Evaluation of Lifelong Learning Systems

Yevhenii Prokopalo[1], Sylvain Meignier[1], Olivier Galibert[2] Loïc Barrault[3], Anthony Larcher[1]
1. LIUM, Le Mans University, 2. LNE, 3. University of Sheffield
[1]firstname.lastname@univ-lemans.fr, [2]olivier.galibert@lne.fr,[3]l.barrault@sheffield.ac.uk

Abstract

Current intelligent systems need the expensive support of machine learning experts to sustain their performance level when used on a daily basis. To reduce this cost, i.e. remaining free from any machine learning expert, it is reasonable to implement lifelong (or continuous) learning intelligent systems that will continuously adapt their model when facing changing execution conditions. In this work, the systems are allowed to refer to human domain experts who can provide the system with relevant knowledge about the task. Nowadays, the fast growth of lifelong learning systems development rises the question of their evaluation. In this article we propose a generic evaluation methodology for the specific case of lifelong learning systems. Two steps will be considered. First, the evaluation of human-assisted learning (including active and/or interactive learning) outside the context of lifelong learning. Second, the system evaluation across time, with propositions of how a lifelong learning intelligent system should be evaluated when including human assisted learning or not.

Keywords: Evaluation, lifelong learning, human assisted learning

1. Introduction

Today's intelligent systems are used in many fields to process data for various tasks amongst which classification and regression (Prince, 2012; Bishop, 2006). Intelligent systems make use of a representation of the world around them, a model, that is initially learnt in laboratories under supervision of machine learning (ML) experts whose role is threefold. First, they select and label the so-called training data required to learn the model used by the system. One can expect the ML experts to select training data that is representative of what the system will be exposed to in the real world in a near future. Second, they set the numerous meta-parameters inherent to the system architecture and train the model by using development data in order to define the meta-parameters of the system and their optimal value. Third, before releasing it in the real world they benchmark the performance of the system on a third data set referred to as test data that is used to assert the generalisation of the system performance (in order to avoid overfitting). In this paper we will use the term "initialisation data" to refer to the set of training, development and test data together.

Once in the real world, intelligent systems are exposed to new incoming data. The life-cycle of those standard automatic systems, which will be referred to as static learning or isolated learning systems (Chen and Liu, 2016), is described in Figure 1-a.

At first, one can expect incoming data to be similar to the initialisation data but, across time, the distribution of this data might move away from the one of the initialisation data. This effect is known as the data set shift and causes severe performance degradation (Quionero-Candela et al., 2009) that can only be solved by requesting machine learning expert to adapt or retrain a new model. To avoid the costly need of ML experts, a trend is to develop systems that can improve themselves across time without the help of ML experts but by continuously adapting on all available data and with the possible help of a human domain expert. Such systems are referred to as lifelong learning intelligent systems (LLIS).[1] LLIS differ from static systems as they implement adaptation steps to sustain performance across time by continuously learning on new data. The life-cycle of LLIS is described by Figure 1-b.

LLIS can adapt in two modes: online and offline. In this work we define online adaptation as the action of a system which updates its knowledge (model) by learning on the incoming data it has to process. In other words, whenever asked to process new data to provide a hypothesis (or output), the LLIS takes benefits of this data to update its model. In this case, the LLIS receives the exact same amount of data than a static learning system.

When adapting offline, LLIS have access to additional adaptation data to learn from. This adaptation data can include the initialisation data for the system to process again, additional data provided by the domain expert, data automatically collected by the LLIS from various accessible sources and even already processed real world data that could be internally memorised by the LLIS. During offline adaptation, the LLIS is not asked to provide any hypothesis (or output) to the end-user and can possibly benefit from more computational resources and time. In this work we assume that online adaptation data come without any label while offline adaptation data might include labeled or unlabeled data.

Both offline and online adaptation processes can rely on unsupervised learning or human-assisted learning, including active learning (AL) with system initiative or interactive learning (IAL) with user initiative. Unsupervised learning is used to adapt the model with the incoming new unlabelled data. AL allows the system to ask the human domain expert for a correction of its hypothesis, which can in turn be used to improve the model. IAL allows the human domain expert to provide additional knowledge given the existing model and the current context.

[1] Sometimes referred to as continuous or sequential learning systems.

