ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Language Grounded Task-Adaptation in Reinforcement Learning

Matthias Hutsebaut-Buysse, Kevin Mets and Steven Latré

Department of Computer Science, University of Antwerp - imec
Sint-Pietersvliet 7, 2000 Antwerp, Belgium

Abstract. Over its lifetime, a Reinforcement Learning agent is often instructed to perform different tasks. How to efficiently adapt a previously learned control policy from one task to another, remains an open research question. In this paper, we investigate how instructions formulated in natural language can enable faster and more effective task adaptation. Our proposed method is capable of assessing, given a set of developed base control policies, which base policy will be the most qualified to adapt to a new unseen task.

1 Introduction

Reinforcement Learning (RL) solves sequential decision-making problems by utilizing a trial-and-error approach guided by a reward signal. RL has achieved tremendous successes, especially in beating humans in video games [1] and robotics [2]. However, RL also suffers from various open problems, such as its sample inefficiency. This sample inefficiency is often caused by the reward function-specification. On the one hand, a sparse and delayed reward signal makes it difficult for the agent to experience any meaningful feedback. On the other hand, designing tasks with a dense reward signal, is often a complex endeavor, and regularly exhibits unwanted side effects [3].

A recent line of research [4], has proposed methods that allow task descriptions to be specified using natural language. A commonly used approach consists of directly embedding both visual observation and language instruction in order to train a control policy [5, 6, 7]. Alternatively, [8] uses natural language reward shaping, by predicting if an action in a trajectory matches a task description. [9] explores the compositional structure of natural language in order to learn abstractions capable of generalizing over different sub-tasks using language instructions. Unfortunately, these methods have proven to still be very sample inefficient, requiring weeks of training in simulations, in order to learn relatively simple tasks.

In this paper, we propose a natural language guided transfer learning method. Our method can make RL methods informed by natural language more sample efficient, requiring less interaction with the environment. We achieve this by providing a viable way of allowing an agent to efficiently adapt previously learned knowledge, to a new previously unseen task. Current algorithms capable of quickly adapting their control policies to solve related tasks, mostly rely on intensive training using a diverse set of tasks [10], often guided by a handcrafted curriculum of increasingly more difficult and diverse tasks. Our method

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

does not require such extensive training, and is capable of, given a small set of pre-trained policies, to make decisions about which previously developed policy will adapt best, in order to solve a new previously unseen task, solely from its task description formulated using natural language.

2 BabyAI environment

In order to demonstrate our method, we make use of the *BabyAI environment* [7]. In this environment, the agent is tasked with completing various tasks in a multi-room 2D gridworld. For our experiments, we consider a single room, and test our method on the *goto* and *pickup* problems. The task the agent is charged with, is described using a synthetic *baby language*. Instructions we use in our transfer experiments follow the same $\langle$*verb*, *object color*, *object*$\rangle$ pattern (e.g. *pickup the yellow box*).

The pixels of the screen, together with this instruction, form the fully observable state-space $\mathcal{S}$. The action-space $\mathcal{A}$ consists of movement, handing objects and opening doors. The reward-signal is only sparsely observed.

3 Method

The main idea of our approach is to start with a limited set of pre-trained parameterized base control policies. When confronted with a new task, described using natural language (the transfer instruction), the best base policy is selected, and the new task is learned more efficiently, based on the parameters of this base policy. A summary of our method can be found in Algorithm 1.

Algorithm 1: Summary of our task-adaptation method.

```
1 $\alpha$: $k$ instructions sampled from the set of possible instructions $Z$
2 $\beta$: $p$ instructions sampled from the set of possible instructions $Z$
3 foreach instruction $i \in \alpha$ do
4     Train base policy $\pi_i$ until convergence
5     foreach instruction $j \in \beta$ do
6         Sample task-adaptation $\pi_i^j$ during $n$ training steps. A new policy
            $\pi_j$ is developed starting with the model parameters from base
            policy $\pi_i$.
7 Train the transfer model
```

3.1 Pre-training base control policies

In the pre-training phase, we train k base control policies $\{\pi_0, ..., \pi_k\}$. A single base control policy $\pi_i(s_t)$ determines the action $a_t \in \mathcal{A}$ an agent takes, based on the state $s_t \in S$ the agent resides in. Each base control policy should reliably be able to perform one instruction i. This task instruction is expressed in natural language (e.g. *go to the blue ball* or *pickup the yellow key*).

Training base control policies can be done using any RL algorithm, capable of learning and forgetting new tasks. The amount of pre-trained control policies

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

should be sufficiently large, but smaller than the entire set of possible instructions ($k \ll |Z|$).

