Lifelong learning et systèmes de dialogue : définition et perspectives

Mathilde Veron[1]
(1) LIMSI, CNRS, Université Paris-Saclay, Campus universitaire, bât. 507, F-91405 Orsay Cedex, France
prénom.nom@limsi.fr

RÉSUMÉ

Le but de cet article est de définir comment le *Lifelong Learning* (LL) pourrait être appliqué aux systèmes de dialogue orientés tâche. Un système de dialogue devrait être en mesure d'apprendre de nouvelles connaissances, après avoir été déployé, et ceci de manière continue grâce à ses interactions avec l'utilisateur. Nous identifions ainsi deux aspects s'appliquant à un tel système : l'amélioration de ses capacités conversationnelles, et l'enrichissement de sa base de connaissances. Nous appliquons ces idées à un chatbot développé dans le cadre du projet LIHLITH. Nous montrons ainsi qu'un tel système doit être capable (1) de détecter la présence d'une situation inconnue (2) de décider quand et comment interagir avec l'utilisateur afin d'extraire de nouvelles connaissances et (3) de s'adapter à ces nouvelles connaissances, tout en considérant la fiabilité de celles-ci.

ABSTRACT

Lifelong learning and dialogue system : definition and discussion

This paper aims to define what could be a Lifelong Learning (LL) task-oriented dialogue system. A dialogue system should be able to learn new knowledge, after deployment and continuously, thanks to its interactions with the user. We identify two ways for a task-oriented dialogue system to perform LL: the improvement of its conversational capabilities, and the enrichment of its knowledge base. We apply these ideas to a chatbot developed as part of the LIHLITH project. We suggest that such a system should be able (1) to detect a new situation, (2) to decide when and how to interact with the user in oder to extract new knowledge and (3) to adapt itself to these new knowledge by considering their reliability.

MOTS-CLÉS : système de dialogue, apprentissage continu, système de dialogue orienté tâche.

KEYWORDS: dialogue system, lifelong learning, task-oriented dialogue system.

1 Introduction

Un système de dialogue permet à un utilisateur d'interagir en langage naturel. Deux familles de systèmes de dialogue existent : les systèmes conversationnels et les systèmes orientés tâche. Un système conversationnel a pour but de générer la réaction la plus appropriée étant donné un énoncé utilisateur et le contexte courant, sans restriction concernant le domaine. Quant à eux, les systèmes orientés tâche ont pour but d'aider l'utilisateur à réaliser des tâches définies, ou bien de l'aider à accéder à des informations. Un système de dialogue consiste généralement en trois modules : la compréhension du langage naturel (en anglais, *Natural Language Understanding* ou *NLU*), la gestion du dialogue, et la génération de langage naturel (en anglais, *Natural Language Generation* ou *NLG*).

Dans cet article, nous nous focaliserons sur les systèmes orientés tâche, et plus précisément sur les systèmes modulaires et non sur les *end-to-end*.

Récemment certains chercheurs se sont intéressés à la création de systèmes de dialogue réalisant de l'apprentissage continu (*Lifelong Learning* ou *LL*). Le LL est un sujet qui a mené à de nombreux travaux dans des domaines variés, tels que l'apprentissage automatique (*machine learning*), l'apprentissage profond (*deep learning*) [1], l'intelligence artificielle et la robotique (Thrun & Mitchell, 1995; Chen & Liu, 2016; Parisi *et al.*, 2018). Le LL est communément considéré comme la capacité à apprendre de manière continue de nouvelles connaissances, tout en conservant les données acquises lors d'expériences passées (Parisi *et al.*, 2018). Réaliser du LL dans chacun des domaines diffère selon la nature de la connaissance à apprendre, et de la façon dont cet apprentissage peut être réalisé. Dans le cadre des systèmes de dialogue, l'application du LL pourrait mener à des systèmes capable d'apprendre de façon continue grâce à leurs interactions avec l'utilisateur, à l'image des humains. Par exemple, un humain peut apprendre de nouvelles connaissances en posant des questions à ses interlocuteurs lorsqu'il fait face à une situation inconnue, ou simplement en déduisant de nouvelles connaissances à partir de l'ensemble des échanges qu'il a pu avoir au cours de sa vie. Un humain est aussi capable d'utiliser directement, ou dans de futures conversations, les connaissances qu'il a acquises. Ces connaissances peuvent correspondre, par exemple, à du vocabulaire, à des informations, ou même à un comportement approprié à avoir dans des conditions spécifiques. Un système de dialogue réalisant du LL devrait donc, à l'image d'un humain, être en mesure d'améliorer ses capacités conversationnelles et, dans le cas de systèmes orientés tâches, d'apprendre de nouvelles connaissances spécifiques à son domaine d'application.

Nous cherchons donc à définir jusqu'où un système de dialogue est en mesure d'imiter l'homme et sa capacité à apprendre de manière continue. Pour cela, nous cherchons à déterminer de façon précise ce qu'un système de dialogue peut apprendre, les limites de son apprentissage, ainsi que les méthodes qui devront être développées pour atteindre cet objectif.

Dans cet article nous proposons une définition du LL adapté aux systèmes de dialogue orientés tâche et montrons comment celle-ci peut s'appliquer dans un cas pratique.

