Renforcement en-ligne pour l'apprentissage conjoint de l'analyseur sémantique et du gestionnaire de dialogue d'un système d'interaction vocale

Matthieu Riou, Bassam Jabaian, Stéphane Huet, Fabrice Lefèvre

Renforcement en-ligne pour l'apprentissage conjoint de l'analyseur sémantique et du gestionnaire de dialogue d'un système d'interaction vocale

Matthieu Riou Bassam Jabaian Stéphane Huet Fabrice Lefèvre

CERI-LIA, Avignon Université, France

{matthieu.riou, bassam.jabaian, stephane.huet, fabrice.lefevre}@univ-avignon.fr

Résumé

Si la conception des systèmes de dialogue a connu de nombreuses avancées ces dernières années, l'acquisition de grands ensembles de données reste une difficulté pour leur développement rapide dans le cadre d'une nouvelle tâche. L'apprentissage en-ligne est considéré dans cet article comme un moyen pratique de surmonter cette limite. Une fois les modules du système initialisés, un unique processus gère la collection des données, leur annotation et leur utilisation dans les algorithmes d'apprentissage. Il faut alors pouvoir contrôler le coût induit pour l'utilisateur lors de cet apprentissage en-ligne. Notre travail s'intéresse à l'apprentissage simultané des modules d'analyse sémantique et de gestion du dialogue. Dans ce contexte, nous proposons différentes variantes d'apprentissage conjoint qui sont testées avec des tests utilisateurs afin de confirmer que quelques centaines de dialogues d'apprentissage seulement permettent d'atteindre de bonnes performances, améliorant celles d'un système expert à base de règles. L'analyse de ces expérimentations dans l'article fait aussi apparaître des difficultés rencontrées par les entraîneurs du système pour établir une stratégie cohérente et stable durant la phase d'apprentissage.

Mots Clef

Apprentissage en-ligne, bandit contre un adversaire, apprentissage par renforcement, apprentissage sans données de référence, systèmes de dialogue.

Abstract

Design of dialogue systems has witnessed many advances lately, yet acquiring a huge dataset remains a hindrance to their fast development for a new task or language. On-line learning is pursued in this paper as a convenient way to alleviate these difficulties. After the system modules are initiated, a single process handles data collection, annotation and use in training algorithms. A new challenge is to control the cost of the on-line learning borne by the user. Our work focuses on learning the semantic parsing and dialogue management modules. In this context, we propose several variants of simultaneous learning which are tested in user trials to confirm that only a few hundred training dialogues allow us to achieve good performance and overstep a rule-based handcrafted system. The analysis of these experiments gives us some insights, discussed in the paper, about the difficulty for the system's trainers to establish a coherent and constant behavioural strategy to enable a fast and good-quality training phase.

Keywords

On-line learning, adversarial bandit, reinforcement learning, zero-shot learning, spoken dialogue systems.

1 Introduction

Les systèmes récents de dialogue basés sur un apprentissage profond de bout en bout (*end-to-end*) offrent des résultats de recherche très prometteurs [24, 17]. Cependant, ces modèles requièrent une très grande quantité de données pour être entraînés efficacement. Il est alors difficile d'initialiser ce type de systèmes à partir de peu de données, pour un démarrage à chaud de leur développement, suivi par leur apprentissage en-ligne par des utilisateurs, tel que décrit dans [10, 13], même si certains travaux récents ont proposé des architectures de bout en bout [8, 22].

Les expérimentations menées dans cet article sont une première étape dans le développement d'un système de dialogue pour l'évaluation neuroscientifique des compétences sociales dans la communication humaine. L'objectif final est de faire interagir des utilisateurs dans une IRM (Imagerie par Résonance Magnétique) fonctionnelle avec un agent artificiel utilisant notre système, enregistré et diffusé au patient allongé. Ces expérimentations seront faites en français, sur une nouvelle tâche (voir la Section 5) pour laquelle aucune donnée n'est disponible pour le moment. Pour cette raison, la situation n'est pas la même que lorsque des corpus, librement disponibles, sont utilisables [3].

De plus, du fait de l'absence de données préalables, le système présenté repose sur une architecture classique aux capacités éprouvées pour les interactions vocales dirigées par la tâche. Différents modules en cascade permettent, à partir d'informations audio provenant de l'utilisateur, de décoder les mots à partir du signal audio (la reconnaissance automatique de la parole, ASR), puis d'en extraire une interprétation (l'analyseur sémantique, SP) ; cette interprétation est

combinée aux informations récoltées préalablement durant le dialogue (l'état de croyance) pour que la politique de dialogue puisse prendre une décision à propos de la prochaine action à effectuer selon certains critères (longueur du dialogue, accomplissement de la tâche, etc.). Le gestionnaire de dialogue (*dialogue management*, DM) est suivi par plusieurs opérations permettant de transmettre la réponse jusqu'à l'utilisateur : la génération en langage naturel de l'action choisie (*Natural Language Generation*, NLG) puis la synthèse vocale. L'architecture HIS (*Hidden Information State*) [25] offre un tel cadre et permet un apprentissage par renforcement de la politique du DM. Elle peut être mise en œuvre avec des algorithmes efficaces d'apprentissage en échantillons [5] et permet un apprentissage en-ligne par des interactions directes avec des utilisateurs [11]. Plus récemment, l'apprentissage en-ligne a été généralisé aux modules d'entrées/sorties (SP et NLG), avec des protocoles permettant de contrôler le coût de ces opérations durant le développement du système (comme dans [9, 12, 20, 21]). Les travaux présentés ici sont un premier effort pour combiner les apprentissages en-ligne de l'analyseur sémantique et du gestionnaire de dialogue dans une seule et même phase de développement. Le but est d'aider à accélérer et simplifier le processus d'apprentissage, mais aussi de bénéficier des améliorations conjointes des modules.

