ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : Informatique

École Doctorale 536 « Sciences et Agrosciences »
Laboratoire d'Informatique (EA 4128)

Apprentissage automatique en ligne pour un dialogue homme-machine situé

par
Emmanuel Ferreira

Soutenue publiquement le 14 décembre 2015 devant un jury composé de :

M.	Wolfgang Minker	Professeur, Institute of Communications Engineering, Ulm (DE)	Rapporteur
M.	Alain Dutech	Chargé de Recherche HDR INRIA, LORIA, Nancy	Rapporteur
M.	Joseph Mariani	Directeur de Recherche CNRS, LIMSI, France	Examinateur
M.	Olivier Pietquin	Professeur, LIFL, Lille	Examinateur
M.	Blaise Thomson	Docteur et CEO de VocalIQ, Apple, Cambridge (UK)	Examinateur
M.	Fabrice Lefèvre	Professeur, LIA, Avignon	Directeur
M.	Bassam Jabaian	Maître de conférence, LIA, Avignon	Co-Encadrant

Laboratoire d'Informatique d'Avignon

Remerciements

Avant d'entrer dans le vif du sujet, je tiens à adresser mes sincères remerciements à mes rapporteurs, Wolfgang Minker et Alain Dutech, ainsi qu'à mes examinateurs c'est-à-dire Joseph Mariani, Olivier Pietquin et Blaise Thomson. Je voudrais les remercier d'avoir accepté de consacrer de leur temps à l'étude de mon manuscrit ainsi que pour la qualité et la pertinence de leurs remarques et questions.

Je voudrais également exprimer ma profonde gratitude à mon directeur de thèse, le professeur Fabrice Lefèvre, qui a su m'accorder sa confiance et son soutien tout au long du chemin. Je le remercie pour sa très grande disponibilité ainsi que pour la qualité de son encadrement, autant sur le plan scientifique et professionnel que sur celui humain. Je voudrais aussi remercier chaleureusement Bassam Jabaian, qui a été pour moi un co-encadrant hors pairs avec lequel j'ai passé beaucoup de bons moments, même lorsqu'il s'agissait de boucler des articles à des heures plus qu'indues (*life is life*).

J'ai évidemment une pensée pour l'ensemble des membres du Centre d'Enseignement et de Recherche en Informatique et du Laboratoire Informatique d'Avignon (enseignants, étudiants, visiteurs sans oublier l'équipe administrative) qui ont partagé mon quotidien au cours de ces dernières années et ce depuis mes débuts en informatique. Merci à tous pour votre bonne humeur, vos qualités humaines et ces moments de détente partagés autour d'un café entre collègues. Tout ceci rend pour moi l'ambiance du CERI/LIA inoubliable et me manque dejà. Je voudrais tout même profiter de ces lignes pour faire une dédicace toute particulière à mes camarades des « bancs d'école » et amis, Killian et Cédric, qui depuis ma troisième année de licence n'ont cessé d'être à mes côtés et avec qui j'ai partagé tant de délires ! Merci également à mes deux compagnons de bureau successifs, Mohamed et Imed, qui ont parmi leurs nombreuses qualités celle d'avoir réussi à me supporter moi, mes fantaisies et mes blagues (un peu lourde il faut bien l'avouer) au quotidien !

Merci également au personnel de la médiathèque Jean Tortel de Sorgues, notamment à Chekib, pour m'avoir si bien acceuilli dans leur locaux lors de la finalisation de mon manuscrit et pour avoir même pris le temps de jeter un œuil à mes écrits.

Je ne peux rédiger ces remerciements sans avoir un mot de tendresse pour mes parents, Anne-Marie et Jean-Claude, mes frères et sœurs Nathalie, Maryline, Fabien et Anne-Lys ainsi que pour toute ma belle famille. Je souhaite leur dire que j'ai été très touché par leur soutien, leur confiance et leur fierté manifestés en toutes circonstances.

