



HAL
open science

Framing Lifelong Learning as Autonomous Deployment: Tune Once Live Forever

Eneko E. Agirre, Anders Jonsson, Anthony Larcher

► **To cite this version:**

Eneko E. Agirre, Anders Jonsson, Anthony Larcher. Framing Lifelong Learning as Autonomous Deployment: Tune Once Live Forever. International Workshop on Spoken Dialogue Systems Technology, Apr 2019, Siracusa, Italy. hal-02280158

HAL Id: hal-02280158

<https://hal.science/hal-02280158v1>

Submitted on 6 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Framing Lifelong Learning as Autonomous Deployment: Tune Once Live Forever

Eneko Agirre, Anders Jonsson, Anthony Larcher

Abstract Lifelong Learning in the context of Artificial Intelligence is a new paradigm that is still in its infancy. It refers to agents that are able to learn continuously, accumulating the knowledge learned in previous tasks and using it to help future learning. In this position paper we depart from the focus on learning new tasks and instead take a stance from the perspective of the life-cycle of intelligent software. We propose to focus lifelong learning research on autonomous intelligent systems that sustain their performance after deployment in production across time without the need of machine learning experts. This perspective is being applied to three European projects funded under the CHIST-ERA framework on several domains of application.

1 Introduction

In the context of Artificial Intelligence, intelligent systems based on machine learning rely on human experts across all phases of the software. According to the de-facto standard CRISP-DM methodology (Cross-industry standard process for data mining [1]), the life-cycle of such software includes a development cycle with interactions to define the business domain, understand and prepare the data, select and train appropriate models, and evaluate the software. After several iterations, development is halted and the system is deployed in production.

Eneko Agirre
IXA NLP Group, University of the Basque Country (UPV/EHU), Manuel Lardizabal 1, 20018 - Donostia, Basque Country, e-mail: e.agirre@ehu.eus

Anders Jonsson
Dept. Information and Communication Technology, Universitat Pompeu Fabra, Roc Boronat 138, 08018 Barcelona, Spain, e-mail: anders.jonsson@upf.edu

Anthony Larcher
LIUM - Le Mans Universit, France e-mail: anthony.larcher@univ-lemans.fr

Such systems are constructed, developed and trained until a certain point of time, and once deployed for production in the target environment they need to be continuously monitored to check that its performance in real life matches the one in development. Current intelligent systems are not able to adapt to situations outside their development conditions. For this reason, the flow of real data and system responses needs to be monitored, to ensure that it follows the same distribution as the data used in development, as otherwise the performance of the intelligent system will degrade with time. Usually human monitoring from field experts is needed to detect whether performance degrades, and at that point, development experts (e.g. machine learning experts) are needed to revise the system and adapt it to the new conditions. In short, intelligent systems are not able to learn from the environment and improve themselves without intervention from their initial developers.

Lifelong Learning in the context of Artificial Intelligence¹ is a new paradigm that is still in its infancy. It is commonly defined as agents that are able to learn continuously, accumulating the knowledge learned in previous tasks and using it to help future learning [3]. The concept of "new task" is not well defined, as it has been used for different phenomena like new instances unseen in training, related tasks or new domains, but it is generally demonstrated on toy tasks [4, 2]. We depart from the focus on learning new tasks and, instead, take a stance from the perspective of the life-cycle of intelligent software that addresses real tasks. We propose to focus lifelong learning research on autonomous intelligent systems that sustain their performance across time *after deployment* without the need of machine learning experts, following the "*Tune once, live forever*" motto. We think that such a capability of incremental autonomous learning, or lifelong learning, is key to the development of truly autonomous intelligent systems, as argued in the CHIST-ERA 2016 call for projects².

In this position paper we first detail our proposal, and then mention how we plan to address this issue in three European projects funded under the CHISTERA framework.