Dans les systèmes de dialogue, le SP extrait une liste d'hypothèses de concepts sémantiques à partir de la transcription d'une phrase de l'utilisateur. Cette liste s'exprime généralement sous la forme d'une séquence d'actes de dialogue (*Dialogue Act*, DA) de la forme suivante : *acte(concept=valeur)*. Les SP de l'état de l'art se basent sur des approches probabilistes et sont appris avec différentes méthodes d'apprentissage afin d'annoter des entrées utilisateurs avec ces concepts sémantiques [16, 7]. Utiliser ces techniques d'apprentissage supervisé requiert une grande quantité de données annotées qui sont dépendantes du domaine et souvent difficiles à obtenir.

Pour surmonter cet obstacle, un premier exemple de méthode proposée a été un algorithme d'apprentissage sans données de référence pour la classification de phrase sémantique [6]. Cette méthode tente de trouver un lien entre les catégories et les phrases dans un espace sémantique. Un réseau de neurones profond peut être entraîné sur une large quantité de données non annotées et non structurées afin d'apprendre cet espace sémantique. En se basant sur cette idée, [9] a présenté une méthode d'apprentissage sans données de référence pour le SP (*Zero-Shot Semantic Parser*, ZSSP) basé sur les plongements de mots [18]. Cette approche ne requiert ni donnée annotée ni donnée contextuelle et a été utilisée récemment pour différents modules de systèmes de dialogue [23, 26, 1]. En effet, le modèle est amorcé en utilisant seulement la description ontologique du domaine visé et un plongement de mots générique (appris sur un grand corpus de données ouvert librement accessible). Cependant, afin d'améliorer encore les modules avec une supervision éclairée et contrôlée des utilisateurs, une stratégie d'apprentissage actif a été introduite en utilisant un algorithme de bandit contre un adversaire [12].
En prolongeant cette idée, un schéma d'apprentissage actif a été mis en place pour l'apprentissage du DM [11] grâce à l'algorithme par renforcement KTD [14]. Il utilise le lissage des récompenses (*reward shaping*) [19] pour prendre en compte des récompenses locales (basées sur les tours de dialogue) données par l'utilisateur pour offrir un meilleur contrôle sur le processus d'apprentissage et pour l'accélérer.
Puisque des solutions existent pour l'apprentissage actif en-ligne des deux modules SP et DM, nous considérons maintenant leur mise en application simultanée pour gérer l'apprentissage global du système. Tout d'abord, une application directe des techniques existantes est présentée et testée ; les modules et les paramètres de leur apprentissage en-ligne restent séparés (un algorithme de bandit pour le SP, un *Q-learner* pour le DM). Puis une nouvelle proposition avec des paramètres partagés dans un seul *Q-learner* est aussi introduite et évaluée.
Dans la suite de cet article complétant notre précédente publication [21], nous présentons les méthodes d'apprentissage en-ligne du SP dans la Section 2 et du DM dans la Section 3. Nous définissons ensuite les stratégies d'apprentissage en-ligne conjoint dans la Section 4. La Section 5 présente une étude expérimentale avec des évaluations humaines des approches proposées, accompagnées par une analyse nous fournissant des informations pertinentes sur les implications concrètes de l'apprentissage en-ligne. Nous concluons dans la Section 6.

2 Apprentissage en-ligne du Zero-Shot SP

Le modèle SP utilisé dans cet article est le modèle ZSSP décrit dans [12]. Le modèle utilise une base de connaissances sémantiques K et un espace de représentation sémantique F. K contient certains exemples de blocs lexicaux associés chacun à un acte de dialogue DA, tandis que F est une représentation issue d'un plongement de mots appris avec des réseaux de neurones sur une grande quantité de données non annotées d'un domaine ouvert [18, 2]. Le modèle ZSSP construit un graphe pondéré des hypothèses à partir des phrases de l'utilisateur. Un algorithme de recherche du meilleur chemin est utilisé pour trouver la meilleure hypothèse d'annotation sémantique pour la phrase utilisateur considérée.
Le modèle est appris avec une stratégie d'adaptation en-ligne, facilitée par l'approche sans données de référence (*zero-shot*), comme présenté dans [12] et brièvement résumé ici. À chaque itération du dialogue, le système choisit une action d'adaptation $i_t \in \mathcal{I}$ et utilise les retours de l'utilisateur pour mettre à jour K. Les gains du système $g(i_t)$, l'effort de l'utilisateur $\phi(i_t)$ et la fonction de coût $l(i_t)$ sont définis ci-dessous et peuvent être estimés durant l'apprentissage en-ligne.
Trois actions sont possibles :

