



Lifelong learning and task-oriented dialogue system: what does it mean?

Mathilde Veron, Sahar Ghannay, Anne-Laure Ligozat, Sophie Rosset

► To cite this version:

Mathilde Veron, Sahar Ghannay, Anne-Laure Ligozat, Sophie Rosset. Lifelong learning and task-oriented dialogue system: what does it mean?. International Workshop on Spoken Dialogue Systems Technology, Apr 2019, Siracusa, Italy. hal-02301089

HAL Id: hal-02301089

<https://hal.science/hal-02301089>

Submitted on 30 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.

Lifelong learning and task-oriented dialogue system: what does it mean?

Mathilde Veron, Sahar Ghannay, Anne-Laure Ligozat, Sophie Rosset

Abstract The main objective of this paper is to propose a functional definition of lifelong learning systems adapted to the framework of task-oriented dialogue systems. We mainly identified two aspects where a lifelong learning technology could be applied in such systems: to improve the natural language understanding module and to enrich the database used by the system. Given our definition, we present an example of how it could be implemented in an existing task-oriented dialogue system that is developed in the LIHLITH project.

1 Introduction

A dialogue system allows a user to interact using natural language. Two families of dialogue systems exist: conversational systems and task-oriented systems. Conversational systems have to generate the most appropriate reaction given a user's utterance and a context, without any restriction about the domain. A task-oriented system aims to help the user perform a task or access information (in this family, both open and limited domains are possible). A dialogue system generally consists

Mathilde Veron
LIMSI, CNRS, Université Paris-Saclay, Campus universitaire, bât. 507, F-91405 Orsay Cedex, France, e-mail: firstname.lastname@limsi.fr

Sahar Ghannay
LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Campus universitaire, bât. 507, F-91405 Orsay Cedex, France e-mail: firstname.lastname@limsi.fr

Anne-Laure Ligozat
LIMSI, CNRS, ENSIIE, Université Paris-Saclay, Campus universitaire, bât. 507, F-91405 Orsay Cedex, France e-mail: firstname.lastname@limsi.fr

Sophie Rosset
LIMSI, CNRS, Université Paris-Saclay, Campus universitaire, bât. 507, F-91405 Orsay Cedex, France e-mail: firstname.lastname@limsi.fr

of three modules: natural language understanding (NLU), dialogue management and natural language generation (NLG). In this paper, we focus on task-oriented system, and more precisely, on classical task-oriented systems and not on end-to-end ones.

There has recently been a growing interest for building lifelong learning (LL) dialogue systems. LL is a subject which has led to numerous works in different fields, such as machine learning, deep learning¹, artificial intelligence or robotics [1, 2, 3]. It is commonly considered as the “ability to continually learn over time by accommodating new knowledge while retaining previously learned experiences” [3]. Performing LL in each field differs, depending on the nature of the new knowledge to learn, and on the way the learning is conducted. In the field of dialogue systems, the application of LL should result in dialogue systems which are able to continuously learn over time, by interacting with the user as two humans would do it in real life. For example, a human can learn new knowledge at any time by asking questions when facing an unknown situation, or simply by inferring new knowledge from the current conversation. A human is also able to use this learned knowledge directly or in future conversations. The new piece of knowledge can be for instance vocabulary, information or even an appropriate behavior to have in a specific situation. A LL dialogue system should thus be able, as a human, to improve its communication abilities and, in the case of a task-oriented system, to learn new knowledge specific to its application domain. This implies that the system should be able to detect new knowledge in the user’s utterances and to understand when it should ask questions in order to learn new knowledge.

We therefore want to delimit how far a dialogue system can mimic the human continuous learning process, by determining exactly what a dialogue system can learn, how far it can *continuously learn over time*, and the methods which have to be developed to achieve this goal. In more simple terms, we want to determine what could be a LL dialogue system. It will also be interesting to wonder if we can adapt LL techniques from other fields to dialogue systems.

We present in Section 2 previous works which have been done about LL, reinforcement learning and learning dialogue systems. In Section 3 we attempt to define what a LL dialogue system is. Finally, we apply this definition to an existing cooking chatbot and try to propose mechanisms in order to perform LL, in Section 4.