Dans la suite de ce document, nous commençons par présenter en section 2 des travaux qui ont été menés sur le LL, sur l'apprentissage par renforcement et sur les systèmes de dialogue apprenant. Dans la section 3, nous cherchons à définir ce que peut être un système de dialogue réalisant du LL ainsi que les méthodes et techniques qui pourront ou devront être utilisées. Enfin, en section 4, nous appliquons cette définition ainsi que les méthodes présentées précédemment à un chatbot dédié à la cuisine. Il ne s'agit ici que de propositions et de discussions et aucun travail expérimental n'est présenté.

2 État de l'art

Le Lifelong learning (LL) est d'intérêt croissant depuis plusieurs années, ce qui a mené à de nombreux travaux et écrits (Thrun & Mitchell, 1995; Chen & Liu, 2016; Parisi *et al.*, 2018).

Dans leur livre (Chen & Liu, 2016), Chen et Liu définissent trois caractéristiques représentatives d'un système réalisant du LL : (1) l'apprentissage continu, (2) l'accumulation et la maintenance des connaissances dans une base de connaissances (en anglais, *Knowledge Base* ou *KB*) et (3) la capacité

1. par exemple, Continual Learning Workshop, NeurIPS 2018

à utiliser les connaissances acquises pour de futurs apprentissages. De plus, ils considèrent qu'un système réalisant du LL doit être capable de détecter lorsqu'une nouvelle situation se présente dans l'usage, et d'adapter son comportement à ces nouvelles situations, en apprenant à réaliser de nouvelles tâches de manière pro-active. Les connaissances acquises et accumulées doivent de plus permettre au système d'apprendre de nouvelles tâches sans nécessiter de gros volumes de données supplémentaires et sans devoir déployer de moyens trop importants. Cependant, le concept de *nouvelle tâche* reste flou et semble couvrir un large spectre de situations, pouvant aller d'une nouvelle instance (par exemple, un nouveau concept pour un modèle) à un domaine complètement différent (Maltoni & Lomonaco, 2018).

Dans les mêmes temps, des avancés ont été faites dans le domaine de l'apprentissage par renforcement (en anglais, *Reinforcement Learning* ou *RL*). Un système ayant recourt à des méthodes de RL apprend à partir d'expériences passées à estimer le processus de décision le plus approprié à la situation courante, en maximisant une fonction de récompense prédéfinie. Le RL appliqué aux systèmes de dialogues pour l'apprentissage de politique de dialogue a été largement étudié (Young, 2006), permettant la considération de scénarios *online* (Papangelis *et al.*, 2012) et le développement de techniques d'*Inverse Reinforcement Learning (IRL)* (Chandramohan *et al.*, 2011). Le IRL agit à l'inverse du RL en extrayant la fonction de récompense à partir de l'observation du comportement de l'agent considéré, c'est à dire de l'utilisateur dans le cas des systèmes de dialogue.
Plus récemment, Li et ses collègues (Li *et al.*, 2017) ont appliqué les méthodes de RL à leur système de dialogue, dont le but est de répondre à des questions dans le domaine du cinéma. Le RL permet au système de décider quand poser des questions à l'utilisateur et quelles questions poser, dans le but de permettre au système d'améliorer ses performances.
Li et ses collègues ont identifié trois situations durant lesquelles leur système peut être amené à poser des questions :

- *Problème de compréhension :* Le système demande à l'utilisateur de formuler une périphrase ou de corriger son énoncé. Li et ses collègues ne se concentrent ici que sur des fautes de frappes et des fautes d'orthographe/conjugaison.
- *Problème opérationnel :* Le système a besoin d'aide pour faire le lien entre la demande de l'utilisateur et les connaissances à sa disposition. Le système demande alors un indice à l'utilisateur.
- *Manque de connaissance :* Le système demande à l'utilisateur la réponse.

Certains chercheurs ont aussi montré leur intérêt dans la construction de systèmes de dialogue capables d'apprendre, après avoir été déployés, et ceci de manière continue, grâce à ses interactions avec l'utilisateur. En particulier, Mazumder et ses collègues (Mazumder *et al.*, 2018) ont développé des méthodes et des techniques, permettant à un chatbot d'apprendre de nouvelles connaissances au cours de son utilisation. Pour cela, ils se sont focalisés sur la résolution du problème de complétion d'une base de connaissances, en domaine ouvert, grâce à l'apprentissage par l'interaction avec un utilisateur et grâce au processus d'inférence. Ils ont ainsi mis en place un système stockage pour les connaissances acquises, ainsi qu'un mécanisme permettant d'acquérir des connaissances même si l'utilisateur n'est pas en mesure de fournir les informations requises par le système.
Dans le seconde édition de leur livre (Chen *et al.*, 2018), Chen et Liu ont dédié le 8ème chapitre à l'apprentissage continu de connaissances dans le cadre des chatbots. Dans ce chapitre, ils s'intéressent principalement à l'article présenté précédemment (Mazumder *et al.*, 2018) et déclarent que le principe de LL est reflété par le fait que les faits acquis au cours des interactions sont stockés directement dans la base de connaissances, et utilisés pour réaliser de l'inférence pour de futures requêtes. Il déclarent, de plus, que les connaissances accumulées, ainsi que la nouvelle base de connaissances, permettent

