Achieving efficient human-like communication in multi-agent reinforcement learning using deep learning

Gabriel Ducrocq

November 2017

1 Introduction

The ability to communicate effectively in order to achieve goals in a cooperation setting is often thought as one of the marker of true intelligence [2], or at least as a highly desirable goal in developing useful intelligent machines. For example in [1], the authors argue that such ability is one of the two key components required for a machine to autonomously make itself helpful for humans.

The last decade saw spectacular improvements in natural language processing and in machine learning in general, leading to more and more efficient communication systems. From machine translation to Question and Answer bots, the progresses are impressive. See the use of LSTM neural networks for machine translation [3] or the design of a Question Answering system [4]. It is important to notice that much of these improvements take place in a supervised learning setting - sometimes in a reinforcement learning setting - and make extensive use of gigantic, static data-sets. These approaches of natural language processing are mainly concerned with optimizing some measure of linguistic intelligence.

Despite these successes, critics have been expressed against this way of conceptualizing natural language processing and communication. It is argued that methods relying on large data-sets of human language, in spite of discovering statistical patterns in a language, are unable to capture its functional aspects [5]. This represents a severe hindrance for true human-machine or machine-machine cooperation.

To solve this problem, a new paradigm of machine language learning has emerged, based on a utilitarian definition of language comprehension [2]: an agent is said to understand language only if it is able to reach goals using language. Language being one of several tools an agent can use to achieve its aim, instead of an end in itself.

This shift in view leads to a shift in the set of the most relevant methods. Instead of using supervised algorithms on static data-sets, multi-agent reinforcement learning for cooperative tasks is used to ground the language in an

environment: several agents, only partially observing their environment, need to exchange information through a communication channel to achieve their shared goal.

In addition to ground the language, recent developments have tried to make this naturally emerging language easier to interpret by humans. Several ways have been used to achieve a better interpretability like discretising communication [5], [7]), enforcing the development of a compositional language [5], enforcing the use of "English-like" tokens [7], [9] etc....

In the light of the recent successful experiments, I would like to answer the following question during my PhD:

Is it possible to achieve effective human-machine cooperation using language in a complex environment using the framework described above ?

I will first review the work done in this direction. Then, I will provide directions that can potentially answer the question, and I will make a proposal for a Dota 2 as a case study. Following this section is a schedule for my PhD.

2 Related works

In this section, the recent related papers will be reviewed. The reader has to keep in mind that only the works making use of multi-agent reinforcement learning in a cooperative setting will be cited here. Furthermore, several axis are important for the development of a language easily interpretable and thus useful for human-machine cooperation: the nature of the language - discrete or continuous -, its compositionality and the use of "English words". I am going to constantly refer to these three components during this review.

While necessary for easy human-machine communication through language, the discreteness of the learned communication protocol is not required for effective communication between artificial agents. Indeed in [6], the authors develop an actor critic algorithm in a multi-agent setting: the policy network (actor) and the Q-network (critic) are both a bi-directional recurrent neural network over agents, thus allowing them to communicate through the internal layers of the bi-directional recurrent network. Though effective - the agents were able to master human-level strategies - the communications between agents are not readily interpretable.

The emergence of a grounded language has also been studied in simpler context with simpler policies. For example in [7] a simple referential game is used to make a language between two agents - here two simple feed-forward neural networks - emerge: both the sender and the receiver networks observe the same two images, but only the sender knows which one is the target image. The sender then sends a vocabulary token - one hot encoded - from predefined vocabulary to the receiver which must guess which of the two images is the target. If it guesses correctly, both of the agents receive a reward of 1, 0 otherwise. The agents successfully develop a discrete communication protocol. Moreover, in order to make the language created by the agents more interpretable, the sender switches

equiprobably between playing this game and the task of supervised classification of the images, hence grounding their communication in human language.