Enfin, je tiens à dédier ce manuscrit à Doriane, mon épouse, ainsi qu'à nos trois enfants : Milo, Alix et la petite dernière, Célie, arrivée quelques semaines seulement avant ma soutenance. Je ne pourrai jamais quantifier la mesure de ce qu'ils m'ont apporté et permis de réaliser. Merci de m'avoir si souvent redonné la dose de courage et de motivation qui pouvait me manquer et le sourire dans les moments les plus difficiles ! Je vous transmets ce profond et immense amour que je ne réserve que pour vous.

Résumé

Un système de dialogue permet de doter la Machine de la capacité d'interagir de façon naturelle et efficace avec l'Homme. Dans cette thèse nous nous intéressons au développement d'un système de dialogue reposant sur des approches statistiques, et en particulier du cadre formel des Processus Décisionnel de Markov Partiellement Observable, en anglais Partially Observable Markov Decision Process (POMDP), qui à ce jour fait office de référence dans la littérature en ce qui concerne la gestion statistique du dialogue. Ce modèle permet à la fois une prise en compte améliorée de l'incertitude inhérente au traitement des données en provenance de l'utilisateur (notamment la parole) et aussi l'optimisation automatique de la politique d'interaction à partir de données grâce à l'apprentissage par renforcement, en anglais Reinforcement Learning (RL).

Cependant, une des problématiques liées aux approches statistiques est qu'elles nécessitent le recours à une grande quantité de données d'apprentissage pour atteindre des niveaux de performances acceptables. Or, la collecte de telles données est un processus long et coûteux qui nécessite généralement, pour le cas du dialogue, la réalisation de prototypes fonctionnels avec l'intervention d'experts et/ou le développement de solution alternative comme le recours à la simulation d'utilisateurs. En effet, très peu de travaux considèrent à ce jour la possibilité d'un apprentissage de la stratégie de la Machine de part sa mise en situation de zéro (sans apprentissage préalable) face à de vrais utilisateurs. Pourtant cette solution présente un grand intérêt, elle permet par exemple d'inscrire le processus d'apprentissage comme une partie intégrante du cycle de vie d'un système lui offrant la capacité de s'adapter à de nouvelles conditions de façon dynamique et continue.

Dans cette thèse, nous nous attacherons donc à apporter des solutions visant à rendre possible ce démarrage à froid du système mais aussi, à améliorer sa capacité à s'adapter à de nouvelles conditions (extension de domaine, changement d'utilisateur, etc.).

Pour ce faire, nous envisagerons dans un premier temps l'utilisation de l'expertise du domaine (règles expertes) pour guider l'apprentissage initial de la politique d'interaction du système. De même, nous étudierons l'impact de la prise en compte de jugements subjectifs émis par l'utilisateur au fil de l'interaction dans l'apprentissage, notamment dans un contexte de changement de profil d'utilisateur où la politique préalablement apprise doit alors pouvoir s'adapter à de nouvelles conditions. Les résultats

obtenus sur une tâche de référence montrent la possibilité d'apprendre une politique (quasi-)optimale en quelques centaines d'interactions, mais aussi que les informations supplémentaires considérées dans nos propositions sont à même d'accélérer significativement l'apprentissage et d'améliorer la tolérance aux bruits dans la chaîne de traitement.

Dans un second temps nous nous intéresserons à réduire les coûts de développement d'un module de compréhension de la parole utilisé dans l'étiquetage sémantique d'un tour de dialogue. Pour cela, nous exploiterons les récentes avancées dans les techniques de projection des mots dans des espaces vectoriels continus conservant les propriétés syntactiques et sémantiques, pour généraliser à partir des connaissances initiales limitées de la tâche pour comprendre l'utilisateur. Nous nous attacherons aussi à proposer des solutions afin d'enrichir dynamiquement cette connaissance et étudier le rapport de cette technique avec les méthodes statistiques état de l'art. Là encore nos résultats expérimentaux montrent qu'il est possible d'atteindre des performances état de l'art avec très peu de données et de raffiner ces modèles ensuite avec des retours utilisateurs dont le coût peut lui-même être optimisé.