de guider les interactions et l'apprentissage futurs.
De leur côté, Hancock et ses collègues (Hancock *et al.*, 2019) se sont focalisés sur la réalisation d'un système purement conversationnel, capable d'extraire de nouveaux exemples à partir des conversations auxquelles il participe. Ces nouveaux exemples lui permettent par la suite de ré-entraîner son modèle afin de s'améliorer au fur et à mesure de son utilisation. Ce système se base en particulier sur les retours de l'utilisateur : si au cours de la conversation l'utilisateur semble satisfait de la réponse du chabot, celle-ci sera considérée comme un nouvel exemple ; s'il ne semble au contraire pas satisfait, le chabot demandera à l'utilisateur une réponse plus appropriée à la situation, et cette réponse sera considérée comme un nouvel exemple. L'estimation de la satisfaction de l'utilisateur via un classifieur permet au système de s'apercevoir lorsqu'une erreur a été commise. Hancock et ses collègues ont d'ailleurs prouvé que cette approche surpassait celle basée sur l'estimation de l'incertitude du système sur la réponse à donner à l'utilisateur.

Cependant, les deux travaux précédents ne donnent que des exemples spécifiques de systèmes de dialogue réalisant du LL. Nous donnons ainsi dans la section 3 une définition plus globale d'un tel système.

3 Le Lifelong Learning appliqué aux systèmes de dialogue orientés tâches

Nous proposons ici notre définition d'un système de dialogue réalisant du LL et nous décrivons les méthodes et techniques qui pourront être utilisées afin de construire un tel système. Nous discutons de plus des protocoles qui devront être créés afin de permettre l'évaluation de tels systèmes.

3.1 Définitions

Avant de définir ce que pourrait être un système de dialogue réalisant du LL, il faut d'abord se rendre compte que le LL ne doit pas être considéré comme une méthode, à l'image de l'apprentissage par renforcement, mais plutôt comme un objectif. En effet, la nature des connaissances pouvant être acquises, les méthodes impliquées ainsi que les objectifs, dépendent fortement du domaine considéré. Il serait cependant intéressant de définir des concepts et des méthodes pouvant s'appliquer à plusieurs domaines d'étude.

Grâce aux définitions données dans (Chen & Liu, 2016; Chen *et al.*, 2018) et aux travaux décrits précédemment (Li *et al.*, 2017; Mazumder *et al.*, 2018; Hancock *et al.*, 2019), nous pouvons dire qu'un système de dialogue réalisant du LL doit être en mesure d'apprendre de nouvelles connaissances, *après avoir été déployé*, et ceci de manière *continue* grâce à ses *interactions* avec l'utilisateur et possiblement en lui posant des questions. Les connaissances acquises doivent ainsi pouvoir être *stockées*, de manière à ce que le système puisse immédiatement y avoir accès, et pour qu'il puisse directement *adapter* son comportement en fonction. Ces modifications devront de plus être conservées pour de futurs dialogues et les connaissances acquises devront aider le système à *déduire* d'autres connaissances et à *apprendre dans le futur de manière plus efficace et plus rapide*. Lors de ses interactions avec l'utilisateur, le système devra donc avant tout être en mesure de *détecter* lorsqu'une *situation inconnue* se présente et être capable d'en extraire de nouvelles connaissances. La nouvelle connaissance peut correspondre, par exemple, à du vocabulaire, à des informations spécifiques au

domaine d'application, à des faits, à des règles, à un comportement à avoir dans des conditions spécifiques, à un nouvel exemple d'entraînement, etc.

Nous nous focalisons à présent sur les systèmes de dialogue orientés tâche. Ce type de système a pour but d'aider l'utilisateur à réaliser une tâche, ou bien de l'aider à accéder à des informations. Pour cela, le système cherche d'abord à comprendre ce que l'utilisateur lui demande en faisant appel au module de compréhension (NLU) qui réalise alors sur l'énoncé utilisateur une opération de détection des concepts (en anglais, *slot-filling*) et une autre de détection de l'intention (en anglais, *intent detection*). Ensuite, le gestionnaire de dialogue cherche à lier les concepts et l'intention détectés avec les données auxquelles il a accès. Enfin, le système retourne le résultat en générant du langage naturel.

À partir de la précédente définition et des distinctions réalisées dans (Li *et al.*, 2017), nous pouvons remarquer qu'un système de dialogue peut apprendre à différents niveaux :

- *Améliorer les interactions :*
  - améliorer la compréhension des énoncés utilisateur (NLU)
    - améliorer la détection des concepts et des intentions
    - être capable de détecter de nouveaux concepts et de nouvelles intentions
    - être capable de détecter des énoncés hors domaine
  - améliorer la génération de langage naturel (NLG)
- *Enrichir la base de connaissance* : le système peut apprendre de nouvelles informations en enrichissant sa base de connaissance
- *Lier des intentions aux données :* le système est capable d'apprendre à faire le lien entre des intentions utilisateur connues ou nouvelles et des données auxquelles il a accès. Il peut ainsi apprendre à réaliser de nouvelles tâches.

Ainsi, nous considérons qu'un système de dialogue réalisant du LL doit être capable (1) de détecter une nouvelle situation (2) de décider quand et comment agir en réponse à une nouvelle situation afin d'en extraire de nouvelles connaissances (3) de s'adapter à ces nouvelles connaissances. Ces trois points sont détaillés en section 3.2 et sont illustrés dans la section 4.