— **Skip** : Ignore la phase d'adaptation pour ce tour ($\phi(\text{skip}) = 0$).
— **AskConfirm** : Une question oui/non est posée à l'utilisateur à propos de la validité des DA sélectionnés par la meilleure hypothèse sémantique. Si toute la phrase est validée, $\phi(\text{YesNoQuestions}) = 1$. Sinon, $\phi(\text{YesNoQuestions})$ est égal à $1+$ le nombre de DA dans la meilleure hypothèse sémantique (une demande de confirmation oui/non par DA).
— **AskAnnotation** : l'utilisateur est invité à réannoter la phrase entière.
$\phi(\text{AskAnnotation}) = 1$ si la phrase est validée dans son ensemble. Sinon, l'utilisateur devra d'abord spécifier les blocs qu'il souhaite annoter ($+1$ par frontière spécifiée), et ensuite le système demandera, itérativement pour chaque DA, les actes, concepts et valeurs si nécessaire ($+1$ pour chaque question intermédiaire).

Afin de minimiser le nombre de demandes d'annotation et ainsi l'effort demandé à l'utilisateur, un algorithme de bandit contre un adversaire est utilisé pour trouver $i_1, i_2, \ldots, i_t$ tels que pour chaque t, le système minimise le coût $l(i_t)$. La fonction de coût $l(i) \in [0, 1]$ est calculée comme suit :

$$l(i) := \underbrace{\gamma g(i)}_{\text{amélioration du système}} + \underbrace{(1-\gamma)\frac{\phi(i)}{\phi_{max}}}_{\text{effort de l'utilisateur}}$$

où $\gamma \in [0, 1]$ équilibre l'importance de l'amélioration du système avec celle de l'effort de l'utilisateur pour le système, et $\phi_{max} \in \mathbb{N}^*$ est le nombre maximum d'échanges entre le système et l'utilisateur (dans un même tour de dialogue). Dans ce travail, la valeur de γ a été fixée à 0.5.

3 Apprentissage en-ligne du gestionnaire de dialogue par renforcement

Le gestionnaire de dialogue utilisé dans cet article adapte un système présenté dans [11]. Il utilise un *framework* de gestion de dialogue basé sur un POMDP (*Partially Observable Markov Decision Process*), le *Hidden Information State* (HIS) [25]. Dans ce *framework*, le système tient à jour une distribution sur les états possibles du dialogue (l'état de croyance) et l'utilise pour générer une réponse appropriée. Un algorithme d'apprentissage par renforcement (*reinforcement learning*, RL) est utilisé pour entraîner le système en maximisant une récompense cumulée pondérée.

À chaque tour, le gestionnaire de dialogue génère plusieurs réponses possibles selon son état de croyance. En pratique il génère toujours 11 actes de dialogue, correspondant aux 11 actes résumés suivants : Greet, Bye, Bold Request, Tentative Request, Confirm, Find Alternative, Split, Repeat, Offer, Inform and Request. De plus, certains actes peuvent être étiquetés comme impossibles à certains moments si aucune conversion vers une action complète n'est possible (par exemple, Inform dans le cas où aucune entité n'est encore sélectionnée).

Le gestionnaire de dialogue choisit ensuite le meilleur acte résumé pour le contexte donné. Pour apprendre cette politique, une approche d'apprentissage par renforcement est utilisée : l'algorithme d'apprentissage KTDQ [15], dérivée du cadre des *Kalman-based Temporal Differences* (KTD). À chaque tour, la politique sélectionne un acte résumé pour répondre à l'utilisateur, puis ce dernier donne un retour permettant d'attribuer un score à la réponse et de mettre à jour la politique. Deux types de retours sont possibles. Le retour global est donné à la fin du dialogue en demandant à l'utilisateur si le dialogue dans son ensemble est un succès ou non. Le retour direct utilisateur s_i est donné à chaque tour i pour attribuer un score seulement à la dernière réponse. Il est composé de deux parties : le score donné par l'utilisateur à cette dernière réponse (appelé retour additionnel a_i) moins la fonction Ψ (qui prend en compte l'historique d'annotation pour lisser ce retour local), et le coût du tour qui permet de pénaliser les dialogues trop longs en ajoutant un score négatif à chaque tour (appelé retour f_i) : $s_i = f_i + (\theta a_i - \Psi)$.

Ici Ψ est le retour additionnel du tour précédent a_{i-1} et $\theta = 0.95$. À la fin du dialogue, la politique est mise à jour selon la totalité des retours collectés. Dans ce travail, la valeur du retour global est fixée à 20 en cas de succès et 0 sinon. Le retour f_i est fixé à -1 pour chaque tour et le retour additionnel $a_i \in \{-1, -0.5, 0, 0.5, 1\}$.

4 Apprentissage en-ligne conjoint

Afin d'apprendre en-ligne efficacement, l'utilisateur doit être capable d'améliorer à la fois l'analyseur sémantique et le gestionnaire de dialogue. Dans ce but, deux protocoles d'apprentissage différents sont proposés.

Le premier, appelé **BR**, juxtapose le bandit permettant d'apprendre le ZSSP et l'approche *Q-learner* RL permettant d'apprendre le gestionnaire de dialogue [1]. Un algorithme de bandit contre un adversaire tel que décrit dans la Section 2 est appliqué pour l'apprentissage du ZSSP, et un *Q-learner* est utilisé pour apprendre la politique du DM, tel que décrit dans la Section 3.