2 Related Work

Lifelong learning (LL) has been of growing interest for several years resulting in numerous works and writings [1, 2, 3]. In their book [2], Chen and Liu define three features characterizing a LL system: continuous learning, knowledge accumulation and maintenance in the knowledge base (KB), and the ability to use past knowledge to help future learning. Moreover, they consider that a LL system should be able to detect new situations in the usage process and to adapt its behaviour to these

¹ see for example the workshop Continual Learning at NIPS in 2018, <https://sites.google.com/view/continual2018/home?authuser=0>

new situations, by learning new tasks and performing self-motivated learning. The learned knowledge, which is accumulated, should also help the system to learn new tasks with little data or effort. In particular, the new task can belong to an other domain (subject area). However, this *new task* concept is unclear and seems to cover a broad variety of situations, going from *new instance* (i.e. a new instance of a concept in a model) to completely new domain [4].

Simultaneously, progress has been made in the domain of Reinforcement Learning (RL). A system using RL methods learns from past experiences to estimate the most accurate decision process by maximizing a predefined reward function. In particular, RL in dialogue systems has been widely studied to enable the system to learn a dialogue policy [5], considering for example online scenarios [6], and inverse RL framework [7]. More recently, Li and his colleagues [8] used the RL framework in their chatbot, which aims to answer questions of the movie domain. Thanks to RL, their system is able to learn when and how to ask questions to a user, in order to improve itself during this specific interaction.

Li and his colleagues [8] identify three situations where their dialogue system can ask questions:

- *Understanding problem*: the system asks for a paraphrase or verification. They focus here only on typos/spelling mistakes.
- *Knowledge operation problem*: the system needs help to perform reasoning steps over the KB.
- *Lack of knowledge*: the system asks for the answer.

RL methods can be applied to a dialogue system in order to perform LL but other methods are possible. For example, Mazumder and his colleagues developed a continuous knowledge learning engine for chatbots [9]. To achieve this, they focus on solving the open-world knowledge base completion problem by interactive learning and inference. They developed techniques to enable the system to continuously and interactively learn new knowledge in particular during real conversations. They use a knowledge store to retain the learned knowledge. They also developed a guessing mechanism, as they consider that the user may not know the information that the system is asking for.

In their second edition of their book [10], Chen and Liu dedicated the 8th chapter to continuous knowledge learning in chatbots. In this chapter, they essentially discuss the article mentioned above [9] and claim:

Lifelong learning is reflected by the fact that the newly acquired facts are retained in the KB and used in inference for future queries, and that the accumulated knowledge in addition to the updated KB including past inference performances are leveraged to guide future interactions and learning.

However, Mazumder and his colleagues focus only on the problem of KB completion and provide a specific example of what can be a LL dialogue system. In Section 3, we propose a global definition of a LL dialogue system and we describe the different types of knowledge that can be learned.

3 What is a LL dialogue system: attempt of a definition

Before defining what a LL dialogue system is, we have to mention that LL should not be considered as a method like Reinforcement Learning, but more as a goal. In fact, the nature of the knowledge that can be learned, the methods involved and the objectives, strongly depend on the field and the application domain. It would though be interesting to define LL concepts and methods that can be shared between the different fields.

Thanks to the definitions given in [2, 10] and the previously described papers [8, 9], we can say that a lifelong learning dialogue system is a conversational system that continuously learns over time through interactions with a real user and possibly by asking questions. The learned knowledge should be retained, in order to allow the system to directly have access to this new knowledge, and to directly adapt its behaviour. The system should also keep the modifications for future dialogues. Moreover, the learned knowledge should help the system to infer additional knowledge and to learn in the future in a more efficient and in a faster manner. During interaction with the user, the system should thus be able to detect new situations and to learn how to adapt.

We now focus on task-oriented dialogue systems. This kind of system aims to help the user to perform a task or to access information. To achieve this, the system first tries to understand what the user is asking for, by calling the NLU sub-system which performs slot-filling and intent detection. Then, the dialogue manager tries to link the slots and the intent detected to the data that it has access to. Finally, the system returns the result by generating natural language.