Our method can be used with a fixed number of base control policies, which are trained during a single pre-training phase. Additionally, our method can also be extended to work in an iterative fashion. In this iterative approach, the agent starts with a small set of k pre-trained base control policies. When confronted with a new task, our method determines the best base control policy to facilitate task-adaptation (e.g. π_i). After training the new policy π_j using the model-parameters from the selected base control policy π_i, the resulting policy π_j can be added to the set of base control policies. This will allow executing more efficient task-adaptations, as more base control policies become available.

We select k instructions $\{z_0, ..., z_k\}$ to train base policies $\{\pi_0, ..., \pi_k\}$, from a uniform random distribution. However, an interesting extension to this research might be to select base control policies based on a more advanced selection criterion, such as maximizing distance between the task instructions in a prior language-embedding.

3.2 Sampling task-adaptations

The second phase of our method consists of utilizing the developed base control policies, in order to sample a limited number of task-adaptations. A single task-adaptation sample π_i^j consists of taking a fully developed base control policy π_i, and using it to perform a new instruction j, different from the one it was trained on. An example of such a sample would include to start from a policy trained on an instruction *go to the yellow box*, and utilize it to perform a different task, such as *pickup the yellow box.*

A task-adaptation sample is performed by loading the parameters of the base policy as the initial parameters of the new policy. Training can be performed using any RL algorithm. During this sampling phase the policy does not need to converge. Training only needs to happen for a limited number of n steps. This amount of required steps is significantly lower than fully developing the policy. After the sampled task-adaptation has been executed for n steps, we measure the performance. This can be done by, for example, calculating the success rate of the agent satisfying the instruction over the last 100 iterations. For each base control policy, we randomly select p different tasks from Z to sample task-adaptation.

In summary, in this phase, to develop our dataset, we run $k \times p$ task-adaptation samples (for n training steps). The resulting policies, which are only partially developed can also potentially be used.

3.3 Training the transfer-model

The final stage of our method consists of training a binary classification model: $f(z_x, z_i, z_j) \rightarrow \{1, 0\}$. This model is capable of generalizing the perceived task-adaptation over unseen adaptations, solely from using the task descriptions, formulated in natural language.

Instruction z_x	**Transfer instruction z_i**	**Transfer instruction z_j**	**Class**
Goto the green key	Pickup the red ball	Goto the yellow box	1
Goto the red ball	Pickup the red ball	Goto the yellow box	0

Table 1: Example input dataset, used to train the transfer-model.

The input of the proposed model (Figure 1) consists of a concatenation of the sampled transfer instruction z_x, combined with the instructions attached to two candidate base policies (z_i and z_j). The output of the model consists of a single binary output. This output is trained to be 1, if the first base policy with instruction z_i will adapt better than the second base policy satisfying instruction z_j. An example dataset is presented in the Table 1.

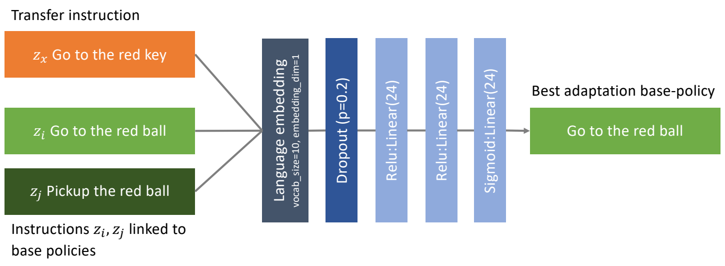

Fig. 1: Transfer-model architecture.

In order to work directly with instructions in natural language, a language embedding is used. This embedding is trained end-to-end, and thus is specifically trained to encode instructions based on their transfer capabilities.

3.4 Transfer-model usage

The resulting transfer-model can be used when the agent is confronted with a new task description, it currently has no developed policy for. Given a set of labeled base policies, and the new task description, the various possibilities can be tested in order to make an assessment of which base policy will result in the fastest task-adaptation.

4 Experiments

In order to find out whether patterns can be discovered in task-adaptations using instructions expressed using natural language, we initially performed a large set of 636 transfer experiments in the BabyAI environment.

This initial experiment taught us that complex relations between parts of instructions exist. For example in Figure 2 the importance of the verb in the task description is presented.