3.2 Méthodes et techniques associées

Pour chacune des compétences listées en fin de section 3.1 peuvent être associées des méthodes et techniques que nous décrivons ici.

(1) *Détecter une nouvelle situation* :
Lors d'un dialogue, le système peut se rendre compte qu'il est face à une situation inconnue de plusieurs manières. Ceci peut se faire par une estimation de son incertitude via l'étude des scores obtenus en sortie des modèles utilisés par le système. Celui-ci peut, en effet, ne pas être sûr d'avoir bien compris l'utilisateur, ou bien, ne pas être sûr de la meilleure manière de réagir. Le système peut aussi se reposer sur les retours de l'utilisateur qui lui indiqueront s'il l'a mal compris ou s'il n'a pas réagit comme il aurait dû, c'est à dire en estimant sa satisfaction à l'aide d'un classifieur par exemple (Hancock *et al.*, 2019). D'autres manières de détecter une nouvelle situation sont envisageables, mais celles-ci dépendent très souvent du domaine d'application du système de dialogue ainsi que des situations auxquelles il doit faire face.

(2) *Décider quand et comment agir* :
Dans certaines situations, ceci peut revenir à décider quand poser des questions à l'utilisateur, et quelle question poser. Cependant, interroger l'utilisateur implique un coût qui reflète la patience de

l'utilisateur (Li *et al.*, 2017). Le système doit donc être en mesure de s'adapter à l'utilisateur, sachant que certains seront plus enclin à donner de leur temps, tandis que d'autres seront moins patients. De plus certaines questions peuvent être plus intéressantes à poser pour permettre au système de s'améliorer le plus possible. L'idée ici serait ainsi d'avoir recours à des méthodes d'apprentissage par renforcement soit classiques, soit inverses, dont les paramètres seraient évolutifs.

(3) *S'adapter aux nouvelles connaissances* :
Dans le cas des système de dialogue orientés tâche, le système peut s'adapter différemment en fonction de la nature de la connaissance acquise. Il peut notamment enrichir sa base de connaissances, ou ré-entraîner son/ses modèle(s) (par exemple ceux associés au NLU ou au NLG). Ces perspectives soulèvent les deux questions suivantes : sous quelles conditions considère-t-on la nouvelle connaissance comme étant pertinente et/ou fiable ? Et doit-on directement enrichir la base de connaissances ou ré-entraîner le modèle associé dès qu'une nouvelle connaissance est acquise ? Ceci implique de plus une entité de stockage spécifique pour ces connaissances acquises qui n'ont pas encore été considérées comme fiables.
En effet, la question de la fiabilité des connaissances acquises se pose puisqu'il est possible que l'utilisateur transmette des informations incorrectes. Par exemple, dans le cas d'un enrichissement de la base de connaissance, le système peut être amené à demander à l'utilisateur une information spécifique qui manquerait à sa base de connaissance. Le problème est qu'il est possible que l'utilisateur ne connaisse pas la réponse, qu'il peut se tromper ou que la donnée peut être subjective. L'utilisateur peut aussi être amené à faire des fautes de frappes ou d'orthographe ou avoir une manière propre de s'exprimer en fonction de son âge et de son lieu de résidence par exemple.
Dans le cadre de l'enrichissement de la base de connaissance, on peut estimer qu'à partir du moment où une connaissance est estimée fiable, celle-ci peut directement être ajoutée à la base de connaissances du système, puisque le coût associé est généralement faible. Dans le cas, cependant, où l'on voudrait enrichir un modèle, étant donné que le ré-entraînement du modèle implique un coût important, ceci devra se faire sous certaines conditions (par exemple de façon périodique, au bout d'un nombre défini d'exemples fiables acquis, etc).
L'apprentissage de nouvelles connaissances peut cependant mener à une baisse de performance sur les tâches concernant les anciennes connaissances, comme on peut l'observer lorsque l'on est amené à ré-entraîner un modèle avec de nouveaux exemples, comme expliqué dans (Parisi *et al.*, 2018).

3.3 Protocoles d'évaluation envisageables

Développer de tels systèmes demande dans le même temps l'accès à des méthodes et des métriques d'évaluation adaptées. Celles-ci devront nous permettre d'évaluer la plus-value du LL sur un système de dialogue. Les difficultés reposent sur notre capacité à évaluer un système *au cours du temps* et sur l'ensemble des diverses sous-tâches impliquées dans le LL. Cette évaluation nécessitera ainsi une définition précise des tâches impliquées dans le LL pour les systèmes de dialogue ainsi que l'élaboration des corpus d'évaluation associés à chaque tâche. Il s'agira de plus d'étudier si l'évaluation pourra se faire avec des métriques existantes ou si celle-ci nécessitera le développement de métriques spécifiques.
Conformément aux compétences nécessaires listées en section 3.1, devront être évaluées (1) la capacité du système à détecter une nouvelle situation (2) la capacité du système à décider quand et comment agir en réponse à une nouvelle situation afin d'en extraire de nouvelles connaissances et (3) la capacité à s'adapter à ces nouvelles connaissances, tout en considérant la fiabilité de celles-ci.

4 Applications possibles du Lifelong Learning à un chatbot appliqué à la cuisine

Dans le cadre du projet LIHLITH [2], nous avons développé une première version d'un chatbot appliqué au monde de la cuisine [3]. Celui-ci permet notamment à l'utilisateur de trouver une recette correspondant à ses critères, tout en interagissant avec le chatbot en anglais.