Considering the previous definition of a LL dialogue system and the distinctions made in [8] we can see that a LL task-oriented dialogue system can learn at different levels:

- *Improve interactions*:
 - by being better at understanding what the user says (NLU)
 - better perform slot-filling and intent detection
 - be able to detect new intents and/or new slots
 - be able to detect out-of-domain user's utterance
 - by being better at generating natural language (NLG)
- *Enrich data*: the system can learn new data by enriching its knowledge base
- *Link new intent to data*: the system can learn to link already known users intent or new ones to the data it has access to. It can thus learn to perform new tasks.

However, we should pay particular attention to the fact that the user may transfer incorrect information, that can be learned by the system. For example, in the case of data enrichment, the system can ask the user about a specific piece of information that is missing in its knowledge base. The problem is that the user can not know the answer, can be wrong, or the data can be subjective. Moreover, he can also make typos or spelling mistakes, and he can have his own way to communicate given his age and where he lives for instance.

This consideration highlights, in particular, the need of defining when, or under which conditions, a learned knowledge should be considered as reliable. After this, the learned knowledge can be used for retraining a model, or can definitely be retained in the knowledge base of the system.

Moreover, to perform LL we could make use of *Online Reinforcement Learning* (RL) techniques, to allow the system to decide when it should ask questions and what questions it should ask, as done in [8]. All the methods which can be used rely on a certain amount of available data and even on data simulation (e.g. user simulation learned on available data in the case of RL).

4 Application to a cooking chatbot

As part of the LIHLITH project², we developed a chatbot in the cooking domain, that allows, among others thing, a user to find a recipe matching his criteria.

In the following sections, we first give an overview of the system and then describe how LL could be implemented in it.

4.1 General description of the LIHLITH cooking chatbot

The LIHLITH cooking chatbot is divided into three sub-systems:

- *NLU module*: takes as input the utterance of the user and returns the slots and the intent associated to this utterance. Considering the following user utterance: "Please find me a recipe of pancakes without eggs", the NLU should detect the slots "recipe: pancakes" and "neg-ingredient: eggs" plus the intent "RECING", that means that the user is looking for a recipe by giving the name of the recipe and the ingredients. The NLU module is based on deep neural network performing both slot-filling and intent detection [11], also known as *joint NLU*. We use an off-the-shelf system³. It has been trained with data which have been automatically generated and annotated thanks to patterns and lists of terms according to the method described in [12].
- *Dialogue manager*: decides what action to perform depending on the results of the NLU module and the current context, using a semantic textual similarity module or a database module.
- *NLG module*: generates the answer with the help of patterns

² Learning to Interact with Humans by Lifelong Interaction with Humans

³ All needed information can be found on <https://github.com/SNUDerek/multiLSTM>

Note that both the dialogue manager and the NLG module are based on rule-based approaches.

The LIHLITH cooking chatbot⁴ handles two different types of scenarios:

1. It searches in the recipes database if the user wants to find a recipe matching his criteria. The user may say for example "Please find me a recipe of pancakes without eggs".
2. It searches in unstructured data by estimating semantic textual similarity, if the user asks a question relative to the cooking domain. The user may say for example "Why is -18°C the ideal freezer temperature?".

For these two types of scenarios we use different types of data: the recipes database and the unstructured data. The recipes database has been built upon Wikipedia:Cookbook, and includes *1,064 recipes*. It contains for example information about the name, the details, the ingredients, the procedure and the categories. The unstructured data is composed of *784 non-recipe documents* that can be found on Wikipedia:Cookbook.

4.2 Lifelong Learning for cooking chatbot

Based on the definitions of Section 3, we decide to focus on two ways of performing LL on our chatbot: the improvement of understanding and the data enrichment. We discuss in the following subsections how it could be done, with the help of dialogue examples.

4.2.1 Improve Understanding

The first way aims at letting the system improve its understanding capabilities. In this case, it should be able to better detect slots and intents. It also should be able to learn new slots and new intents and to link each new intent to the right data.

There are several situations, where the system can detect that it actually misunderstood what the user said:

- If there is a contradiction between the slots and the intent which have been detected.
- If the score of the detected intent or slot is low.
- If the user reacts negatively, by saying for example "You misunderstood me" or "You're wrong".

When facing this kind of situation, the goal is not only to detect the problem and deal with it in the current dialogue, but is also to infer/obtain new information allowing the system to adapt itself.