A slightly different version of the referential game as well as different policies can be found in [8]. In this paper, the policies of the sender and the receiver are both LSTM networks. The sender observes a target image and outputs a message of arbitrary length composed of sampled tokens from a predefined set of tokens (vocabulary). The receiver observes this message and its hidden state is used to compute the reward. While the agents successfully develop a language, this one is hierarchical, making it impractical for efficient human-machine collaboration.

Another way to make the language more "human-like" and interpretable is designed in [9]. In this work, the authors develop a questioner and an answerer bots in the context of a guessing game: the answerer bot observes an image and the questioner bot asks questions about it to the answerer in order to output the "best description" of the unknown image. This process is repeated several rounds, each agent accessing the history of the conversation at each round. The questions and answers are encoded using LSTM neural networks. The strategy of the author to ensure that the agents' utterances do not deviate from English is to pre-train their model in a supervised way on a human dialog data-set.

In a simpler environment, the authors of [10] modelize their agents as LSTM networks - through their DDRQN architecture, allowing them to take into account the utterances of other agents.

Finally, Igor Mordatch and Pieter Abbeel also used the multi-agent reinforcement learning framework to foster the emergence of a ground-based and compositional language [5]. They immersed several agents in a simple two-dimensional world with landmarks - characterized by their position, color and shape - and assign them to cooperative tasks, allowing the agents to utter, at each time step, a token from a predefined vocabulary set. Because their agents continually emit a stream of utterances over time, the authors added a memory mechanism to the agent's policies so that they can capture the meaning of the stream over time [5]. Their experiment was successful in the sense that a compositional language emerged between agents. Furthermore, as the environment was fairly simple and the vocabulary size was small - limited to twenty different symbols, the learned language was relatively easy to interpret by humans. However, each symbol is one-hot encoded. This can pose a problem of scalability if the vocabulary size is largely increased.

Aside from this general presentation of related works, several tricks have been used to ensure an easier training of the designed systems. For example, we could cite the use of parameters sharing between agents ([11], [5]), the design of end-to-end differentiable architecture within agents ([11]), across agents [11] or even across agents and through time [5].

Moreover, it should be noticed that some of these works assumes partial observability ([11], [10], [5]) in order to encourage inter-agent communication.

In spite of these recent successes in the emergence of communication in a cooperative setting, the ability of the designed systems to efficiently collaborate

with a human agent in a complex environment is yet to demonstrate. Indeed, some research demonstrated the efficiency of a learned communication protocol for a cooperation task in a complex environment [6] but without enforcing the development of an easily interpretable language, preventing human to collaborate with the artificial agents, or even to readily understand their choices.

On the contrary, other works emphasized the development of "human-like" language in a simple environment ([5], [11]) and sometimes with only one time-step ([7], [10]), achieving the emergence of a communication protocol easy to interpret.

In the next section are highlighted potential directions to achieve human-machine cooperation through human-like language in a complex environment.

3 Proposed directions

Based on the review of the last section, several ingredients still need to be put together in a complex environment: grounded concepts, compositionality, "English-like" words. Drawing on the recent success of [5], the different architectures used in ([9], [8]) and the mentioned tricks, I propose three steps to achieve the proposed goal.

3.1 Step 1: outputting messages at the character level

Most of the cited works deal with discrete vocabulary size using one-hot encoding ([5], [9], [8]), posing a scalability issue. This could be a major hindrance if we were to design a system that can efficiently collaborate with human through language. Indeed, English is composed of more than a hundred-thousands words. Though it would not be a problem for agents collaborating with humans on very specific tasks, needing only a limited vocabulary, this would represent an obstacle to general-purpose collaborative agents.

To avoid this problem it could be wise to design agents generating their utterances at the character level instead of the word level. For example, English alphabet is only twenty-six character long - leading to a one-hot encoding of size 26 only -, a small number compared to the possible number of words composed with this alphabet. To do so and taking inspiration from [9], two recurrent networks could be integrated to the agents' policies - one for encoding, the other one for decoding the message - outputting several characters at each time-step. This architecture is not without problems: to ensure the compositionality of the learned language, the vocabulary size has to be much smaller than the number of concepts describable [5]. In a fairly simple environment, allowing a recurrent network to compose two characters from a character set of size 26 would allow a vocabulary of size 676. This is even worse when we consider character's sequence of arbitrary length. Hence, in preliminary experiments, the number of different characters and the maximum length of the sequence will need to be chosen carefully.