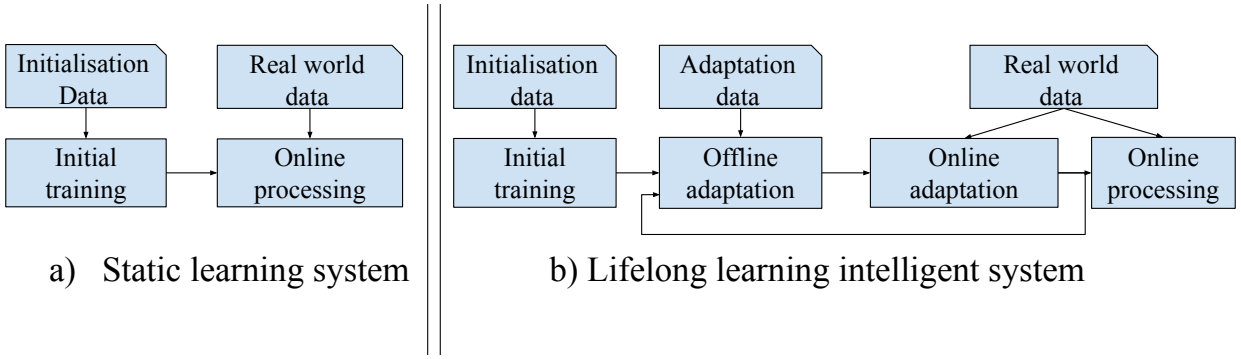


Figure 1: Comparison of static (a) and lifelong learning (b) systems life cycles. (a) Static systems use initialisation data to learn an initial model that is then deployed to process the sequence of real world incoming data. (b) For Lifelong Learning Intelligent Systems, the training data is used to learn an initial model that is then deployed to process the sequence of real world incoming data. Across time, the model used in the system can be updated via an offline adaptation process making use of adaptation data and/or an online adaptation process that learns directly from the real world incoming data.

Any system (static or LLIS) must be evaluated before being deployed for a real world application. Most of the time, this is done by running a benchmark evaluation of different systems in order to select the best one for the targeted business. Note that this work focuses only on system evaluation and not on proposing any new system architecture.
Evaluating a static system is a well known process that requires an unseen evaluation data set, a metric specific to the considered task and an evaluation protocol. For many tasks, evaluation of LLIS lacks those three elements (data set, metric and protocol) and the modification of those existing for static systems is not straightforward. The evaluation of LLIS must allow the evaluation of the overall system across time as well as the performance of the AL and IAL individual modules. Additionally, the evaluation of LLIS differs from static ones as it must take into account the data chronology. Indeed, an LLIS can be used to process new incoming data or data from the past; performance of the system in those two cases might not have the same importance for the end-user. We further introduce this as a *user policy* that must be taken into account by the evaluation metric.
In this paper, we propose generic metrics and protocols for the evaluation of human assisted learning for the case of both offline and online adaptation (see Section 2.1. and Section 2.2. respectively). We also propose a metric to evaluate LLIS that use unsupervised learning together with a user policy (see Section 3.). A metric for evaluation of LLIS that integrate human assisted learning is described in Section 3.2.. Aware that evaluation metrics and protocols are always limited and do not allow to evaluate all aspects of automatic systems we conclude this article with a discussion on our different propositions, their benefits and limitations (Section 4.).
This paper considers that static systems are evaluated with a metric similar to an error rate (lower is better), but one should notice that the proposed solutions also apply to any scalar metric without loss of generality.

2. Evaluating human-assisted learning

In the process of freeing intelligent systems from the support of machine learning experts, one can call upon human domain experts (HDE). We do not consider here any evaluation of the systems across time as this section deals with the evaluation of human assisted learning (HAL) only. HAL evaluation across time will be discussed in the context of lifelong learning in Section 3.2..

The interaction between an LLIS and a HDE can take place in two modes: offline and online.

In offline mode, a HDE interacts with the LLIS on data available to the system without time constraint, i.e. the LLIS does not have to provide a hypothesis (output) within a limited time range. In this context LLIS systems are free to learn from any available unlabelled data and might have the opportunity to ask the HDE a limited number of questions regarding this data. During this active learning process (AL), a LLIS must select the questions to ask in order to maximize the generalisation of the answer. Section 2.1. describes a way to evaluate offline active learning.