Nous commençons en section 4.1 par présenter le système de manière générale. Ensuite en section 4.2, nous illustrons à l'aide d'exemples, comment le LL peut s'appliquer à ce chatbot et discutons des méthodes et techniques envisageables.

4.1 Description générale du chatbot

Le chatbot prend en charge deux types de scénarios :

1. L'utilisateur veut trouver une recette qui répond à ses besoins : le système va alors chercher dans la base de données des recettes. Pour cela, l'utilisateur peut par exemple demander "Please find me a recipe of pancakes without eggs".
2. L'utilisateur pose une question concernant le domaine de la cuisine en général : le système va alors chercher dans des données non structurées en ayant recours à des calculs de similarité textuelle sémantique. Pour cela, l'utilisateur peut par exemple demander "Why is -18˚C the ideal freezer temperature ?".

La base de recettes a été construite à partir d'articles de type recette venant de Wikipedia:Cookbook et comporte *1,064 recettes*. Cette base de données contient notamment des informations sur le nom de la recette, sur les ingrédients nécessaires, sur les étapes à suivre et sur le temps nécessaire.

Le chatbot est divisé en trois sous-systèmes :

— *Le NLU :* Il prend en entrée un énoncé utilisateur et retourne les concepts et l'intention associés [4]. Ce module est basé sur un réseau de neurones profond réalisant à la fois l'étape de *slot-filling* et celle *d'intent detection* (Tur & Mori, 2011), aussi connu sous le nom de *joint NLU*. Nous utilisons pour cela un système opérationnel [5]. Le modèle de compréhension a été entraîné à partir de données qui ont été automatiquement générées et annotées grâce à des patrons et des listes de termes comme proposé dans (Neuraz *et al.*, 2018).

— *Le gestionnaire de dialogue :* Il décide quelles actions réaliser et quelle réponse donner à l'utilisateur en fonction des résultats du NLU et du contexte courant. Il peut en fonction de la situation, accéder à sa base de recettes ou bien chercher des informations dans les données non structurées.

— *Le NLG :* Il génère du langage naturel à partir de patrons de phrases pour permettre au gestionnaire de dialogue de répondre à l'utilisateur

Le gestionnaire de dialogue et le NLG sont tous les deux basés sur des systèmes de règles.

2. LIHLITH : Learning to Interact with Humans by Lifelong Interaction with Humans

3. Le chatbot peut être testé à l'adresse suivante : `https://lihlith.limsi.fr/dialog.php` avec le login "lihlith" et le mot de passe "recipe ?"

4. Dans l'exemple "Please find me a recipe of pancakes without eggs", les slots détectés seraient "recipe : pancakes" et "neg-ingredient : eggs" et l'intention serait "RECING", c'est à dire que l'utilisateur cherche une recette en donnant le nom de la recette ainsi que des ingrédients.

5. Des informations supplémentaires sur ce système peuvent être trouvées sur : https ://github.com/SNUDerek/multiLSTM

4.2 Exemples d'applications possibles du Lifelong Learning

A partir des définitions données en section 3.1, nous avons identifié plusieurs manières de réaliser du LL dans notre chatbot. Les deux possibilités que nous présentons ici ne concernent que la recherche d'une recette selon des critères utilisateurs. Elles correspondent à l'amélioration de la compréhension et à l'enrichissement de la base de données. Des exemples de dialogue illustrant ces possibilités sont donnés dans cette section. Nous discutons de plus de la manière dont nous pourrons les implémenter dans le système, des méthodes et techniques pouvant être envisagées, ainsi que des difficultés pouvant être rencontrées.

4.2.1 Amélioration de la compréhension

Au cours du dialogue, le chatbot doit pouvoir tirer profit de ses interactions avec l'utilisateur afin d'améliorer ses capacités de compréhension. Dans ce but, le système doit être capable d'apprendre à mieux détecter les concepts et les intentions, d'apprendre de nouveaux concepts et de nouvelles intentions, et d'apprendre à les mettre en lien avec les données dont il a accès. Cependant, le système doit d'abord être en mesure de détecter lorsqu'il fait face à une situation inconnue, comme défini dans la section 3.1. Dans notre cas, la nouvelle situation peut s'illustrer par une mauvaise compréhension de l'énoncé utilisateur. Le système peut alors détecter qu'il a mal compris l'utilisateur de plusieurs manières :

1. s'il y a un conflit entre les concepts et l'intention qui ont été détectés [6] ;
2. si le score associé aux concepts et à l'intention détectés sont bas ;
3. si l'utilisateur réagit négativement en disant par exemple "You misunderstood me" ou "You're wrong".

Dans le deuxième cas, il s'agirait d'évaluer l'incertitude du système de compréhension, avant qu'une erreur puisse être commise ; tandis que dans le troisième cas, il s'agirait d'estimer, après coup, la satisfaction de l'utilisateur pour savoir si une erreur a été commise (Hancock *et al.*, 2019). Dans notre cas, il serait intéressant de se demander si la combinaison de l'estimation de la satisfaction et de l'incertitude serait plus profitable que l'utilisation d'une des deux méthodes uniquement.

La Figure 1 illustre un exemple de conflit entre concept et intention ainsi que les décisions qui pourraient être prises par le système afin d'extraire des connaissances à partir de cette nouvelle situation.

