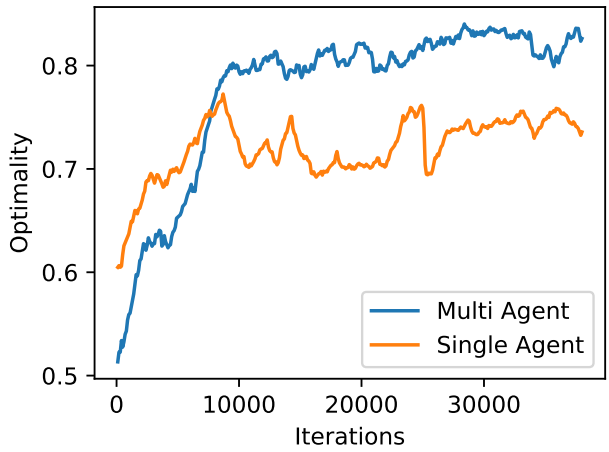

Fig. 3. EWMA of the optimality of each agent during the training process

In the multi agent scenario, through the entire training process, the scores of the two independent agents remain relatively close to each other, only ever diverging by less than 0.5 points, as shown in Figure 2. Both agents quickly increase their scores to about 5.5 and improve slowing from there. The incremental improvement continues for about 30,000 further iterations, reaching about 6.2 for both agents.

4.2 Optimality

In order to present a single metric denoting the overall success of the agents, optimality, defined previously, was measured through the training process (Figure 3). Results indicate single agent training yielded gains in optimality faster initially, likely corresponding to the faster convergence seen earlier. However, once the exploitation began, at around 9000 iterations, the optimality of the single agent reinforcement learning agent decreased. Interestingly, while the optimality decreased, the reinforcement learning agent's reward during this period still increased overall. Thus, aside from some noise, the entirety of the loss in utility was from the static agent.

The multi-agent optimality curve follows a similar patter as its training curves, with fast initial gains and slower gains later in the process. However, the curve does stabilize at about

Environment:

Item	Count	A Value	B Value
Book	1	3	7
Hat	1	1	3
Ball	3	2	0

Dialogue:

ALICE: I would like the book and the balls
BOB: I need the book and at least one ball
ALICE: I would like the balls and the book
BOB: I can't make that deal. I want the book and the hat
ALICE: I would like the balls and the book
BOB: ok

Score:

ALICE: 1 Book, 0 Hats, 3 Balls → 9 Points
BOB: 0 Books, 1 Hats, 0 Balls → 3 Points

Fig. 4. Sample dialogue between the single agent reinforcement learning bot and the imitation learning bot

Environment:

Item	Count	A Value	B Value
Book	1	1	8
Hat	3	3	0
Ball	2	0	1

Dialogue:

ALICE: I would like the hats and the book
BOB: I need book and balls to make a deal
ALICE: I need the hats and you can have the rest
BOB: Okay deal

Score:

ALICE: 0 Book, 3 Hats, 0 Balls → 9 Points
BOB: 1 Books, 0 Hats, 2 Balls → 10 Points

Fig. 5. Sample dialogue between the multi agent reinforcement learning bots

A	B	A Mean Score	B Mean Score
Imitation	Imitation	4.66	4.72
Imitation	SARL	3.54	7.22
Imitation	MARL	5.64	6.70
SARL	SARL	0.25	0.22
SARL	MARL	2.31	2.13
MARL	MARL	6.29	6.13

Fig. 6. Head-to-head results from playing various agents against each other

0.8, significantly higher than the single agent case. Additionally, the curve for multi-agent reinforcement learning seems to vary much less than that of single agent reinforcement learning. One possible reason for this could be, as illustrated in the examples below, that the agents in the multi-agent reinforcement learning case seem to have a strategy with inherently less variance than the strategy pursued in the single agent environment.

4.3 Agent Behavior

Analyzing the dialogue of the agents allows for insight into the strategy each agent pursues to maximize their goals. Overall, the dialogues yielded consistent strategies, although sometimes there seemed to be some noise in the dialogue. These strategies are illustrated in Figures 4 and 5, sample dialogues in each of the scenarios.

In the single agent reinforcement learning case, the reinforcement learning agent, Alice, has a strategy in which they repeatedly asks for a favorable allocation of the resources, to which the static agent more often than not agrees. As seen in the training curves, the strategy does seem to be higher variance, since there are times where the static agent refuses the offer and no agreement is made. However, on average, the reinforcement learning agent's strategy offers high reward by exploiting the static agent's propensity to agree.

Overall, in the multi-agent reinforcement learning environment, the negotiation dynamics have several distinctions. In this case, the two agents almost always agree, and often rather quickly. The dynamics of the game are such that both agents can reach a near maximum individual score in the majority of generated scenarios. As a result, these agents' strategy tends to involve each agent initially pursuing high value items, yet being willing to compromise on lower value items to ensure agreement.

4.4 Head-to-Head Results

By placing these agents against each other, the effectiveness of each agent against other agents can be noted (Figure 6). Here, imitation refers

to the agent trained using only imitation learning, SARL refers to a reinforcement learning agent trained against the imitation learning agent, and MARL refers to an agent trained in the multi-agent reinforcement learning environment. These results show that the SARL agent does well against the imitation agent, but fairs extremely poorly in each of the other circumstances. In fact, when two SARL agents play each other, both end up doing exceptionally poorly, rarely earning any reward. On the other hand, the MARL agent proves to be effective against both another MARL agent and the imitation learning agent. Although its score against the imitation learning agent is not as high as that of SARL agent, it manages to yield reward without relying on exploiting the imitation learning agent.