In online mode, the HDE is part of the production line in the sense that the interaction with the LLIS takes place during the time lapse between the input of the data to process and the output of the hypothesis. Although it is not mandatory, we consider in this work that in online mode, an LLIS only interacts with the HDE regarding the incoming data to process, i.e., the data for which an output hypothesis is currently expected. In online mode, given a batch, X, of data to process, the LLIS is also allowed to perform active learning (i.e. ask questions to the HDE). Additionally, it is also possible for the HDE to monitor the hypothesis generated by the LLIS and to provide corrections on this hypothesis to the LLIS in order for it to learn from the correction, re-process the same batch X and hopefully output a better hypothesis. This adaptation process, called interactive learning (IAL) is really beneficial if the system is able to generalise a small amount of information provided by the HDE. In the case of IAL, evaluation of LLIS consists of evaluating the ability of the system to ask relevant questions to the HDE, to generalise the received informa-

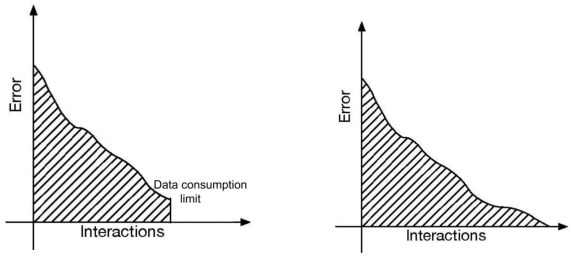
tion and eventually evaluating the final performance of the LLIS. Section 2.2. introduces a novel method to evaluate online HAL

In order to guarantee the fairness of evaluation, a human domain expert simulator is used in order to allow reproducibility of both offline and online adaptation processes.

2.1. Evaluating offline human-assisted learning

The efficiency of human assisted learning lies in its ability to improve the performance of the system while minimizing the cost of human interactions.

In the literature, many documents advise to measure the quality of this adaptation process by considering the dependence between cost of interactions and performance of the system (Krogh and Vedelsby, 1995; Siddhant and Lipton, 2018; Drugman et al., 2019; Beluch et al., 2018; Pérez-Dattari et al., 2018; Celemin and Ruiz-del Solar, 2019). This approach is popular and we do not propose any new proposal, but only report here this measure as a way to evaluate offline human assisted learning. Given an initial model, an iterative questions/answer interaction is initiated by the system with the goal of reducing the error rate on a given dataset. The system can potentially ask questions related to any document it has access to: the adaptation set, and its performance is evaluated on a dataset it is not aware of and that can be different from this adaptation set. In this context, there is no certainty that the error rate on the evaluation set will reduce and reach zero along the active learning process. For this reason, a maximum cost of interaction must be defined for evaluation purpose in order to trigger the end of active learning when reached (cf. Figure 2-a).



(a) Active learning metric (b) Interactive learning metric

Figure 2: Example of evolution of the error rate obtained by automatic systems during an active or interactive learning process (a and b respectively) as a function of the quantity of interactions with the human domain expert. The area under the curve can be use as a metrics to evaluate the quality of the human assisted learning.

Note that during offline adaptation, the system is not explicitly asked to produce hypotheses on specific data. Therefore we consider in this work that all offline interactions with the Human Domain Expert (HDE) are confined in active learning. The case where the HDE initiates an interaction could be seen as a case of supervised or lightly supervised learning.

2.2. Evaluating online human-assisted learning

During online learning a system has to process a sequence of real world data and return the corresponding hypotheses (outputs). Depending on the scenario, the two types of human/system interactions are possible. On one hand, the LLIS might be allowed to initiate an active learning session before providing its final hypothesis while on the other hand, the human domain expert (HDE) might initiate an interactive learning session by feeding the system with corrections of the current hypothesis in an iterative process. The curves from Figure 2 can be used to evaluate the quality of those two human assisted learning (HAL) processes.

In the case of IAL, it is reasonable to assume that the HDE can provide corrections until the system produces a 100% correct hypothesis. According to this, IAL can be evaluated by the Minimal Supervision Rate (MSR) (Geoffrois, 2016), i.e. the overall cost of error correction needed to get a correct output.

Different systems will allow different interaction modalities. The cost of human interactions can thus be estimated in many ways that depends on the user interface (e.g. time of interaction, amount of simple actions like mouse clicks, etc.) (Broux et al., 2018). An ideal performance measure must provide a fair comparison of multiple systems using different types of user interfaces. For this reason, we propose to take into account the cost of HAL as a scalar metric given in the same unit as the performance score (error rate). This penalisation can be equally applied to AL or IAL and allows comparing systems that have different types of human/system interfaces. Moreover, measuring the cost of interaction in the same unit as the performance of the system allows summing the score and the cost of interactions in order to provide with a unique measure the combine the evaluation of the final performance of the system together with the cost of human assisted learning.