Le second protocole, appelé **RR**, insère directement les actions d'apprentissage du ZSSP dans la politique RL du gestionnaire de dialogue. Il combine ainsi les deux processus d'apprentissage dans une seule politique [2].

Cette variante de l'apprentissage conjoint fusionne les deux politiques dans un seul *Q-learner*. Dans ce but, le vecteur d'états résumés du DM est augmenté avec des dimensions liées au ZSSP. Pour limiter cet agrandissement, une seule dimension est ajoutée. Cette nouvelle dimension est calculée à partir d'un ensemble d'indices de qualité des annotations faites par le modèle ZSSP. Sur une échelle de 3 points, cinq caractéristiques sont utilisées :

1. BR correspond au protocole Bandit-SP et RL-DM.
2. RR correspond à un système appris avec un RL-SP et un RL-DM.

1. **confidence** : le score de confiance de l'analyseur sémantique sur $[0, 1]$.
2. **fertility** : le ratio du nombre de concepts sur le nombre de mots dans la phrase sur $[0, 1]$, puisque le ZSSP tend à produire une sur-segmentation en insérant des concepts.
3. **rare** : la présence ou non (binaire) de concepts rares dans l'annotation. Ces concepts rares sont help, repeat, restart, reqalts, reqmore, ack or thankyou, et sont souvent mal annotés.
4. **known chunks** : le ratio du nombre de blocs annotés présents dans la base de connaissances sémantiques K sur le nombre total de blocs annotés, compris dans $[0, 1]$.
5. **gap** : la différence entre les scores de confiance de la première et de la deuxième annotations. Puisque ces différences sont très faibles (< 0.01), un logarithme naturel est appliqué pour éclater les données.

La dimension liée au ZSSP est calculée à partir de ces caractéristiques comme suit :

0 *all clear* : rare $= 0$ et confidence $<= 0.499$ et fertility $<= 0.4$ et known chunks $>= 0.5$ et gap $>= -5.5$

1 *average condition* : rare $= 0$ et fertility $<= 0.5$ et known chunks $>= 0.15$ et gap $>= -6.5$ et (confidence > 0.499 ou fertility > 0.4 ou known chunks < 0.5 ou gap < -5.5)

2 *alarming* : rare $= 1$ ou fertility > 0.5 ou known chunks < 0.15 ou gap < -6.5

Dans le protocole RR, les deux actions d'annotation du ZSSP (Askconfirm et Askannotation, voir Section 2) sont aussi incluses dans la liste des actions résumées qui peuvent être sélectionnées par la politique de dialogue. Dans ce cas, la fenêtre appropriée d'annotation est présentée à l'utilisateur dans l'interface graphique du système et permet de corriger l'annotation proposée. L'utilisation d'interactions purement vocales pour ce processus est à l'étude. Même si c'est réalisable, cela reste une tâche compliquée qui pourrait introduire de nouvelles erreurs ; ainsi il semble plus approprié d'évaluer dans un premier temps le processus en utilisant une interface graphique limitant les erreurs. Une fois l'annotation faite, le tour est mis à jour (l'annotation a donc pris la place d'une réponse orale normale de l'utilisateur) et le dialogue reprend. Même si la politique serait capable de l'apprendre d'elle même, nous avons choisi de désactiver les actions d'annotation en séquence (elles sont étiquetées comme impossibles au tour suivant). Finalement, ces deux actions d'annotation du ZSSP ont un retour social spécifique : au lieu d'être fixé à -1, le retour f_i utilise la fonction de coût $l(i)$ définie dans la Section 2 et réajustée pour obtenir un score $\in [-1, 1]$: $f_i = (1.0 - l_i) \times 2 - 1$.

5 Étude expérimentale

5.1 Description de la tâche

Les expériences présentées dans cet article concernent un système de tchat reformulé pour traiter une tâche orientée vers un but. Dans ce contexte, les utilisateurs discutent avec le système à propos d'une image (sélectionnée parmi un petit ensemble prédéfini de 6 images), et essayent de deviner le message porté par l'image en coopération avec le système, comme décrit dans [4]. Afin d'utiliser un système orienté par la tâche, l'application du système a été construite autour d'une base de données contenant plusieurs centaines de combinaisons possibles des caractéristiques de l'image. Chaque combinaison est associée à une hypothèse sur le message envoyé. Durant l'interaction avec le système, il est attendu que l'utilisateur apporte progressivement des éléments de l'image correspondant à une entité dans la base de données. De ce fait, le système sélectionne un petit sous-ensemble des entités possibles depuis lesquelles il peut proposer des caractéristiques additionnelles pour informer l'utilisateur et, ultimement, un message prédéfini proposant une explication plausible de l'objectif de l'image. Cela permet à l'utilisateur de parler assez librement de l'image pendant quelques dizaines de secondes, avant d'argumenter brièvement sur le message. Aucune argumentation n'est possible du côté du système, celui-ci peut seulement proposer un message prédéfini et la discussion est censée durer autour d'une minute au plus.

La base de données, dépendante de la tâche et utilisée dans cette expérience, est dérivée de la tâche de description d'images de fruits proposée par l'INT [4], ainsi que des tâches génériques de dialogue. La sémantique du domaine est décrite par 16 types d'actes, 9 concepts et 51 valeurs. Les 53 formes lexicales utilisées pour modéliser les actes ont été élaborées manuellement.