Après avoir détecté la nouvelle situation, le système doit décider s'il doit entamer un processus d'échange avec l'utilisateur, afin d'en extraire une nouvelle connaissance. Dans l'exemple, le processus d'échange consiste en la clarification du concept et de l'intention qui auraient dû être détectés. En effet, il n'est pas toujours intéressant pour le système d'entamer ce processus : l'utilisateur peut, par exemple, être peu enclin à aider le chabot à s'améliorer ou il peut être devenu impatient après avoir répondu à plusieurs questions, ce qui peut entraîner le système à ne pouvoir extraire aucune nouvelle connaissance. Cette décision pourrait se reposer sur des méthodes d'apprentissage par renforcement classiques ou bien inverses dont le paramètre de coût et la fonction de récompense seraient évolutifs selon le comportement de l'utilisateur.

6. Par exemple le NLU peut avoir détecté l'intention/la classe RECING, c'est à dire que l'utilisateur cherche une recette à partir du nom de la recette et de critères sur les ingrédients, mais n'avoir détecté que le slot recipe (nom de la recette)

	Énoncé utilisateur	Action du système
U	i'd like to prepare a mousse	Le système n'a pas détecté que "mousse" correspond au concept "recipe" mais a détecté l'intention "REC"
S	Are you asking for a specific recipe?	Demande confirmation sur l'intention détectée
U	Yes	
S	Can you tell me the name of the recipe alone?	Demande le concept manquant
U	Mousse	Enregistre l'énoncé annoté : i'd / like / to / prepare / a / mousse / INTENT O / O / O / O / O / b-recipe / REC
S	I found 3 recipes, including: atkins-friendly chocolate mousse, chocolate mousse cake (vegan) and chocolate mousse . Which one do you want?	Répond à la requête initiale

FIGURE 1 – Exemple de dialogue entre le système (S) et un utilisateur (U) présentant une contradiction entre les concepts et l'intention détectés.

Si le système a pu extraire une connaissance de cette nouvelle situation, il lui faudra non seulement régler le problème au cours du dialogue courant, mais aussi s'adapter à cette nouvelle connaissance et en déduire d'autres connaissances pour de futurs dialogues. Dans l'exemple, la clarification du concept et de l'intention permet au système d'annoter l'énoncé utilisateur dont était issu la contradiction. Cette annotation peut être utilisées de différentes manières. Dans un premier temps, on peut imaginer enregistrer cette annotation dans un fichier spécifique. Ensuite, lorsque le système sera de nouveau confronté à des contradictions entre concepts et intention, il pourra vérifier si la situation qui lui pose problème n'a pas déjà été rencontrée. Cette vérification pourra être faite en comparant la similarité des énoncés annotés enregistrés dans le fichier, avec l'énoncé courant. L'annotation lui permettra alors de résoudre la contradiction sans avoir à déranger l'utilisateur, à la fois dans le dialogue courant, et dans de futurs dialogues. Dans un deuxième temps, on peut imaginer, grâce aux retours utilisateur et aux statistiques d'accès au fichier, qu'un certain nombre d'énoncés annotés peuvent être considérés comme utiles et fiables. Ces énoncés annotés pourront alors servir d'exemples d'entraînement et le système pourra décider sous certaines conditions de ré-entraîner le modèle de compréhension avec ces nouveaux exemples. Lors du ré-entraînement du système, il faudra prêter attention à respecter les proportions entre les classes (intentions) afin de contrer au maximum les pertes de performances liées à l'ajout de nouveaux exemples d'entraînement (Parisi *et al.*, 2018).
Dans notre cas, où nous générons et annotons automatiquement des énoncés utilisateur pour entraîner le modèle de compréhension, il est intéressant de déduire un patron de l'énoncé utilisateur annoté. Dans le cas de l'exemple présenté en Figure 1, le patron issu de la clarification serait "i'd like to prepare a $recipe". Ce patron nous permettra ensuite de générer et d'annoter automatiquement des énoncés supplémentaires pour ré-entraîner le modèle de compréhension. Ceci nous permettra possiblement d'améliorer de façon plus significative les performances du système de compréhension.

Le système peut aussi être amené à détecter une *nouvelle intention* dans l'énoncé utilisateur, comme présenté en Figure 2. Dans cet exemple, l'utilisateur a choisi une recette en particulier et demande des informations supplémentaires (sous-dialogue). L'enjeu ici, est de réussir à comprendre, à partir de l'énoncé utilisateur, qu'aucune intention recensée jusqu'ici ne correspond à la demande de l'utilisateur. On peut imaginer que cette nouvelle intention peut être détectée, par exemple, en observant les scores associés aux concepts et intentions détectés. Nous remarquons cependant que cette détection de

nouvelles intentions peut être difficile à mettre en place, à cause de la complexité et de la diversité de la langue naturelle, et à cause des diverses nouvelles intentions pouvant apparaître. De plus, si la nouvelle intention doit être liée à des concepts, il faudra être capable de replacer ces concepts aux bons endroits. Dans l'exemple, l'intention n'est associée à aucun concept, ce qui facilite grandement le processus.