A first idea could be to compute the cost of interaction according to the quantity of information given by the user. However, some evaluation functions (for example BLEU (Papineni et al., 2002) for machine translation) are not linear. In this case it is not possible to estimate the cost of interaction as a function of the quantity of data provided by the user. We propose to compute a penalisation as the quantity of score that corresponds to the data corrected by the human expert during the process. For this purpose, it is proposed to compute two intermediate values: the corrected (S_{cor}) and the impaired (S_{imp}) scores. Computation of those scores is described on Figure 3

Let's assume that the system produces a first hypothesis (see Figure 3-A) and obtains the score S_{base} before applying any online AL/IAL. Starting the human assisted learning, the human domain expert (HDE) corrects (or is asked to correct) part of the current hypothesis. This corrected part of the hypothesis is shown in Figure 3-B and the resulting hypothesis obtains a score S_{cor}. Depending on the task, the part of the hypothesis that is corrected by the HDE might not be entirely wrong. For instance, in a speech transcription task, the HDE might correct a whole sentence while the current transcription of this sentence might include both correct and wrong words. The difference between S_{base} and S_{cor} corresponds to the decrease of score

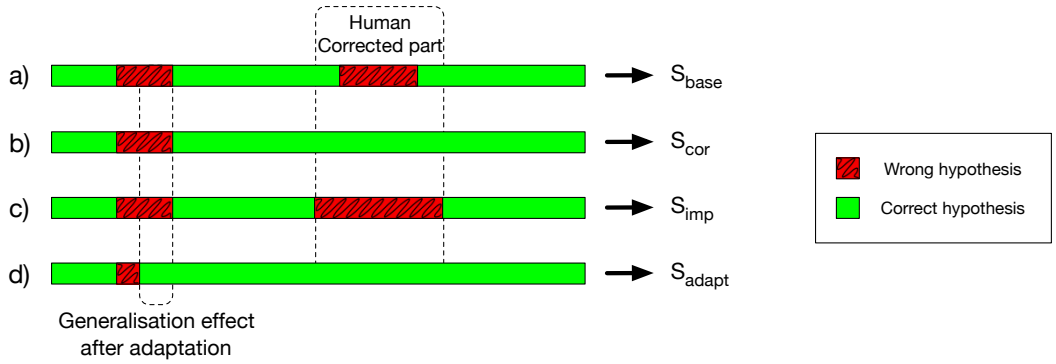


Figure 3: An hypothesis (a) produced by the system contains correct parts (green) ans errors (red) and obtains a score S_{base}. During human assisted learning, the human applies (or is asked to apply) corrections on a part of the hypothesis that might be partially correct. To penalise the system, we introduce a first score, S_{cor}, computed on the corrected hypothesis (b) and a second score, S_{imp}, computed when the human corrected part of the hypothesis is replaced by a wrong hypothesis (c). After the system receives the human correction, it is allowed to generate a final hypothesis (d) by taking into account the correction. Hopefully, the system will generalise the knowledge learnt from the correction to other parts of the data and improve to obtain a score S_{adapt}.

(improvement on error rate) resulting from the corrections provided by the HDE only.

This difference, $S_{base} - S_{cor}$, does not reflect the cost of interaction as it is only related to the part of the corrected data for which the hypothesis was wrong. This is why we compute another score, S_{imp}, that is obtained on another version of the current hypothesis shown on Figure 3-C and where the hypothesis corresponding to the corrected part of the data has been modified with strictly incorrect values. The difference between the impaired score S_{imp} and the score obtained with the user correction (S_{cor}) gives the quantity of score that corresponds to the whole corrected part of the data and that could be considered somehow correlated to the cost of interaction. Eventually, the corrected hypothesis is fed into the system that reprocesses the data with regard to this correction and generates a new hypothesis simulated on Figure 3-D where the system takes into account the correction and might leverage this new knowledge to generalise on other parts of the data to obtain a score S_{adapt}. The penalised score is then computed according to Equation 1

$$S_{pen} = S_{adapt} + (S_{imp} - S_{cor}) \quad (1)$$

Note that a system which does not take the correction into account or ask for already known information may be penalised twice: once in a sub-optimal S_{adapt} and once by the second term of Equation 1.