5.2 Résultats

Cette section présente l'évaluation des deux approches d'apprentissage conjoint. Deux systèmes complémentaires sont proposés en comparaison : **ZH** est un système *baseline* sans apprentissage en-ligne qui utilise le ZSSP initial et une politique de gestionnaire de dialogue prédéfinie par un expert, tandis que **BH** combine l'apprentissage en-ligne par bandit du ZSSP avec la politique prédéfinie du gestionnaire de dialogue.

Trois experts différents ont appris un modèle de chaque approche. Pour chaque apprentissage, l'utilisateur expert communique avec le système pour entraîner le modèle. Puis un groupe de 11 utilisateurs naïfs (ainsi que 2 utilisateurs experts supplémentaires pour le modèle **ZH**) ont testé chaque modèle. À la fin de chaque session, les utilisateurs devaient indiquer si le dialogue était un succès ou non, et donner un score entre 0 (le pire) à 5 (le meilleur) aux capacités de compréhension et de génération du système. La quantité de dialogues pour l'apprentissage et le nombre de tests pour chaque configuration sont donnés dans le Ta-

Modèle	Apprentissage (#dial)	Test (#dial)	Succès (%)	Récompense moy. cum.	Score de compréhension du système	Score de génération du système
ZH	0	142	29	-1,9	1,6	4,0
BH	80	96	70	7,0	3,2	4,6
BR	140	96	89	10,9	3,3	4,6
RR	140	96	65	4,4	2,9	3,8

TABLE 1 – Évaluation des différentes configurations de l'apprentissage en-ligne

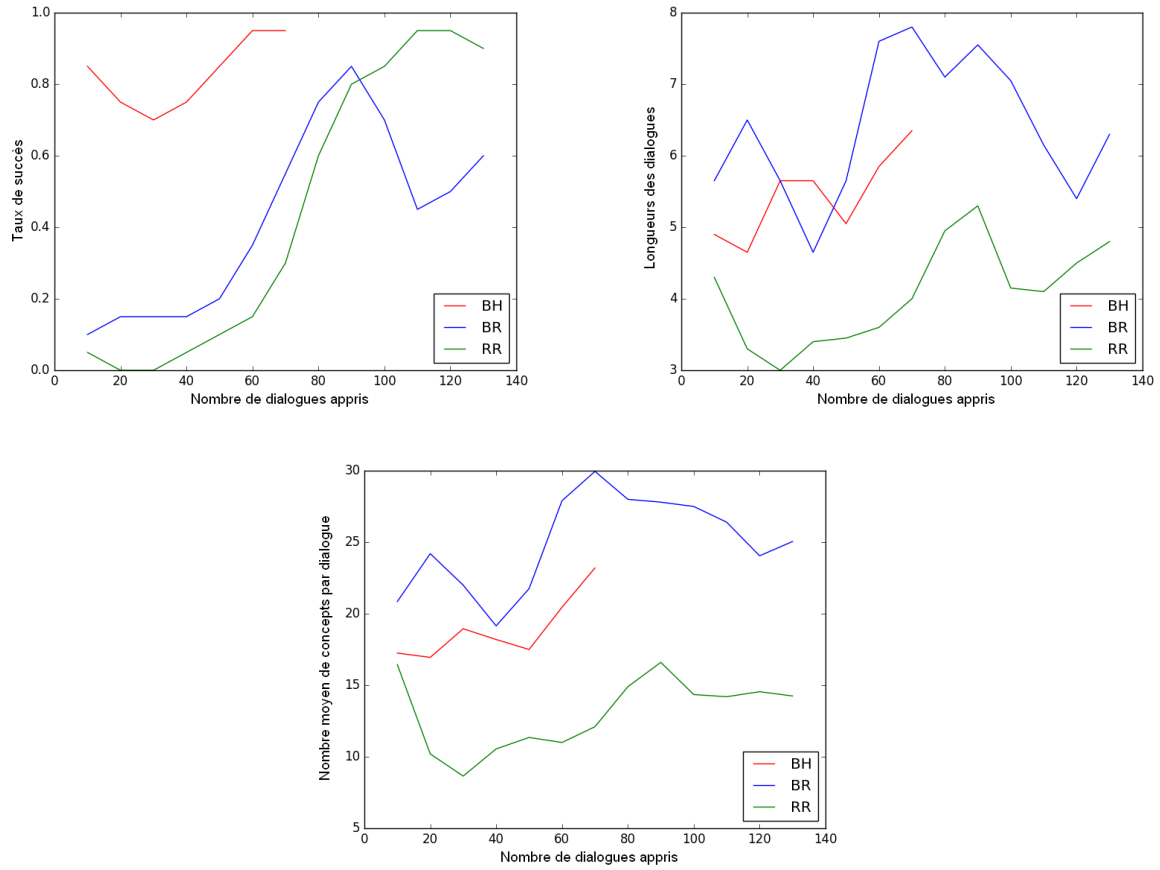

FIGURE 1 – **(a)** Taux de succès **(b)** Longueurs des dialogues **(c)** Nombres de concepts par dialogue durant la phase d'apprentissage pour les différents modèles