Enfin nous aborderons un autre cadre applicatif, cette fois dans le domaine du dialogue Homme-Robot (tâche spécifique à cette thèse), dans lequel l'apprentissage et les tests du système seront faits par l'intermédiaire d'interactions avec de vrais utilisateurs. Nous profiterons de ce contexte spécifique pour étudier en quoi l'incarnation physique du système au travers du robot peut aider l'interaction et ce notamment grâce à la notion de prise de perspective. En effet, nous proposons dans cette thèse une extension de la méthode de prise de décision mise en œuvre jusqu'alors pour être capable de prendre en compte cette information située dans le mécanisme d'apprentissage de la politique. Ainsi, nous montrons dans cette étude préliminaire que cette information peut sensiblement aider le système à réagir plus naturellement et efficacement.

Abstract

A dialogue system should give the machine the ability to interact naturally and efficiently with humans. In this thesis, we focus on the issue of the development of stochastic dialogue systems. Thus, we especially consider the Partially Observable Markov Decision Process (POMDP) framework which yields state-of-the-art performance on goal-oriented dialogue management tasks. This model enables the system to cope with the communication ambiguities due to noisy channel and also to optimize its dialogue management strategy directly from data with Reinforcement Learning (RL) methods.

Considering statistical approaches often requires the availability of a large amount of training data to reach good performance. However, corpora of interest are seldom readily available and collecting such data is both time consuming and expensive. For instance, it may require a working prototype to initiate preliminary experiments with the support of expert users or to consider other alternatives such as user simulation techniques.

Very few studies to date have considered learning a dialogue strategy from scratch by interacting with real users, yet this solution is of great interest. Indeed, considering the learning process as part of the life cycle of a system offers a principle framework to dynamically adapt the system to new conditions in an online and seamless fashion.

In this thesis, we endeavour to provide solutions to make possible this dialogue system cold start (nearly from scratch) but also to improve its ability to adapt to new conditions in operation (domain extension, new user profile, etc.).

First, we investigate the conditions under which initial expert knowledge (such as expert rules) can be used to accelerate the policy optimization of a learning agent. Similarly, we study how polarized user appraisals gathered throughout the course of the interaction can be integrated into a reinforcement learning-based dialogue manager. More specifically, we discuss how this information can be cast into socially-inspired rewards to speed up the policy optimisation for both efficient task completion and user adaptation in an online learning setting.

The results obtained on a reference task demonstrate that a (quasi-)optimal policy can be learnt in just a few hundred dialogues, but also that the considered additional information is able to significantly accelerate the learning as well as improving the noise tolerance.

Second, we focus on reducing the development cost of the spoken language understanding module. For this, we exploit recent word embedding models (projection of words in a continuous vector space representing syntactic and semantic properties) to generalize from a limited initial knowledge about the dialogue task to enable the machine to instantly understand the user utterances. We also propose to dynamically enrich this knowledge with both active learning techniques and state-of-the-art statistical methods. Our experimental results show that state-of-the-art performance can be obtained with a very limited amount of in-domain and in-context data. We also show that we are able to refine the proposed model by exploiting user returns about the system outputs as well as to optimize our adaptive learning with an adversarial bandit algorithm to successfully balance the trade-off between user effort and module performance.

Finally, we study how the physical embodiment of a dialogue system in a humanoid robot can help the interaction in a dedicated Human-Robot application where dialogue system learning and testing are carried out with real users. Indeed, in this thesis we propose an extension of the previously considered decision-making techniques to be able to take into account the robot's awareness of the users' belief (perspective taking) in a RL-based situated dialogue management optimisation procedure. Thus, we show in this preliminary study that this information can enable the system to cope with both the communication ambiguities due to noisy channel and the possible misunderstandings due to some divergence among the beliefs between the robot and its interlocutor.