However, the discovered relationships between task instructions and transfer capabilities were not straightforward, thus in our second experiments we exam-

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

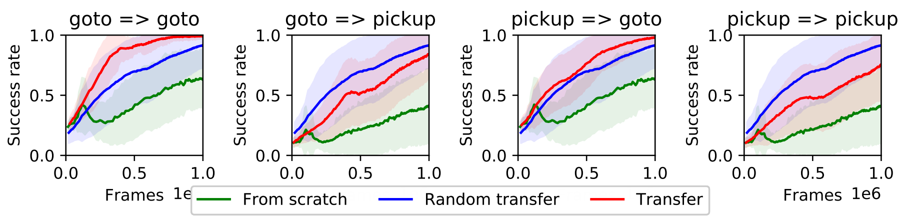

Fig. 2: Comparison of how well different base control policies adapt to new tasks, based on whether the verb in the instruction is the same or different. The solid lines represents the mean, the shading represents the standard deviation. Measured over 636 different transfer tasks.

ined if our proposed classification model could successfully generalize over the sampled task-adaptations. In order to do this, we trained different amounts k of randomly sampled base control policies. While training can be done using any RL algorithm, we used DQN [11] in our experiments. Training a base control policy is done using at least 1M steps, and ends when the policy achieves a success rate of at least 95%, measured on the previous 100 iterations.

After developing k different base control policies, we sampled p adaptations for each base control policy (n=100,000). The results gathered from these task-adaptations were used to train the transfer model (Adam lr= 0.001, 1M steps).

In Table 2, we present the performance of our model, using various numbers of base control policies (k), and different numbers of task-adaptation samples (p). We measure model accuracy over a holdout-set consisting of all possible expressible task-adaptations not seen during sampling. This accuracy measures the percentage that our model selected the best base policy (2M steps).

	p=8	**p=10**	**p=12**	**p=14**	**p=18**	**p=20**
k=8	0.61 ±0.03	0.62 ±0.03	0.61 ±0.05	0.64 ±0.05	0.65 ±0.02	0.66 ±0.03
k=10	0.62 ±0.03	0.62 ±0.05	0.64 ±0.06	0.62 ±0.04	0.66 ±0.03	0.67 ±0.02
k=12	0.67 ±0.02	0.67 ±0.01	0.66 ±0.02	0.67 ±0.02	0.68 ±0.02	0.66 ±0.04
k=14	0.64 ±0.04	0.66 ±0.02	0.67 ±0.03	0.69 ±0.01	0.69 ±0.03	0.68 ±0.01
k=18	0.67 ±0.03	0.68 ±0.02	0.68 ±0.03	0.71 ±0.01	0.70 ±0.02	0.71 ±0.02
k=20	0.69 ±0.01	0.68 ±0.05	0.70 ±0.02	0.69 ±0.04	0.71 ±0.03	0.71 ±0.03

Table 2: Accuracy of the task-adaptation classifier model. The different rows represent the various amount of base control policies used during training (k), the columns represent the amount of task-adaptations (p) sampled for each base control policy. Results are averaged over 5 runs.

Our results show that even with a limited number of k base control policies, and p sampled task-adaptations, a transfer model can be developed. There is still room for improvement regarding the accuracy of the model, however the stochastic nature of RL, makes task-transfer inherently noisy.

Nevertheless, efficient task-adaptation realized by our method, proves to be a quintessential building block in a lifelong learning setting [12].

5 Discussion

In this paper, we presented a method capable of predicting, given a set of base control policies, labeled using natural language, which of these base control policies will adapt the fastest to a new previously unseen task. In order to make assessments about task-adaptation, our method uses a for this task specifically trained language embedding as part of an end-to-end binary classification model.

Our preliminary results experimentally demonstrate that this approach is capable of assessing adaptation performance, solely from task descriptions. When confronted with an expanding set of tasks in a lifelong-learning setting, our method has the potential to vastly improve sample efficiency.

Acknowledgements

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen“ programme.

References

[1] O. Vinyals, I. Babuschkin, and W.M. Czarnecki et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[2] OpenAI. Learning dexterous in-hand manipulation. *IJRR*, 39(1):3–20, 2020.

[3] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML99*, 1999.

[4] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A Survey of Reinforcement Learning Informed by Natural Language. In *IJCAI19*, 2019.

[5] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

[6] Dipendra Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. *arXiv preprint arXiv:1704.08795*, 2017.

[7] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. In *ICLR19*, 2018.

[8] Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. Using Natural Language for Reward Shaping in Reinforcement Learning. In *IJCAI19*, 2019.

[9] Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. Language as an Abstraction for Hierarchical Deep Reinforcement Learning. *arXiv:1906.07343 [cs, stat]*, 2019.

[10] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task Deep Reinforcement Learning with PopArt. In *AAAI19*, 2019.

[11] V. Mnih, K. Kavukcuoglu, and D. Silver et al. Human-level control through deep reinforcement learning. *Nature*, 518, 2015.

[12] Daniel L Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*, 2013.