bleau 1, aux colonnes 2 et 3.
Les résultats des tests utilisateurs pour les deux apprentissages de chaque approche sont présentés dans le Tableau 1. Ces résultats montrent que les différentes configurations du système offrent des performances acceptables. Le modèle BR, entraîné sur 140 dialogues, montre le meilleur taux de succès (89%), significativement [3] meilleur que les autres modèles. De plus, le modèle ZH donne des taux de succès significativement [3] moins bons que les autres modèles. La différence de performances entre les modèles ZH et BH (+41 points) montre un impact de l'apprentissage du ZSSP sur le succès global d'une conversation, visible aussi dans le score de compréhension (1,6 pour ZH contre 3,2 pour BH). La récompense cumulée moyenne dans les tests est directement corrélée avec le taux de succès et confirme les conclusions précédentes. De plus, en raison de l'ajustement du système de génération à base de patrons, le score de génération du système est élevé ($\geq 3,8$) pour toutes les configurations.

Le protocole RR donne un taux de succès inférieur à BH et BR (65% pour RR contre 89% pour BR). En étudiant l'historique de l'apprentissage, il semble que cela soit dû à un seuil de déclenchement des actions d'apprentissage du ZSSP dans RR beaucoup trop bas après la période d'exploration par rapport à l'utilisation du bandit dans BH et BR. Pour y remédier, l'état de la politique doit être modifié pour mieux prendre en compte les situations qui devraient mener

3. La significativité statistique a été calculée avec un test t de Welch bilatéral. Les résultats sont considérés significatifs pour une p-valeur $< 0,001$.

à des actions d'annotations, tout en préservant ses capacités de discriminations pour les actes de dialogue. Mais cette approche reste à développer et à améliorer, puisqu'elle permet de simplifier grandement le développement de système en étant basée sur un unique *framework* pour l'apprentissage conjoint.

5.3 Analyse de l'apprentissage en-ligne

Afin d'améliorer le protocole d'apprentissage et d'extraire des informations pertinentes de nos expériences, les historiques des apprentissages ont été analysés. Ces apprentissages nous permettent d'avoir une meilleure idée des propriétés du processus d'apprentissage en-ligne et des stratégies implémentées par les experts. Ces derniers étaient assez libres dans leur choix d'apprentissage. Par exemple, ils n'ont eu aucune restriction sur la façon d'utiliser les retours additionnels.

En ce qui concerne la politique du DM, tous les experts disent avoir préféré utiliser dans un premier temps des dialogues simples pour construire une politique efficace. Puis, lorsque le système a commencé à être utilisable (des séquences régulières de 2-3 dialogues ayant été réussies), ils ont proposé des dialogues plus sophistiqués pour étendre les capacités du système. Ces choix sont visibles dans les historiques des apprentissages. Dans les figures 1 (a) et (b), on peut observer que le taux de succès et la longueur des dialogues varient énormément durant l'apprentissage [4] : au début, les dialogues sont courts (avec un minimum de 3 tours pour RR et autour de 6 tours pour BH et BR) et le taux de succès augmente progressivement. Puis, après 40 à 60 dialogues, ils deviennent plus complexes, menant à une diminution de la récompense et du taux de succès, ainsi qu'à de plus longs dialogues. Notons aussi que pour le modèle BH, les experts produisent plus rapidement des dialogues sophistiqués puisque ce modèle ne requiert pas d'apprendre la politique du DM.

Pour approfondir l'analyse de l'apprentissage du ZSSP, nous considérons le nombre total de concepts discutés par dialogue durant l'apprentissage. Pour compter les concepts discutés, chaque triplet acte/concept/valeur a été considéré comme un concept. Les résultats sont présentés dans la Figure 1 (c). Durant les premières étapes, lorsque les experts utilisent des dialogues simples, ils ont plutôt tendance à limiter le nombre de concepts échangés. Puis, au fur et à mesure de l'amélioration du système, ils diversifient les concepts utilisés. L'approche RR montre moins de concepts par dialogue que BH et BR. Cela peut être expliqué par le seuil de déclenchement des actions d'apprentissage du ZSSP très bas pour ce modèle ; ainsi les experts n'ont pas pu étendre efficacement les concepts utilisés durant les dialogues, et ont eu tendance à conserver des expressions en évitant d'ajouter trop de bruits dans les modèles.

Les retours additionnels permettent aux experts de récompenser localement ou pénaliser des réponses spécifiques du système, en complément de la récompense globale (conditionnée au succès de la tâche et pénalisée par la longueur du dialogue). [11] a montré que dans un environnement simulé, les retours négatifs permettent de mieux guider l'apprentissage du dialogue que les retours positifs et améliore la convergence du processus d'apprentissage. Or, nos expériences révèlent que les experts ont instinctivement tendance à plus souvent pénaliser les mauvaises réponses que récompenser les bonnes. On peut l'observer dans la Figure 2 : les retours négatifs sont utilisés régulièrement pour BR et RR (la politique de BH étant prédéfinie, elle n'implique pas de retours utilisateurs). À l'inverse, l'utilisation des retours positifs est plus variable : presque aucun pour BR et plus de retours positifs que négatifs pour RR. Même si l'utilisation élevée de retours positifs demande plus d'efforts aux experts, cela permet aussi d'améliorer la stabilité des choix valides dans la politique de dialogue.