⁴ see <https://lihlith.limsi.fr/dialog.php>. To try it, use the following login and password: "lihlith" and "recipe?"

We consider the example presented in Figure 1, where the system should detect a contradiction between a slot and an intent⁵.

	Utterance	System action														
User	i'd like to prepare a mousse	The system hasn't detected that "mousse" corresponds to the slot "recipe" but has detected the intent "REC"														
System	Are you asking for a specific recipe?	Ask confirmation about the intent detected														
User	Yes															
System	Can you tell me the name of the recipe alone?	Ask for the missing slot														
User	Mousse	<div>Save the annotated utterance:</div> <table><tr><td>i'd</td><td>like</td><td>to</td><td>prepare</td><td>a</td><td>mousse</td><td>INTENT</td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>b-recipe</td><td>REC</td></tr></table>	i'd	like	to	prepare	a	mousse	INTENT	O	O	O	O	O	b-recipe	REC
i'd	like	to	prepare	a	mousse	INTENT										
O	O	O	O	O	b-recipe	REC										
System	I found 3 recipes, including: atkins-friendly chocolate mousse, chocolate mousse cake (vegan) and chocolate mousse . Which one do you want?	Answer the initial query														

Fig. 1 Example of contradiction between detected slot and intent.

When facing this kind of problem, the system could first ask the user to clarify the slots and the intent it has detected. This clarification process would thus lead to the creation of an annotated user's utterance.

In order to allow the system to learn from this clarification, and to be able to reuse it directly or in future conversations, this annotated data can first be saved in a specific file. Then, when detecting a misunderstanding, the system would, before even asking the user about clarification, look into this specific file if the user's utterance that the system doesn't understand, corresponds or has high similarities with an annotated utterance in the file.

Moreover, we could consider retraining our NLU model with the accumulated annotated user's utterances consisting in new training examples. This could be done for example after a specified number of annotated user's utterances have been added to this file and if the annotated data seems to be useful and reliable according to user's feedbacks. However, we should be careful about respecting the proportions of the different slots and intents in the training data, when retraining the model with the new examples, in order to minimize the problem of catastrophic forgetting and performance decrease [3].

We could also infer a pattern from the annotated user's utterance and generate automatically annotated data from this pattern to retrain our understanding model. In the example presented in Figure 1, we could infer the pattern "i'd like to prepare a \$recipe".

⁵ The intent REC means that the user searches for a recipe by giving the name of the recipe

One can also face a situation where the system detects a new intent as it is shown in Figure 2.

	Utterance	System action
User	I'd like to cook roast chicken	
System	OK, for roast chicken . Do you want the ingredients?	Begin sub-dialogue
User	What temperature for my oven?	
System	I don't know. Can you give me an example of an appropriate answer?	Ask for clue
User	210°C or 370 degree F	
System	I found "Preheat your oven to 200°C (390°F)" Are you satisfied with this answer?	Look in the recipe document for the sentence which is the most similar to the example of the user and ask for feedback
User	Yes	

Fig. 2 Example of new intent (sub-dialogue).

In such a situation, the question is not understood at all by the system and it then decides to ask the user to give an example of what could be a plausible answer to this question. Using this plausible answer, the system can apply other techniques to find the relevant piece of information, by searching in its database or in external resources (here the unstructured data it has access to). If the user confirms that the answer of the system is appropriate, the system could enter to another process to complete automatically its database. As proposed in [13], the system could thus check in its database if this new intent could be applied to other recipes. In the example of Figure 2, this process could result in the creation of the feature "oven_temperature" and a value could be associated with the new feature for all the recipes needing an oven.

It is important that the system chooses the right moment to ask its question to the user. As it is done in [8], methods based on (inverse) reinforcement learning framework might be very interesting.

4.2.2 Data enrichment

This section presents the second way to perform LL in order to improve our system, which is about new factual knowledge acquisition and more specifically enriching the recipe database.

At some point in the dialogue, the user and the system are speaking about one specific recipe. We consider, in this case, that if a piece of information is missing in the database, like the time needed, the difficulty or the serving, the system should ask

the user about it, in order to enrich its database. This situation is presented in Figure 3.