Table des matières

Chapitre 1

Introduction

Sommaire

1.1 Contexte général

Le terme « dialogue » est défini dans le Trésor de la Langue Française [1] par « Communication le plus souvent verbale entre deux personnes ou groupes de personnes ». Cependant, peut-on encore à ce jour limiter un dialogue à des participants humains comme semble le suggérer cette définition ? Ou peut-on plus largement l'étendre à une communication entre entités dotées de capacités cognitives ?

Comme l'atteste l'apparition et la généralisation progressive des assistants personnels vocaux (Siri, Cortana, Google Now, etc.) sur nos téléphones intelligents (*smartphones*), ou encore la démocratisation des logiciels de dictée vocale (Dragon, Philips, etc.), le fait d'interagir vocalement avec les outils électroniques du quotidien commence à entrer progressivement dans les mœurs. Si les performances actuelles de ces solutions commerciales sont souvent en deçà des attentes clients et ne permettent pas encore de pouvoir envisager la tenue d'un véritable dialogue intelligible entre un Homme et une Machine, des initiatives, comme celle prise par Apple, avec le rachat cet octobre (2015) de la startup anglaise VocalIQ fondée par des universitaires de Cambridge, visent à progressivement améliorer cette expérience.

La différence entre « dialogue » et « outil de dictée » capable d'interpréter des commandes vocales est de taille. En effet, le fait de maintenir une interaction nécessite de savoir donner véritablement un sens aux informations transmises mais également de

1. http ://atilf.atilf.fr/tlf.htm

générer des réponses élaborées, cohérentes et compréhensibles par l'interlocuteur humain afin de poursuivre l'interaction. Aussi, si la question commence à se poser pour un ordinateur ou un smartphone, au combien se pose-t-elle pour un robot domestique, qui outre les capacités dont disposent les premiers, partagerait en plus notre environnement, et serait capable de s'y mouvoir et d'y agir.

Dans l'inconscient collectif, la notion de machine pensante, capable d'apprendre par l'expérience et de prendre des décisions dans le monde qui nous entoure inspire encore la crainte. Par exemple, des personnalités de renom, tel le célèbre astrophysicien britannique Stephen Hawking ou encore l'entrepreneur américain Bill Gates y voient l'augure d'un terrible danger pour l'humanité. Ce ressentiment est également partagé par bon nombre de concitoyens européens comme l'attestent les résultats d'une vaste étude d'opinion [2] commandée par la Commission européenne au sondeur TNS sur des échantillons de population des 28 pays membres de l'UE et qui place la France dans le « Top 6 » des nations plutôt défavorables au développement du secteur robotique (avec 52 % des sondés contre). Si la science-fiction a sa part de responsabilité dans la chose, avec la création de personnages tels que HAL 9000 l'ordinateur assassin de « 2001, Odyssée de l'espace » de Kubrick ou du robot vengeur *Terminator* de James Cameron, des angoisses sociétaires comme la suppression/dévalorisation d'emplois peu qualifiés sont cependant plus concrètes.

Malgré ces réticences, les robots sont appelés à progressivement intégrer l'environnement domestique de tout un chacun comme c'est le cas au niveau industriel depuis quelques années (bras industriel, robots aspirateurs, robots humanoïdes, etc.). À l'instar du Japon, de la Corée, ou encore des États-Unis, l'Europe considère le secteur de la robotique de service et domestique (robots assistants/équipiers) comme un des enjeux économiques de ce siècle, comme l'attestent ses nombreux financements FP7 et Horizon 2020 (RoboHow, ICARUS, MuMMER, etc.). En effet, on estime que d'ici à 2020, le marché de la robotique de services (tous secteurs confondus) pourrait représenter un volume supérieur à 100 milliards d'euros par an [3]. La robotique domestique pourrait par exemple contribuer à minimiser les coûts associés aux dégradations de l'autonomie d'une certaine part de la population en permettant :

— leur inclusion dans la société numérique via les technologies de communication moderne ;
— un maintien du lien social en facilitant l'accès à des technologies comme Internet ;
— de renforcer un sentiment de dignité en permettant de prolonger l'autonomie à domicile.