5 Discussion

By comparing the results for both single agent and multi-agent reinforcement learning, emergent complexity can be analyzed. With single agent reinforcement learning, the reinforcement learning agent's reward steadily increases. For the first 9000 iterations the static agent's reward increases as well, indicating that both agents benefit from the updates. Afterwards, however, opponent static agent's reward steadily decreases. Looking into the dialogue reveals the exploitation. The reinforcement learning agent essentially learns that repeatedly asking for the same thing will cause the static agent to eventually agree. Since the static agent was trained to imitate training data, given that the dialogue in the training data results in agreement after several timesteps, it seems that the static agent automatically agrees after several back-and-forths. In fact, it seems that there is a distinct moment when the reinforcement learning model learns to exploit this, at around 9,000 iterations. The exploitative behavior by the reinforcement learning agent is to be expected; in the absence of an adaptive opponent, reinforcement learning in this case reduces to optimizing a policy in a stationary partially observed Markov decision process.

On the other hand, with multi-agent reinforcement learning, the reward and optimality curves both indicate that each agent continuously improves, and as a whole, the agents gain value. Analyzing the dialogues show that agent's negotiations are more two-sided. Reasonably, the agents' negotiation strategies did not contain any obvious weaknesses. As a whole, these robust strategies make sense theoretically; a scenario where two agents are optimized relative to the other agent constitutes a Nash equilibrium. While the agents did not seem to fully converge to a Nash equilibirium, their strategies do seem, from a qualitative standpoint, fairly reasonable and close to optimal.

Interestingly, there is a period where both agents rapidly improve exists in both the single agent and multi-agent environments, which stops when the each agent's scores is about 5.7 in both cases. A hypothesis into the similarities is that, up to that point, the easiest gains are made through better communication and parsing between the agents, which would lead to less scenarios where agents misunderstand each other (in which both agents are reward 0), which would drastically increase both agents average reward in either environment.

Comparing agents trained under the two methods, it seems evident that multi-agent reinforcement learning yields a more overall useful agent. Given the relative weakness of the imitation learning agent, the single agent reinforcement learning bot likely would not generalize well against superior bots. In fact, the strategy it seems to converge upon, asking for the same items repeatedly, evidently does not generalize well against more competent agents. Additionally, there are likely nuances in the bot that are exploitable, since the agent has no incentive to amend them under single agent reinforcement learning. The multi-agent reinforcement learning bots, on the other hand, are constantly trained against an adversary that improves and seeks to exploit them. Hence, such agent is likely more robust against unknown adversaries. That is, it is less likely that there exists another bot that can receive high reward at the expense of the multi-agent reinforcement learning bot.

One weakness of multi-agent reinforcement learning is that, in a non-stationary environ-

ment, the results are slower to converge, since the convergence of one agent is dependent on the convergence of the other agent. The imitation learning at the beginning seems necessary such that the state when training begins somewhat resembles the final state. Additionally, multi-agent reinforcement learning seems brittle, even more so than the original reinforcement learning algorithm. The various modifications, including frequent supervised updates, were instrumental in ensuring the agents did converge. With the increased complexity of the environment and the non-stationary nature, it seems natural that both of these issues persist, although more novel algorithms specifically for multi-agent reinforcement learning [5] may help mitigate the issues.

One of the properties of the original paper that was kept was the use of English as the primary communication protocol between the two agents, enforced through aforementioned supervised learning updates. Removing this constraint cause language to diverge from English, becoming unintelligible not only for humans, but also for the agents themselves: agreement and negotiation broke down when utterances diverged. (This is contrary to many news reports that came out claiming that Facebook created bots that communicated to each other in their own language. It would be more accurate, if less exciting, to say Facebook created bots that forgot how to talk and became unproductive as a result.) Information theoretically, English is clearly far from the most efficient communication protocol. The emergence of language in the context of reinforcement learning is an open research topic, and recently, there have been papers discussing language in the multi-agent context [8] [9], including a paper [10] in the negotiation environment. A natural next step would be to adapt the communication mechanism in a way to expand the sort of information transfer in which agents are able to engage.

6 Conclusion

We presented a multi-agent reinforcement learning approach to the negotiation game presented in Lewis et al. 2017 paper [6]. Using the blackbox language model presented in that paper, a pair of independent negotiation bots were simultaneously trained against one another, each seeking to maximize their own reward based on the allocation of materials. A standard policy gradient algorithm was augmented with supervised learning based on a previous language model in order to prevent divergence from an established communication protocol.

The behavior of the agents were analyzed and several metrics noted. Training logs indicated a difference in the strategies pursued by agents in each case. Under the single agent reinforcement learning, the reinforcement learning agent develops exploitative behavior, earning reward based on weakness of the static bot against which it is trained. While this strategy works well against this specific opponent, it yields poor results against other opponents.

Multi-agent reinforcement learning, in comparison, yields a strategy that is effective at producing reward for a general array of opponents. The algorithm itself is more brittle, but with several ad-hoc constraints, started to converge. Since the locally optimal strategy in a multi-agent optimization game is a Nash equilibrium, the results, in addition to being empirically sound, are likely to be approximately optimal. Thus, in a multi-agent environment like the one presented in the Lewis et al. 2017 paper, multi-agent reinforcement learning yields agents that are significant more robust and generalizable.

REFERENCES

[1] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[3] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *CoRR*, abs/1511.08779, 2015.

[4] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR*, abs/1706.02275, 2017.

[5] Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *CoRR*, abs/1709.04326, 2017.

[6] Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *CoRR*, abs/1706.05125, 2017.

[7] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256, May 1992.

[8] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016.

[9] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *CoRR*, abs/1705.11192, 2017.

[10] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. *CoRR*, abs/1804.03980, 2018.