2 Lifelong Learning for autonomous deployment

Given a task, the final goal is to design a system which is able to continue learning after deployment, adapting itself to the changes in the distribution of the input data when necessary, without the need of further development. Figure 1 shows two possible scenarios. In the first scenario (left of figure) the lifelong learning system (LL system for short) is able to learn from a stream of data over time, improving the performance over time. The right side of the figure shows a more realistic scenario, where, at some point of time, there are changes in the distribution of the input data, and the system performance drops. The LL system (in green) will be able to

¹ Also called continual learning [2], among other terminological options

² <http://www.chistera.eu/sites/chistera.eu/files/CHIST-ERA%20Call%202016%20-%20Call%20Announcement.pdf>

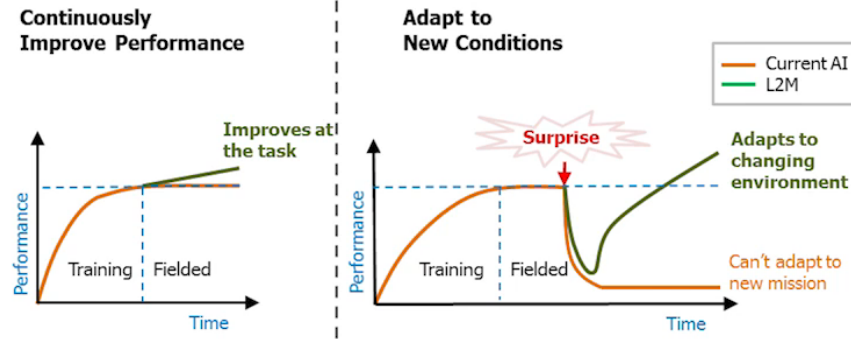


Fig. 1 Lifelong learning under two scenarios. In the left, the distribution of input data does not change with time after deployment. In the right, the distribution of input data changes after deployment. Source: L2M DARPA program <https://www.darpa.mil/staff/dr-hava-siegelmann>

recognize the change in the environment and take actions to adapt and improve performance, reaching, or even surpassing, the original performance level. The figure also shows a standard system (in yellow), which is not able to adapt.

We envision that such LL systems need to solve two key challenges:

Auto-evolution: An LL system needs to identify relevant new information that was not present at the time of development, and integrate this information into its learning model. An LL system also needs to balance the importance of old and new data, since an excessive focus on old data may prohibit the system from adapting to changes, while an excessive focus on new data may lead to catastrophic forgetting.

Auto-evaluation: An LL system needs to adapt to changes in the underlying input distribution, which may require modifying the objective function. For example, in supervised learning a system needs to automatically select new inputs and associate labels with these inputs, which implicitly modifies the objective function to include a measure of how well the system covers these new examples.

Although the final goal is a fully autonomous system, a weaker system which only requires field experts would also represent an important breakthrough, as it would lower the reaction time and costs of the current model, where the interaction with developers is also needed. In this weaker case, field experts might help recognizing the degradation of performance, gathering relevant adaptation data and label it when necessary.

Another key challenge is evaluation. Metrics and protocols for traditional machine learning task bench-marking have been thoroughly developed for decades through intensive efforts and numerous evaluation campaigns. Adapting them to match the LL paradigm is a necessary step to catalyse the development of autonomous systems and provide means of reproducible and fair evaluation.

We have applied the perspective outlined in this section to three European Projects: ALLIES³, DELTA⁴ and LIHLITH⁵.

3 Approach in three heterogeneous projects

In this Section we present three different approaches for LL systems as envisioned in three projects: ALLIES, DELTA and LIHLITH.

The ALLIES project envisions LL systems guided by domain experts through active learning and continuous supervision. The human in the loop is expected to provide a safeguard across time but also make sure the adaptation policy is matching the customer needs. A key aspect of ALLIES lays in the supervised evaluation of such systems which performance must be considered according to the human effort provided across time. Implementations of machine translation and speaker diarization autonomous systems will help design the requested metrics and protocols. The main outcomes of the project consist of a secured platform to evaluate autonomous systems in a reproducible manner and two bench-marking campaigns that will validate the evaluation scenarios.

The DELTA project aims at developing novel algorithms for lifelong reinforcement learning, an area of machine learning that studies sequential decision making. Concretely, the objective of DELTA is to address three key problems related to lifelong reinforcement learning, and develop solutions that improve on the state-of-the-art: 1) plan high-quality sequences of actions; 2) efficiently explore the environment; and 3) decompose the overall task into subtasks. Auto-evolution will be tested by changing observable variables and available actions, and measuring how robust a system is to such changes. Auto-evaluation and benchmarking are already part of reinforcement learning, in the sense that the evaluation metric takes the form of a reward signal that the system has to learn to maximize. Since reinforcement learning is online by nature, a system constantly improves over time, and the balance of old and new data is controlled by tuning a learning rate. However, the reward is usually defined by a human expert, while in the lifelong setting the system itself has to define the reward associated with new, previously unseen tasks.