L'originalité de l'approche européenne réside cependant dans une vision de la robotique plus orientée vers l'apprentissage et l'intelligence artificielle afin d'envisager le développement de solutions pérennes par définition adaptatives et capables d'être optimisées à partir de données.

En s'inscrivant dans un point de vue semblable, nous nous appliquerons dans cette thèse à proposer des techniques d'apprentissage automatique permettant d'améliorer

2. http ://ec.europa.eu/public_opinion/archives/ebs/ebs_427_en.pdf
3. http ://europa.eu/rapid/press-release_IP-12-978_fr.htm ?locale=FR

la qualité des systèmes (vocaux et/ou robotiques) grâce à leur mise en confrontation directe face à de vrais utilisateurs en situation d'interaction. L'objectif visé par notre approche est de réduire leur coût de développement sur de nouvelles tâches, mais aussi d'améliorer leur niveau de robustesse, d'efficacité et de naturel général pour gagner l'acceptation du grand public.

1.2 Motivations de la thèse

Cette thèse s'inscrit dans le cadre du projet de l'Agence Nationale pour la Recherche, *ANR*, *MaRDi*[4], financé dans le cadre de l'appel à projet Contenu et Interactions. Les travaux réalisés dans ce cadre ont été faits en collaboration avec le Laboratoire d'Informatique Fondamentale de Lille (LIFL), l'École supérieure d'électricité (Supélec), le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), le groupe Acapela et le Laboratoire Informatique d'Avigon (LIA).

Ce projet a pour axe d'étude l'apport d'une approche « située » du dialogue Homme-Machine. Le terme « situé » est ici relatif à l'incarnation physique d'un système de dialogue dans une plateforme robotique qui va permettre d'envisager l'intégration d'informations issues des perceptions physiques du robot dans le contexte de l'interaction pour espérer compléter ou lever des ambiguïtés introduites par le medium vocal. Ce projet s'inscrit également dans l'utilisation de méthodes d'apprentissage numérique, exploitant les données collectées au travers de la conduite de véritables interactions afin d'améliorer l'efficacité et le naturel du système dans le temps. L'originalité de l'approche est de ne pas considérer les technologies vocales comme disponibles et dissociées de la tâche d'interaction Homme-Robot, mais bel et bien comme moyen d'en améliorer l'expérience et les performances.

Pour atteindre ce but, la Machine doit être capable de maintenir un contexte d'interaction suffisamment riche pour pouvoir être à même de prendre des décisions sur la suite à donner à celle-ci. Ce contexte intègrera les entrées fournies par l'humain mais aussi les informations issues de la perception de l'environnement et des proprioceptions du robot (mesures du système sur lui-même comme par exemple l'angle de rotation de ses armatures, le niveau charge de sa batterie, etc.). Sur ces aspects en particulier, le projet étend les recherches menées par le LAAS dans le domaine du raisonnement spatial et surtout de la prise de perspective pour tenter de résoudre des ambiguïtés. La prise de perspective est un processus par lequel une machine adopte le point de vue d'un autre agent (humain ou artificiel) afin de raisonner sur ce qui peut être vu par l'un ou l'autre.

Pour prendre des décisions afin de poursuivre l'interaction, la machine devra s'appuyer sur un contexte de l'interaction (historique, état de l'environnement, etc.) et tenir compte de son aspect incertain. En effet, dans le cadre situé, le contexte de l'interaction ne peut être considéré comme une donnée sûre car de possibles erreurs ont pu être introduites par la chaîne de traitement automatique des entrées vocales et visuelles.