Another, more experimental way to to explore in order to generate characters, is to use a continuous communication channel, adding noise to create a kind of discretisation like in [11]. In this way we would completely avoid the one-hot encoding of character vocabulary.

3.2 Step 2: uttering "English-like" words

Once we can make a compositional language emerge naturally between agents, the next step is to enforce the use of English-like words, relieving human agents from the burden of interpretation. This can be done by exposing the agent to human languge [5]. Several ways to achieve this have been suggested or tried: in [2], Jon Gauthier and Igor Mordatch suggest the integration of some fixed-language agents that speak a conventional language like English to the environment. We could also pre-train the encoding and decoding modules in a supervised fashion like in [9] or use supervised training for these modules at the same time the agents are train in the environment like in [7].

However, the pre-training of a communication module leads to the loss of the end-to-end characteristic of the learning. The different options need to be carefully tried and weighted.

3.3 Step 3: scaling

Once the two previous steps carefully explored in a more or less simple settings, the next stage is to scale the potentially successful architectures: testing them in more complex environments, requiring larger vocabulary size and a more complex syntax, like suggested in [5]. A cooperative video game involving partial observability of the environment would be best suited for this step. This would allow us to see if the emerged language is still compositional and does not deviate from English. In case of success, we could remove one of the artificial agent and replace it by a human-agent, demonstrating efficient human-machine collaboration using language.

4 Dota 2: a case study

In the past few years, video games have been used as environments to benchmark the results of reinforcement learning algorithms in complex settings and to compare the achieved scores to human performance ([12], [6]). Indeed, video games represent a complex environment with a rich set of concept to understand. Moreover, the variety of possible actions allow for the development of sophisticated strategies.

In order to test the architectures developed in step 1 and 2, we propose to use Dota 2 as environment. This game is a cooperative/competitive game: two teams of at most five players try to defeat each other. The game is won by the team that succeeds in destroying its target: a building situated in the enemy camp. In order to achieve this, each player controls a hero of its choice - among

113 possibilities, two players of the same team cannot choose the same one - having specific abilities and cooperates with its team members.

This video game seems ideal to test the ability of the designed architecture to achieve efficient cooperation in a complex environment while keeping an "English-like" language. A 2 vs 2 setting seems the best to begin with.

However, the observations are yet to define. They can be raw pixels from the screen, or specific metrics, like position of the players on the map, life remaining etc...

In the next section a schedule for the PhD is detailed.

5 Schedule

In this section a schedule of the research plan for my PhD is established. As three steps has been proposed, it is natural to detail this schedule in three parts, each one corresponding to a step.

5.1 Step 1: First six to eight months

The first step is about using recurrent neural networks in order to output messages at the character level, avoiding scalability issues. At this stage, the goal is to ensure that we can use a recurrent neural network (RNN) to encode and decode messages, while generating them at the character level and keeping the compositional structure of the learned language.

As the use of RNN to encode/decode messages using a discrete vocabulary has been previously successful ([8], [9]), one can be confident that the use of RNN to output messages at the character level will work. Moreover, the compositionality of the language essentially depends on the vocabulary size as well as the richness of the concepts describable in the environment [5]. Thus, one can think that the success of this step will reside in the tweaking of these two characteristics more than in the RNN architecure.

Drawing on the recent successes of RNN for the emergence of language, the first six to eight months may be sufficient to achieve the first step.

5.2 Step 2: the next one year

The second step is to enforce the use of "English-like" words. This part is much more exploratory than the previous one.

Some recent work have attempted to foster the emergence of a language that does not deviate "too-much" from plain English ([9], [7]). However, such experiment usually take place in a very simple environment, sometimes involving only one time step. Enforcing the use of "English-like" words and syntax has not been done in more complex - but still quite simple - environment like in [5].