The effect of penalisation is illustrated in Figure-4. The score, S_{base}, illustrated by column A is obtained before online AL/IAL. Columns B, C and D illustrate the penalised scores, S_{pen}, obtained for different cases of adaptation of the initial model (A). The plain part of columns B, C and D is the score, S_{adapt}, obtained after applying HAL. By nature, it is supposed to be lower than score A. The hatched part of those columns represent the penalisation, computed as in Equation 1. The difference between score S_{base} and S_{pen} is the gain obtained when the system is able to generalise the corrections provided by the HDE to correct other parts of the data.

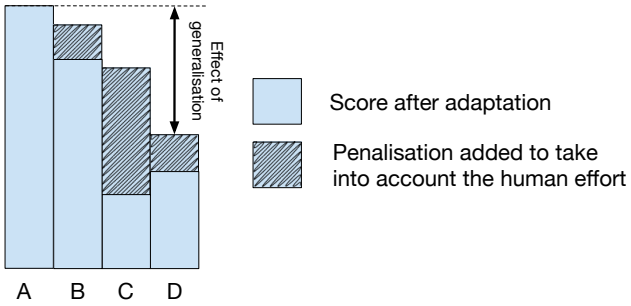


Figure 4: Illustration of the effect of the penalisation method for different cases. Score (A) is the score obtained on the system hypothesis returned before human assisted learning, S_{base}, while the three other columns represent scores S_{pen} obtained on final hypotheses generated after performing human assisted learning with three different methods and using the proposed penalisation policy (note that lower scores are consider better). Plain parts of the columns B, C and D correspond to the score obtained after adaptation, S_{adapt} while the hatched area is the part added to penalise according to the cost of human interaction. Score (B) illustrates the case of a system that takes a limited benefit from a limited amount of user interaction, for (C) a great score reduction is obtained with a strong human interaction and (D) illustrates the case where a limited interaction strongly benefit the system adaptation. Note that the difference between S_{base} and S_{pen} (illustrated for score D on the figure) corresponds to the gain obtained by the system when generalising on the human corrections.

Penalisation helps to figure out the optimal model based on two parameters, final score and cost of interactions. In general the optimal system is a system with the largest generalisation effect, but it can depend on the end-user needs.

Examples of penalisation applied to the BLEU score, an automatic evaluation of machine translation, as well as the Diarization Error Rate (DER) for speaker diarization are presented in Annex A and B, respectively.

3. Evaluation across time

The evaluation of LLIS across time requires taking the life cycle of such a system into account (illustrated by Figure 1-b). At initialisation time, the LLIS will be trained to generate a first version of its model. From then on, the LLIS receives a sequence of time-labelled data, $\{X_{l_i}\}_{i=1...T}$, referred to as *lifelong set* where l_i is the time at which data X_{l_i} has been produced. The LLIS processes this sequence, producing a new version of the model after adapting on each input X_{l_i}. Let's denote as M_0 the initial version of the model and as M_i the version of the model obtained after processing the input X_{l_i}. The performance of the LLIS is evaluated on each batch, X_{l_i}, of lifelong data, producing a sequence of scores $\{S(M_i, l_i)\}_{i=1...T}$.

3.1. Evaluating with respect to a user policy

Because the nature of incoming data evolves across time, LLIS have to continuously update their model to sustain their performance level. However, it might happen that incoming data looks more like data from the past than to contemporary data. In this case, systems could suffer from a catastrophic forgetting (French, 1999) and thus perform poorly on this " data from the past " while they would perform well on contemporary data.

Whether this forgetting effect is critical or not for the end-user depends on the application and this information, we'll call user policy in the remaining, of this paper, is part of what the end-user must define before deploying the system. For some applications, the end-user might want a policy where the LLIS remembers about the past in order to perform equally well on past and contemporary data, but for other applications, the end-user might prefer a policy where the LLIS focuses more on contemporary data or keep a strong memory about the past without adapting much. Those three policies are illustrated on Figure 5.

Now as an external system evaluator, it is important to evaluate the ability of the system to adapt accordingly to the chosen *user policy*. For this purpose, another time-labelled sequence of data, $\{Y_{t_j}\}_{j=1...N}$, is used for test. Time labels for the test data must spread over the same period of time than the *lifelong set*. In other words: $l_1 < t_1 < l_T < t_N$. Each version M_i of the model is then evaluated on all test data Y_{t_j} to get a new sequence of scores $\{s(M_i, t_j)\}_{j=1...N}$. Note that this sequence is computed using the exact same version, M_i, of the model. Computing this sequence of scores on test data that have been produced in the past allows exhibiting whether the system suffers from catastrophic forgetting or not.