	Énoncé utilisateur	Action du système
U	i'd like to cook roast chicken	
S	OK, for roast chicken . Do you want the ingredients?	Entre dans le sous-dialogue
U	What temperature for my oven?	
S	I don't know. Can you give me an example of an appropriate answer?	Demande un indice
U	210°C or 370 degree F	
S	I found "Preheat your oven to 200°C (390°F)" Are you satisfied with this answer?	Cherche dans le document recette la phrase qui comporte le plus de similarité avec l'exemple donné par l'utilisateur et demande un retour utilisateur
U	Yes	Déduit des connaissances

FIGURE 2 – Exemple de dialogue entre le système (S) et un utilisateur (U) présentant une nouvelle intention (sous-dialogue).

Pour être capable de traiter la nouvelle intention, nous pouvons imaginer que le système demande à l'utilisateur un indice, en lui demandant un exemple de réponse qui serait appropriée dans la situation courante. Cette méthode suppose cependant que la réponse ait une forme identique selon les contextes et que celle-ci soit simple. Le système pourrait alors chercher dans sa base de données ou dans des documents externes, des données textuelles similaires à l'exemple donné par l'utilisateur, comme envisagé dans l'exemple de la Figure 2. Ceci constitue une première méthode assez simple qui pourra être améliorée.
Après avoir donné sa réponse, et si l'utilisateur valide la pertinence, le système pourrait de plus entrer dans un processus de complétion automatique de sa base de données. En effet, comme proposé dans (Peñas *et al.*, 2019), le système pourrait vérifier dans sa base de données, si cette nouvelle intention peut aussi être appliquée aux autres recettes. Dans l'exemple de la Figure 2, le processus pourrait permettre la création de l'attribut "oven_temperature", et d'associer pour chacune des recettes nécessitant une cuisson au four, une valeur à cet attribut.

4.2.2 Enrichissement de la base de données

Au cours du dialogue, le chatbot devrait aussi être en mesure de tirer profit de ses interactions avec l'utilisateur, afin d'enrichir sa base de données.
Nous considérons l'exemple de la Figure 3, dans lequel l'utilisateur a choisi une recette spécifique (sous-dialogue). On peut imaginer qu'une information manque dans la base de recettes, comme le temps de préparation, la difficulté ou bien le nombre de parts. Dans ce cas, le système pourrait décider d'interroger l'utilisateur sur cette information manquante afin d'enrichir sa base de données. Cet exemple diffère des autres, dans le sens où la détection de la *nouvelle situation* n'est pas effectuée à partir d'un énoncé utilisateur, mais à partir du système lui-même (apprentissage pro-actif), en considérant que la nouvelle situation se manifeste par le manque d'information.

	Énoncé utilisateur	Action du système
S	Hello, are you hungry?	
U	i'd like to cook basic scones	
S	You're telling me about basic scones . Is that OK?	Demande confirmation
U	Ok	
S	OK, for basic scones . Do you want the ingredients?	
U	Yes	
S	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?	Le système a détecté qu'au moins une information manque dans la base de données et décide d'interagir en demandant à l'utilisateur s'il a déjà préparé cette recette.
U	Yes	
S	Do you remember how much time did it take?	Demande l'information manquante
U	I needed approximately 30 min	Ajoute à la base de données "cooking_time = 30 min"
S	OK, thank you very much!	

FIGURE 3 – Exemple de dialogue entre le système (S) et un utilisateur (U) avec un enrichissement de la base de connaissance.

L'apprentissage par renforcement pourrait alors être utilisé afin d'apprendre au système quand il doit demander des informations à l'utilisateur, et sur quelle type d'information l'interroger, comme cela a été fait dans (Li *et al.*, 2017). Le processus s'illustrerait de la manière suivante (cf. Figure 3) :

a. Le système détecte qu'au moins une information est manquante pour la recette courante
b. Pour chacune des informations manquantes, le système estime la récompense cumulée associée au fait de demander cette information. Il estime aussi celle associée au cas où il ne poserait aucune question à l'utilisateur. Ces estimations lui permettent de décider s'il doit interroger l'utilisateur et sur quelle information l'interroger.
c. Si le système a décidé d'interroger l'utilisateur, il commence par lui demander s'il a déjà eu l'occasion de réaliser la recette courante.
d. Si ce n'est pas le cas, le système déduit que l'utilisateur ne pourra lui fournir aucune information et le processus s'arrête là. Dans l'autre cas, le système interroge l'utilisateur sur la question choisie en (b).
e. Si l'utilisateur donne une réponse, le système enregistre celle-ci. Il peut ensuite décider d'interroger l'utilisateur sur une autre information en ayant recours une nouvelle fois à de l'apprentissage par renforcement. Si l'utilisateur n'a pas connaissance de l'information demandée, il utilise aussi la méthode de RL afin de décider s'il interroge l'utilisateur sur une autre information.

La récompense associée au fait d'obtenir une certaine information pourra de plus dépendre du type de l'information et évoluer au cours du temps. Par exemple, si les utilisateurs demandent régulièrement des informations sur le temps de préparation, le système devra associer une plus grande récompense au fait d'obtenir cette information. De même, la fonction de récompense doit s'adapter au niveau de connaissance en cuisine de l'utilisateur courant ; si l'utilisateur connaît peu de chose en cuisine, il est peu intéressant pour le système de l'interroger. Le paramètre de coût pourra aussi évoluer au cours du temps en fonction du comportement de l'utilisateur. Il faudra cependant définir quels

paramètres doivent être partagé entre les différents utilisateurs et ceux qui ne devront être relatifs qu'aux utilisateurs eux-même.
Avant de décider d'ajouter l'information donnée par l'utilisateur à la base de données, le système pourrait la stocker de manière temporaire afin de tester sa fiabilité en questionnant d'autres utilisateurs sur la même recette et sur la même information par exemple.