As it is still unexplored, it is arduous to assess the time needed to end this step. Moreover the success of this step depends on what we consider a "success": what degree of similarity with plain English is considered sufficient ?

Given the highly exploratory characteristic of the step, about a year must be dedicated to this work, maybe more.

5.3 Step 3: one year and a half

Assuming a success in the two previous steps, the third one is to scale the experiments and make a case study using the video game Dota 2: a success would be the discovery of "human-like" strategies using "human-like" communication, allowing a human agent to understand the communication of the artificial agents at once. An extra-step would be to replace one of the artificial agents by a human one, demonstrating easy human-machine cooperation.

A success at this stage will require to keep the compositionality and the use of "English-like" words and grammar while complexifying the environment. This has not been achieved yet so it is highly speculative for the moment. It is also hard t have an idea of the difficulty of this step given a success on the two previous ones: the transfer from a simple environment to a complex one may be straightforward or not.

One year and a half should be dedicated to this step.

6 Conclusion

The understanding of human language is a key feature needed by machines as this will allow them to cooperate efficiently with humans. Recently, some critics have been expressed against the supervised approach of human language learning: it is argued that such supervised algorithms cannot capture the functional aspects of language and are limited to the discovery of statistical patterns.

Based on the utilitarian definition of language understanding - an agent understands a language if it is able to reach goals using that language - a new framework has been proposed. This framework relies heavily on multi-agent reinforcement learning algorithm in a cooperative setting.

While recent works in the field have been successful, each one is showing different key ingredients: some achieve compositionality and interpretability in a simple environment, other works achieve interpretability and efficiency in a simple setting while different works achieve efficiency in a complex environment, at the cost of the "human-like" characteristic of the language as well as its interpretability.

Finally, I proposed to follow a three step plan during my PhD to put these pieces together and to demonstrate efficient human-machine cooperation in a complex environment, applying the designed algorithm to the 2 vs 2 setting of Dota 2.

References

[1] Tomas Mikolov, Armand Joulin, Marco Baroni *A Roadmap towards Machine Intelligence.* https://arxiv.org/abs/1511.08130

[2] Jon Gauthier, Igor Mordatch *A Paradigm for Situated and Goal-Driven Language Learning* https://arxiv.org/abs/1610.03585

[3] Ilya Sutskever, Oriol Vinyals, Quoc V. Le *Sequence to Sequence Learning with Neural Networks* https://arxiv.org/abs/1409.3215

[4] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, Xiaoming Li *Neural Generative Question Answering* arXiv:1512.01337v4 [cs.CL]

[5] Igor Mordatch, Pieter Abbeel *Emergence of Grounded Compositional Language in Multi-Agent Populations* arXiv:1703.04908v1 [cs.AI]

[6] Peng Peng, Ying Wen, Yaodong Yang, Yuan Quan, Zhenkun Tang, Haitao Long, Jun Wang *Multiagent Bidirectionally-Coordinated Nets* arXiv:1703.10069v4 [cs.AI]

[7] Angeliki Lazaridou1, Alexander Peysakhovich, Marco Baroni *Multi-agent cooperation and the emergence of (natural) language* arXiv:1612.07182v2 [cs.CL]

[8] Serhii Havrylov, Ivan Titov *Emergence of language with multi-agent games: learning to communicate with sequences of symbols* arXiv:1705.11192v2 [cs.LG]

[9] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, Dhruv Batra *Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning* arXiv:1703.06585v2 [cs.CV]

[10] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, Shimon Whiteson *Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks* arXiv:1602.02672v1 [cs.AI]

[11] Jakob N. Foerster, Nando de Freitas, Yannis M. Assael, Shimon Whiteson *Learning to Communicate with Deep Multi-Agent Reinforcement Learning* arXiv:1605.06676v2 [cs.AI]

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller *Playing Atari with Deep Reinforcement Learning*