An example of such a score sequence is given on Figure 5-1 where a version M_i of the model obtained after adapting until l_i is used to process a sequence of test data. This fake example illustrates the scores of a system that is performing well on contemporary data (data produced at l_i) but which performance degrades on past data due to the forgetting effect. The same figure shows three different *user policy* functions:

(A) expects the system to perform well on contemporary data and progressively forget about the past;

(B) expects the system to perform equally well on past and contemporary data;

(C) expects the system not to learn aggressively on new data in order to sustain performance on past data.

Given the raw score sequence, $\{s(M_i, t_j)\}_{j=1...N}$, obtained by the LLIS with a fixed model M_i, (Figure 5-1) the evaluator weighs those scores according to the chosen user policy. Those weighted scores are shown on Figure 5-2 for all three policies. By summing all weighed scores, one can obtain a single value that represents the performance of the system with fixed model M_i across time for a given *user policy*, $\mathcal{P}$.

$$\mathcal{S}_{\mathcal{P}}^{M_i} = \sum_{t_j \in [l_1; l_i]} s(M_i, t_j) \cdot \mathcal{P}_{t_j} \tag{2}$$

One can observe in Figure 5-3 that the fake system obtains a lower error rate for the (A) policy focusing more on contemporary data, which is consistent with the fact that raw error rates of the system are lower for contemporary data.

3.2. Evaluating human assisted learning across time

Evaluating LLIS including human assisted learning (HAL) across time requires considering the cost of human interaction for all adaptation steps of the LLIS life-cycle (cf. Figure 1-B). When deploying a LLIS system, the cost of human interaction for offline and online learning are definitely different as online learning requires a human expert dedicated to the task in order not to slow down the service while offline learning can be done on the expert spare time.

The paradigm of offline learning defined in this work assumes that the system can learn and ask questions related to any data available to it while being evaluated on a different set of data. In this context, one can not compute the penalisation value described in Section 2.2. as its computation requires to work with labeled data while in the scope of our work, offline learning can also make use of unlabeled data. The cost of HAL must then be computed using ergonomic factors (time, number of actions, etc.) and is thus dependent of the human/system interface and out of the scope of our generalised framework. Consequently, this aspect will not be taken into account during the evaluation of HAL across time.

Online learning, as defined in this work, considers that the system is adapted using the sequence of incoming real world data. This process can be penalised the way introduced in Section 2.2.. The scores obtained on each dataset can thus be weighed using the user policy as described in Section 3.1. and the user policy weighed score across time is now given by Equation 3 where M_i is the model version after offline adaptation on data X_{l_i}.

$$\mathcal{S}_{\mathcal{P}}^{M_i} = \sum_{t_j \in [l_1; l_i]} S_{pen}(M_{ij}, t_j) \cdot \mathcal{P}_{t_j} \tag{3}$$

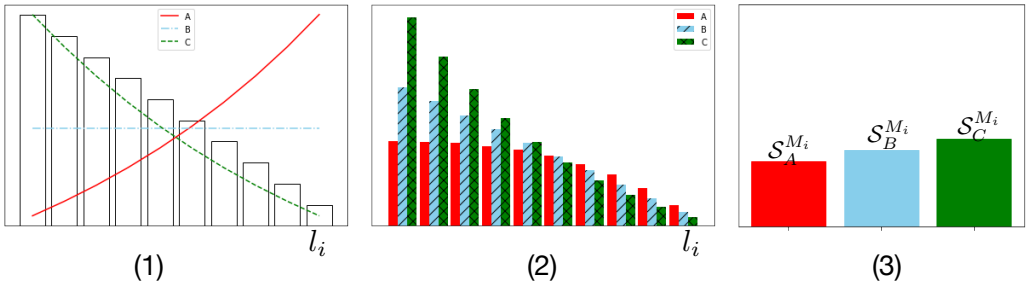


Figure 5: LLIS scores across time along with 3 different user policies (1). User policy weighted scores across time (2). Average score of the system when weighted by the 3 different user policies (3).

4. Discussion

This article proposes three contributions for the evaluation of lifelong learning intelligent systems. First we propose a new way to take into account the cost of human interactions in a human assisted learning process (active or interactive learning). This measure of interaction is given in the same unit as the system score and allows a wider use of this penalisation for system comparison. Second we introduce the concept of user policy in order to evaluate the ability of a lifelong learning system to forget past data according to a policy defined by the end-user. Third we combine those approaches in order to provide a complete evaluation framework for lifelong learning intelligent systems. Our proposal considers the general task of evaluating lifelong learning automatic systems, using an error-rate type of metric (lower is better) but it can similarly be applied to any scalar metric that is bounded from one side.