Afin de trouver des améliorations possibles pour le processus d'apprentissage, nous avons étudié les causes régulières d'échecs des dialogues. 48 dialogues de test ont été analysés en détail. Une grande différence des taux de succès entre les modèles peut être expliquée par des erreurs d'annotation de l'analyseur sémantique. Pour étudier cet aspect, une annotation des sorties correctes de l'analyseur sémantique dans le sous-ensemble de dialogues de test a été réalisée par un expert, en précisant en cas d'erreur si elle était due à une erreur de reconnaissance vocale (empêchant toute interprétation correcte ensuite). Ainsi, un score de succès du SP a été calculé comme le ratio du nombre de concepts correctement produits sur le nombre de concepts dans la référence plus le nombre de concepts insérés. ZH présente un taux de succès du SP de 41% contre 70% pour BH, 60% pour BR et 49% pour RR. Ces variations permettent d'expliquer les différences des taux de succès entre les différents modèles.

En regardant des exemples spécifiques d'erreurs du SP (voir Tableau 2), on observe que certaines erreurs sont régulièrement responsables d'échecs de dialogues, notamment :

- une fausse identification d'un acte au revoir, souvent due à des erreurs de reconnaissance vocale, entraînant un arrêt prématuré du dialogue ;
- des erreurs du SP créant de fausses croyances dans le gestionnaire de dialogue, poussant le système à croire que l'utilisateur parle d'une autre image. Ces erreurs sont parfois dues à des erreurs de reconnaissance vocale : environ 10% pour ZH, 25% pour BH, 28% pour BR et 27% pour RR. Mais ces valeurs doivent être liées aux taux de succès du SP : un meilleur SP implique moins d'erreurs et donc une plus grande proportion d'erreurs causées directement par la reconnaissance vocale.

De plus, les analyseurs sémantiques proposent toujours des résultats variables pour les phrases négatives. Par exemple « ce n'est pas un citron » n'est pas comprise par les modèles ZH et RR, mais l'est généralement dans BH et BR.

4. Ces taux sont calculés sur une fenêtre glissante de 20 dialogues.

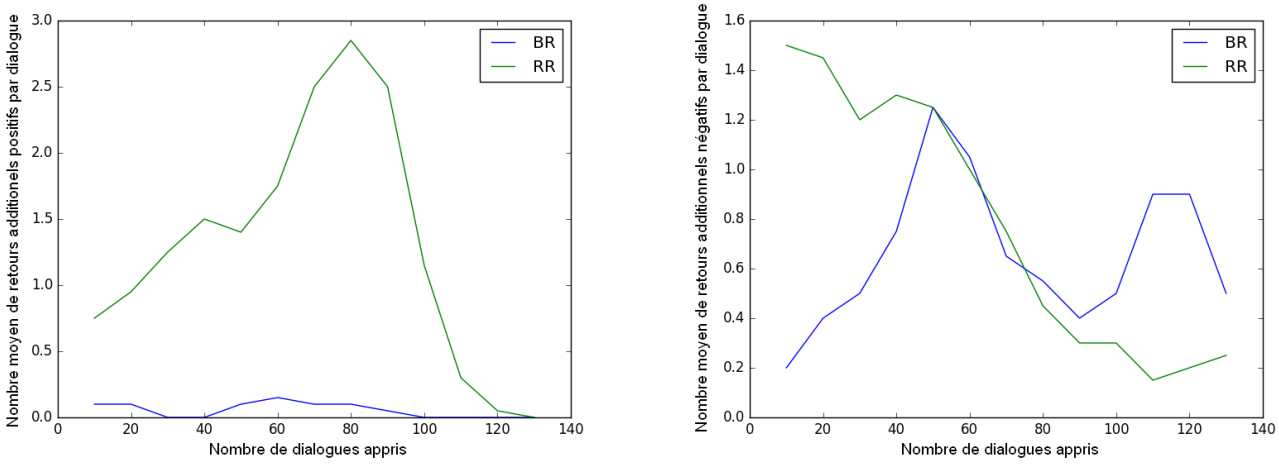

FIGURE 2 – **(a)** Nombre moyen de retours additionnels positifs par dialogue **(b)** Nombre moyen de retours additionnels négatifs par dialogue durant l'apprentissage pour les différents modèles

TABLE 2 – Exemples d'erreurs du SP impactant les dialogues.

ASR	il y a des bras des dieux et des jambes
	L'erreur ASR est due à la proximité phonétique entre dieux *et* yeux .
SP	inform(looks_like=superhero,possesses=legs,possesses=arms)
	Le SP associe incorrectement dieux *au concept de* super-héros.
ASR	le fruit a l'air d'avoir baissé les bras
SP	inform(possesses=bras), reqalts()
	Les expressions figées sont difficiles à interpréter, entraînant des erreurs du SP.
	De plus, le SP a parfois tendance à ajouter des actes comme requalts() *ou* ack() *(en particulier pour ZH et RR).*

Néanmoins, « ce n'est pas un citron, c'est une pomme » reste toujours mal gérée, quel que soit le modèle.