The LIHLITH project applies lifelong learning to the interaction of humans and machines on specific domains. LIHLITH focuses on human-computer dialogue, where each dialogue experience is used by the system to learn to better interact, based on the success (or failure) of previous interactions. The dialogues are designed to produce a reward, allowing the chatbot system to know whether the interaction was successful or not. The reward is used to train the domain and dialogue management modules of the chatbot, improving the performance, and reducing the development cost, both on a single target domain but specially when moving to new

³ <https://projets-lium.univ-lemans.fr/allies/>

⁴ <https://www.upf.edu/web/delta>

⁵ <http://ixa2.si.ehu.es/lihlith/>

domains. Contrary to other domains, in human-machine dialogue auto-evaluation is more natural, in the sense that user feedback can be naturally blended in the conversation, providing new learning instances. Auto-evolution can be triggered when detecting that negative feedback is associated with patterns in the data (e.g. change of domain). Please see [5] for a recent successful example, where simple patterns are used to get feedback in the framework of simple chatbots.

4 Relation to recent work on lifelong learning

Research in lifelong reinforcement learning has mainly focused on learning strategies for solving a set of tasks [6, 7]. However, in this setting the set of tasks is usually fixed and known. There also exists work on transfer learning that investigates how an existing strategy can be adapted to solve a new task [8]. Preliminary work in DELTA has studied how exploration can be adapted to the case for which model components change over time [9]. There are also preliminary results investigating the theoretical properties of hierarchical decomposition [10, 11], but this work assumes that the decomposition is already given rather than inferred by the system.

5 Conclusion and Discussion

In this position paper we depart from the focus on learning new tasks and instead take a stance from the perspective of the life-cycle of intelligent software. We propose to focus lifelong learning research on autonomous intelligent systems that sustain their performance after deployment in production across time without the need of machine learning experts. We briefly report how this perspective is applied to three European projects funded under the CHIST-ERA framework on several domains of application.

Acknowledgements This work has been partially funded by the ALLIES, DELTA and LIHLITH projects supported by the EU ERA-Net CHIST-ERA and the Spanish Research Agency (LIHLITH, PCIN-2017-118; DELTA, PCIN-2017-082).

References

1. C. Shearer. The CRISP-DM model: the new blueprint for data mining. *Data Warehousing*, 5:13–22, 2000.
2. Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. On Training Recurrent Neural Networks for Lifelong Learning. *CoRR*, abs/1811.07017, 2018.

3. Z. Chen, B. Liu, R. Brachman, P. Stone, and F. Rossi. *Lifelong Machine Learning: Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2018.
4. Jiwei Li, Alexander H Miller, Sumit Chopra, Marc’ Aurelio Ranzato, and Jason Weston. Learning through dialogue interactions by asking questions. In *ICLR*, 2017.
5. Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazaré, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! *CoRR*, abs/1901.05415, 2019.
6. H. Bou Ammar, R. Tutunov, and E. Eaton. Safe policy search for lifelong reinforcement learning with sublinear regret. In *Proceedings of the International Conference on Machine Learning (ICML’15)*, 2015.
7. C. Tessler, S. Givony, T. Zahavy, D. Mankowitz, and S. Mannor. A deep hierarchical approach to lifelong learning in minecraft. In *Proceedings of the AAAI conference on Artificial Intelligence (AAAI’17)*, 2017.
8. D. Abel, Y. Jinnai, Y. Guo, G. Konidaris, and M. Littman. Policy and value transfer in lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML’18)*, 2018.
9. P. Gajane, R. Ortner, and P. Auer. A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions. In *Proceedings of the ICML / IJCAI / AAMAS Workshop on Lifelong Learning: A Reinforcement Learning Approach (LLARLA 2018)*, 2018.
10. E. Brunskill and L. Li. Pac-inspired option discovery in lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML’14)*, 2014.
11. R. Fruit, M. Pirotta, A. Lazaric, and E. Brunskill. Regret minimization in mdps with options without prior knowledge. In *Advances in Neural Information Processing Systems*, pages 3166–3176, 2017.