The proposed penalisation of system performance has been designed to take into account the cost of human interactions regardless of the interaction modality and in the same unit of measurement as the performance itself in order to be directly combined and offer a unique performance indicator. This penalisation measure can be used to evaluate the ability of the systems to generalise the information provided by the human expert to other parts of the data. However, the generality of this measure does not allow to express the cost of human interaction in terms of concrete units such as time spent by the operator, onerousness, strenuousness, etc.. For some specific cases, the computation of impaired scores rises questions related to the definition of *wrong hypothesis*. For instance, for the case of a binary classification task, the impaired hypothesis consists of allocating the wrong label to an element. The case of machine translation evaluated by the Translation Edit Rate (TER, (Snover et al., 2006)) is problematic. Because it is always possible to worsen the score by inserting words, then the worst translation cannot be identified. A simple way to deal with such a case is to bound the maximum error to the sequence length, which sounds a reasonable compromise. In appendix, we illustrate the penalised evaluation for the cases of another machine translation metric (BLEU) and a speaker diarization metric (Diarization Error Rate).

The proposed computation of penalisation assumes that the answer provided by the human expert (or the correction in case of interactive learning) can be directly applied to the hypothesis in order to compute a new score, S_{cor}. This assumption might not be always realistic depending on the type of question the system ask. For instance, a human expert providing the only information that a class label is wrong for the case of multi-class classification problem does not allow the system to get the correct answer. In this case the system just knows its answer is wrong and this information can not be used to compute a corrected score. Our penalisation computation is thus only applicable for the case where the human provides an information that can be used to correct part of the hypothesis.

In order to assert a fair and reproducible evaluation of systems including human assisted adaptation, one must implement a human-expert simulation module that could simulate the interaction process between the system and the human domain expert. Different simulations might affect the evaluation and all systems to be compared have to make use of the same human-expert simulation. Ideally, and depending on the task, several human-expert simulation could be used but the development of those simulations for evaluation purpose is a completely different research topic that is not addressed in this work.

As a second contribution, we introduced a user-policy that enables the human expert to define the model adaptation policy depending on his/her empirical knowledge of the real world data. This policy enables the user to set the system adaptation for tasks where future data will never look like the past one or to plan for cyclic evolution of the data for instance. On the positive side, the introduction of this policy allows the system to benefit from the domain expert's empirical knowledge. On the other side, defining such a policy could be a difficult task that assumes a good knowledge of the field and a good understanding of the performance measure. This aspect of the evaluation process would require a careful design of the system parameter interface.

In this work, our goal has been to define the most general framework possible that would fit a variety of tasks, systems and performance measures. As it is the case for many well known evaluation metrics or protocols the proposed ones suffer from limitations and have to be combined or completed in order to render satisfactory information to the evaluator. Moreover, we didn't address all possible system architectures in order not to complicate our proposal too much, considering that more complex architectures would require an extended work that could follow this one.

The proposed metric are being derived for the tasks

of speaker diarization and machine translation and will be used in two open evaluations, namely the ALLIES/ALbayzin 2020 evaluation and WMT2020. At the end of the evaluations, implementations of the metrics, protocols as well as new data collected to evaluate lifelong learning systems for those two modalities will be publicly released.

Acknowledgments
This work has been funded by the CHIST-ERA project ALLIES (ARN-17-CHR2-0004-01) `https://projets-lium.univ-lemans.fr/allies/` and the CapDiff project (AAP PME 2017) `https://lium.univ-lemans.fr/capdiff/`.

5. Bibliographical References

Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M. (2018). The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Broux, P.-A., Doukhan, D., Petitrenaud, S., Meignier, S., and Carrive, J. (2018). Computer-assisted speaker diarization: How to evaluate human corrections. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Celemin, C. and Ruiz-del Solar, J. (2019). An interactive framework for learning continuous actions policies based on corrective feedback. *Journal of Intelligent & Robotic Systems*, 95(1):77–97.

Chen, Z. and Liu, B. (2016). Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145.

Drugman, T., Pylkkonen, J., and Kneser, R. (2019). Active and semi-supervised learning in asr: Benefits on the acoustic and language models. *arXiv preprint arXiv:1903.02852*.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3:128–135.

Geoffrois, E. (2016). Evaluating interactive system adaptation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 256–260.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*. ACL.