6 Conclusion

Après avoir proposé des méthodes pour entraîner de manière interactive en-ligne à la fois l'analyseur sémantique et le gestionnaire de dialogue, cet article a proposé et évalué différentes manières de combiner ces méthodes dans un processus d'apprentissage conjoint. Les expérimentations ont été menées en conditions réelles et sont donc peu nombreuses. Toutefois, elles ont permis de montrer qu'un apprentissage conjoint peut être effectué, et qu'après environ une centaine de dialogues, les performances des diverses configurations testées étaient généralement satisfaisantes par rapport à un système expert. En outre, ces expérimentations ont permis de relever des informations pertinentes sur les caractéristiques du processus d'apprentissage en-ligne. Par exemple, dans nos expérimentations les experts se concentrent plus souvent sur des dialogues simples dans un premier temps pour augmenter le taux de succès, puis essaient ensuite de produire des dialogues plus complexes pour améliorer à la fois le gestionnaire de dialogue et l'analyseur sémantique. De plus, les retours négatifs sont généralement préférés afin de guider le processus d'apprentissage, ce qui est un bon point puisqu'une précédente étude a montré en simulation qu'ils étaient plus profitables pour le processus d'apprentissage [11].

En se basant sur ces résultats, une prochaine tâche consistera à proposer une fusion des politiques issues de différents apprentissages, afin d'être en mesure de construire des données d'apprentissage issues de différents utilisateurs et d'économiser ainsi encore plus de temps aux développeurs du système.

Remerciements

Ce travail a été partiellement soutenu par un financement de l'ANR-16-CONV-0002 (ILCB) et ANR-11-LABX-0036 (BLRI).

Références

[1] A. Bapna, G. Tur, D. Hakkani-Tur, and L. Heck. Towards zero-shot frame semantic parsing for domain scaling. *arXiv preprint arXiv :1707.02363*, 2017.

[2] J. Bian, B. Gao, and T. Liu. Knowledge-powered deep learning for word embedding. In *ECML*, 2014.

[3] I. Casanueva, P. Budzianowski, P.-H. Su, N. Mrkšić, T.-H. Wen, S. Ultes, L. Rojas-Barahona, S. Young, and M. Gašić. A Benchmarking Environment for

Reinforcement Learning Based Task Oriented Dialogue Management. *arxiv.org*, nov 2017.

[4] T. Chaminade. An experimental approach to study the physiology of natural social interactions. *Interaction Studies*, 18(2) :254–276, 2017.

[5] L. Daubigney, M. Geist, S. Chandramohan, and O. Pietquin. A comprehensive reinforcement learning framework for dialogue management optimization. *Selected Topics in Signal Processing*, 6(8) :891–902, 2012.

[6] Y. Dauphin, G. Tur, D. Hakkani-Tur, and L. Heck. Zero-shot learning and clustering for semantic utterance classification. *arXiv preprint arXiv :1401.0509*, 2014.

[7] A. Deoras and R. Sarikaya. Deep belief network based semantic taggers for spoken language understanding. In *INTERSPEECH*, 2013.

[8] B. Dhingra, L. Li, X. Li, J. Gao, Y.-N. Chen, F. Ahmed, and L. Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, volume 1, pages 484–495, 2017.

[9] E. Ferreira, B. Jabaian, and F. Lefèvre. Online adaptative zero-shot learning spoken language understanding using word-embedding. In *ICASSP*, 2015.

[10] E. Ferreira and F. Lefèvre. Expert-based reward shaping and exploration scheme for boosting policy learning of dialogue management. In *ASRU*, 2013.

[11] E. Ferreira and F. Lefèvre. Reinforcement-learning based dialogue system for human-robot interactions with socially-inspired rewards. *Computer Speech & Language*, 34(1) :256–274, 2015.

[12] E. Ferreira, A. Reiffers-Masson, B. Jabaian, and F. Lefèvre. Adversarial bandit for online interactive active learning of zero-shot spoken language understanding. In *Proceedings of ICASSP*, 2016.

[13] M. Gašić, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *IEEE ICASSP*, pages 8367–8371, 2013.

[14] M. Geist and O. Pietquin. Kalman temporal differences. *Artificial Intelligence Research*, 39(1) :483–532, Sept. 2010.

[15] M. Geist and O. Pietquin. Managing uncertainty within value function approximation in reinforcement learning. In *Active Learning and Experimental Design workshop (collocated with AISTATS 2010), Sardinia, Italy*, volume 92, 2010.

[16] S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, and G. Riccardi. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE TASLP*, 19(6) :1569–1583, 2010.

[17] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, volume 1, pages 733–743, 2017.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[19] A. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations : Theory and application to reward shaping. In *ICML*, 1999.

[20] M. Riou, B. Jabaian, S. Huet, and F. Lefèvre. Online adaptation of an attention-based neural network for natural language generation. In *Proceedings of INTERSPEECH*, 2017.

[21] M. Riou, B. Jabaian, S. Huet, and F. Lefèvre. Joint On-line Learning of a Zero-shot Spoken Semantic Parser and a Reinforcement Learning Dialogue Manager. In *IEEE ICASSP*, 2019.

[22] P. Shah, D. Hakkani-Tur, B. Liu, and G. Tur. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, pages 41–51, 2018.

[23] S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, and L. Heck. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE, 2018.

[24] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. Rojas Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, pages 438–449, 2017.

[25] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model : A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2) :150–174, 2010.

[26] T. Zhao and M. Eskenazi. Zero-shot dialog generation with cross-domain latent actions. *arXiv preprint arXiv :1805.04803*, 2018.