	Utterance	System action
System	Hello, are you hungry?	
User	i'd like to cook basic scones	
System	You're telling me about basic scones . Is that OK?	Ask for confirmation
User	Ok	
System	OK, for basic scones . Do you want the ingredients?	
User	Yes	
System	The ingredients are: 1 cup flour (100g), 1 tsp baking powder (5g), 1 pinch salt, 2 Tbs butter (50g), 1 Tbs sugar (50g), 1 cup milk (125ml) + extra milk for glazing 2. Have you ever cooked basic scones?	The system has detected that at least a piece of information about the recipe is missing in the DB and decides to interact by asking how familiar the user is with the recipe.
User	Yes	
System	Do you remember how much time did it take?	Ask the user for the missing data
User	I needed approximately 30 min	Add to the database "cooking_time = 30 min"
System	OK, thank you very much!	

Fig. 3 Example of data enrichment.

RL could be used to train the system to decide when it should ask for a new piece of information as it is done in [8]. RL will also allow the system to learn which piece of information is the most interesting to ask. The system can base its decisions on various parameters such as the number of time such a missing piece of information is asked by different users, if the current user seems to be willing to answer system's questions, if he seems to be knowledgeable in cooking, etc. And obviously RL will help to build a model on such an action to collect new data in a proactive way.

5 Conclusion and Discussion

In this paper we try to define what a LL task-oriented dialogue system could be and to apply this definition to our existing chatbot devoted to the cooking domain. We consider that two ways of improvement are possible for our chatbot: the system can learn to better understand and can enrich its database. We consider different situations in which the system might want to interact with the user in order to obtain his help. In such cases, a RL framework can be used to learn in which condition a question could be asked.

When applying our hypotheses to our chatbot, we can observe that a lifelong learning task-oriented dialogue system involves:

- having for the system, the ability to understand from user’s utterance when it can learn new knowledge and the ability to handle it in the current context;
- some criteria or score functions which evolve over time to know when to ask questions and what questions to ask and to whom;
- a way to store the learned knowledge;
- a criterion to know when to retrain a model and an other one to know when to permanently store the learned knowledge in case of knowledge base enrichment.

However, we presented only simple LL aspects and put aside the complex ones such as LL for improving dialogue quality. Such an aspect would be interesting in any case and more specifically when working with conversational systems.

Acknowledgements This work has been partially funded by the LIHLITH project (ANR-17-CHR2-0001-03), and supported by ERA-Net CHIST-ERA, and the Agence Nationale pour la Recherche (ANR, France).

References

1. Sebastian Thrun and Tom M. Mitchell. Lifelong robot learning. *Robotics and Autonomous Systems*, 15:25–46, 1995.
2. Zhiyuan Chen and Bing Liu. *Lifelong Machine Learning*. Morgan & Claypool Publishers, 2016.
3. German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *CoRR*, abs/1802.07569, 2018.
4. Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *arXiv preprint arXiv:1806.08568*, 2018.
5. Steve Young. Using pomdps for dialog management. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 8–13. IEEE, 2006.
6. Alexandros Papangelis, Vangelis Karkaletsis, and Fillia Makedon. Evaluation of online dialogue policy learning techniques. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), 2012.
7. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
8. Jiwei Li, Alexander H Miller, Sumit Chopra, Marc’Aurelio Ranzato, and Jason Weston. Learning through dialogue interactions by asking questions. In *ICLR*, 2017.
9. Sahisnu Mazumder, Nianzu Ma, and Bing Liu. Towards a Continuous Knowledge Learning Engine for Chatbots. *CoRR*, abs/1802.06024, 2018.
10. 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.
11. Gokhan Tur. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons, January 2011.

12. Antoine Neuraz, Leonardo Campillos Llanos, Anita Burgun, and Sophie Rosset. Natural language understanding for task oriented dialog in the biomedical domain in a low ressources context, NIPS Workshop. In *Machine Learning for Health (ML4H): Moving beyond supervised learning in healthcare*, Montréal, Québec, Canada, 2018.
13. Anselmo Peas, Mathilde Veron, Camille Pradel, Arantxa Otegi, Guillermo Echegoyen, and Alvaro Rodrigo. Continuous Learning for Question Answering. In *Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, 2019.