5 Conclusion et discussion

Dans cet article, nous avons défini ce que pourrait être un système de dialogue orienté-tâche réalisant du LL. Celui-ci doit être en mesure d'apprendre de nouvelles connaissances, après avoir été déployé, et ceci de manière continue grâce à ses interactions avec l'utilisateur. Ensuite, nous avons appliqué cette définition à un un chatbot dédié à la cuisine et montré que celui-ci peut apprendre de deux manières différentes : en améliorant sa compréhension des énoncés utilisateur, et en enrichissant sa base de connaissances. Ces deux axes d'apprentissage pourront de plus être facilement adaptés à d'autres domaines d'applications.

Nous avons ainsi montré que réaliser du LL sur un système de dialogue orienté tâche implique que le système ait la capacité de :

— comprendre à partir des énoncés utilisateur en langage naturel qu'une nouvelle situation se présente, c'est à dire qu'une nouvelle connaissance peut être acquise. Ceci implique des méthodes de détection reposant par exemple sur l'estimation de la satisfaction de l'utilisateur ou de l'incertitude du système.
— décider quand et comment agir en réponse à une nouvelle situation, afin d'en extraire une nouvelle connaissance, tout en ayant recours à des critères évolutifs. Dans ce cas, l'apprentissage par renforcement peut être envisagé afin d'apprendre au système dans quelles conditions il est judicieux pour lui d'interroger l'utilisateur.
— de s'adapter à cette nouvelle connaissance dans le dialogue courant, ainsi que dans les futurs, tout en étant capable de juger de la fiabilité de la nouvelle connaissance. Cette adaptation pourra se traduire dans certains cas par la modification de la base de connaissance du système ou par le ré-entraînement d'un modèle associé au système.

Il faudra de plus prendre en compte le fait que les méthodes citées précédemment impliquent une certaine quantité de données supplémentaires, voire des simulations de données (par exemple des simulations utilisateur). De plus, dans le cadre du LL, ces méthodes et les données associées devront être capable de s'adapter aux nouvelles tâches que le système aura appris à réaliser.

Il faut aussi noter que nous n'avons considéré que des aspects relativement simples du LL appliqué aux systèmes de dialogue orientés tâche, et que nous avons laissé de côté des aspects comme l'amélioration de la qualité du dialogue. Un tel aspect serait par exemple intéressant dans le cas de systèmes uniquement conversationnels.

Remerciements

Cet article est tiré de la traduction de l'article suivant : (Veron *et al.*, 2019). Ce travail a été partiellement financé par le projet LIHLITH (ANR-17-CHR2-0001-03) et soutenu par ERA-Net CHIST-ERA, ainsi que l'Agence Nationale pour la Recherche (ANR).

Références

CHANDRAMOHAN S., GEIST M., LEFEVRE F. & PIETQUIN O. (2011). User simulation in dialogue systems using inverse reinforcement learning. In *Twelfth Annual Conference of the International Speech Communication Association*.

CHEN Z. & LIU B. (2016). *Lifelong Machine Learning*. Morgan & Claypool Publishers.

CHEN Z., LIU B., BRACHMAN R., STONE P. & ROSSI F. (2018). *Lifelong Machine Learning : Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

HANCOCK B., BORDES A., MAZARE P.-E. & WESTON J. (2019). Learning from dialogue after deployment : Feed yourself, chatbot !

LI J., MILLER A. H., CHOPRA S., RANZATO M. & WESTON J. (2017). Learning through dialogue interactions by asking questions. In *ICLR*.

MALTONI D. & LOMONACO V. (2018). Continuous learning in single-incremental-task scenarios. *arXiv preprint arXiv :1806.08568*.

MAZUMDER S., MA N. & LIU B. (2018). Towards a continuous knowledge learning engine for chatbots. *CoRR*, **abs/1802.06024**.

NEURAZ A., CAMPILLOS LLANOS L., BURGUN A. & ROSSET S. (2018). 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.

PAPANGELIS A., KARKALETSIS V. & MAKEDON F. (2012). 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).

PARISI G. I., KEMKER R., PART J. L., KANAN C. & WERMTER S. (2018). Continual lifelong learning with neural networks : A review. *arXiv preprint arXiv :1802.07569*.

PEÑAS A., VERON M., PRADEL C., OTEGI A., ECHEGOYEN G. & RODRIGO A. (2019). Continuous learning for question answering. In *Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*.

THRUN S. & MITCHELL T. M. (1995). Lifelong robot learning. *Robotics and Autonomous Systems*, **15**, 25–46.

TUR G. & MORI R. D. (2011). *Spoken Language Understanding : Systems for Extracting Semantic Information from Speech*. John Wiley & Sons. Google-Books-ID : RDLyT2FythgC.

VERON M., GHANNAY S., LIGOZAT A.-L. & ROSSET S. (2019). Lifelong learning and task-oriented dialogue system : what does it mean ? In *Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*.

YOUNG S. (2006). Using pomdps for dialog management. In *Spoken Language Technology Workshop, 2006. IEEE*, p. 8–13 : IEEE.