Pérez-Dattari, R., Celemin, C., Ruiz-del Solar, J., and Kober, J. (2018). Interactive learning with corrective feedback for policies based on deep neural networks. *arXiv preprint arXiv:1810.00466*.

Prince, S. J. (2012). *Computer vision: models, learning, and inference*. Cambridge University Press.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. The MIT Press.

Siddhant, A. and Lipton, Z. C. (2018). Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. *arXiv preprint arXiv:1808.05697*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation.

A Penalizing BLEU

Let's assume we have a sentence S of length l_S along with a human generated reference sentence of length r_S. BLEU score calculation for a corpus C is given by the following:

$$\log \text{BLEU} = \min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n$$

where N is the maximum n-gram length (generally set to 4), p_n the clipped precision score for n-grams of length n and w_n the corresponding weight (generally $1/N$). r_C and c_C are the total lengths of the references and hypotheses, respectively. The clipped precision are computed as follows:

$$p_n = \frac{\sum_{S \in C} \sum_{n\text{-gram} \in S} Count\_clip(n\text{-gram})}{\sum_{S' \in C} \sum_{n\text{-gram'} \in S'} Count(n\text{-gram'})}$$

This metric is computed on a whole corpus (and not a single sentence).
BLEU is a precision measure, this means that the score mostly depends on the system output (modified n-gram precision). In order to calculate a lower bound of the BLEU score regarding a particular sentence, we propose to consider that the translation process provided an entirely wrong sentence (not a single correct 1-gram). In consequence:

- $Count\_clip(n\text{-gram})$ are 0 for this particular sentence
- total n-gram counts in the hypothesis remain unchanged

As an example, let's assume that a toy corpus is composed of 10 sentences.

NAME	**Correct n-grams** 1g / 2g / 3g / 4g	**BP**	**hyp. len.**	**ref. len.**	**%BLEU**
BASE	140 / 77 / 41 / 25	0.975	233	239	25.81
IMPAIRED	133 / 73 / 40 / 25	0.975	233	239	24.99
CORRECTED	144 / 83 / 49 / 33	0.957	229	239	29.70
ADAPTED	151 / 89 / 54 / 37	0.953	228	239	32.25
Penalised scores					
NO LEARNING					21.10
NO GENERALISATION					24.99
GENERALISATION					27.54

Table 1: Statistics for the %BLEU evaluations under several conditions. BP is the brevity penalty, hyp. len. and ref. len. correspond respectively to the hyopothesis and reference length.

The statistics for the BLEU evaluations under the different conditions (BASE, IMPAIRED, CORRECTED and ADAPT) are presented in Table 1. The sentence considered for correction contains 15 tokens. The BLEU score associate with this sentence is given by the difference between CORRECTED and IMPAIRED scores, namely 4.71% BLEU. The ADAPT output has been simulated by correcting several n-grams in the CORRECTED output.

B Penalizing DER

Traditionally, the performance of diarization systems are given in terms of Diarization Error Rate (DER). The DER is computed as the fraction of speaker time that is not correctly attributed to its speaker. This score will be computed over the document collection to be processed; including regions where more than one speaker is present (overlap regions). This score will be defined as the ratio of the overall diarization error time to the sum of the durations of the segments that are assigned to a speaker label. Given the data set to evaluate, each document is divided into contiguous segments at all speaker change points found in both the reference and the hypothesis, and the diarization error time for each segment n is defined as:

$$E(n) = T(n)[\max(N_{ref}(n), N_{sys}(n)) - N_{Correct}(n)] \tag{4}$$

where $T(n)$ is the duration of segment n, $N_{ref}(n)$ is the number of speakers that are present in segment n of the reference file, $N_{sys}(n)$ is the number of system speakers that are present in segment n and $N_{Correct}(n)$ is the number of reference speakers in segment n correctly assigned by the diarization system.

$$DER = \frac{\sum_{n \in \Omega} E(n)}{\sum_{n \in \Omega} (T(n) N_{ref}(n))} \tag{5}$$

In order to calculate a lower bound of the DER score regarding a particular audio recording, we propose to consider that the diarization process provided an entirely wrong hypothesis there is no intersection between hypothesis and human generated reference. For simulation of totally wrong hypothesis it is proposed to keep the non-speech part of the hypothesis as non-speech and to allocate the speech part to a non-existent speaker. This is equivalent to setting $E(n)$ to zero for the segment corrected by the human in the loop.