4. Man-Robot Dialogue - http ://mardi.metz.supelec.fr

C'est pourquoi les approches stochastiques que nous employons nous permettent de modéliser les différentes hypothèses avec leur score de confiance respectif. Ainsi, la stratégie d'interaction employée par la machine devra tenir compte des ambiguïtés potentiellement générées et en garder trace tout au long du dialogue. Nous traiterons cette problématique par l'utilisation de modèles permettant l'optimisation statistique du mécanisme de prise de décision. La faculté d'adaptation à un nouveau profil utilisateur ou plus généralement à des situations contextuelles et dialogiques différentes lors de l'apprentissage est une caractéristique désirée qui fera également l'objet de nos travaux.

Une fois sa décision prise, le système devra également pouvoir la restituer vocalement et physiquement à l'humain. Ainsi il faudra qu'il soit capable de planifier ses mouvements et ses actions physiques de façon précise pour répondre aux besoins de l'utilisateur (déplacer un objet, se rendre dans une pièce, etc.), mais également capable de s'exprimer de façon adéquate pour se faire comprendre par l'utilisateur et lui témoigner de sa compréhension du contexte interactif courant. Pour cela, le robot devra par exemple adopter des attitudes physiques particulières, ou encore utiliser un timbre de voix particulier.

Étant donné que le LIA a à sa charge la réalisation de la plateforme de dialogue *MaRDi*, nous nous intéresserons tout particulièrement dans ce manuscrit aux problématiques de la gestion de l'interaction et de la compréhension de la parole et des gestes expressifs. Compte tenu de l'aspect innovant du projet, il a fallu procéder à un développement de « zéro » du système (aucun prototype ni données d'apprentissage à notre disposition). Face à la complexité et aux coûts qu'engendrent la réalisation d'une collecte de données sur une tâche de cette nature (interaction réelle entre un robot et un utilisateur coûteuse en temps et en moyens déployés), notre objectif est de limiter notre dépendance en exploitant au mieux les interactions préliminaires (situées ou non) grâce à des techniques d'apprentissage statistique efficaces. Ainsi, nous avons considéré en première instance l'utilisation d'une plateforme de dialogue minimale puis l'automatisation de son amélioration grâce à son utilisation face à de véritables utilisateurs. Ce faisant, nous avons proposé le développement d'une plateforme capable de s'adapter plus facilement à des conditions changeantes (profils utilisateur, contexte).

Cette solution constitue une alternative concrète aux techniques d'apprentissage classiques. En effet, dans la littérature, le processus d'optimisation d'une politique d'interaction est souvent décrit comme un processus en deux étapes. Dans une première étape, des interactions préliminaires (généralement exploitant un système télé-opéré par un « expert ») sont réalisées pour collecter des données d'apprentissage, puis dans une seconde étape, ces données sont utilisées pour concevoir une première solution. Ici, nous proposons au contraire que le système apprenne de « zéro » (qui sera synonyme d'absence de données d'apprentissage dans le reste du manuscrit) grâce à sa mise en situation directe face à des utilisateurs. Pour faciliter la tâche, nous prendrons cependant la précaution dans les phases initiales de l'apprentissage d'avoir recours à des utilisateurs plus tolérants à l'échec, comme par exemple le concepteur du système ou un panel réduit d'utilisateurs correctement informés de la situation et capables d'agir en conséquence (permettre l'amélioration de la situation du système).

1.3 Contributions associées

Dans cette thèse, nous nous intéresserons au développement d'un système de dialogue reposant sur des approches statistiques, tout particulièrement au système faisant usage du cadre formel des Processus Décisionnel de Markov Partiellement Observable, en anglais *Partially Observable Markov Decision Process* (POMDP), qui à ce jour fait office d'état de l'art dans la littérature en ce qui concerne la gestion statistique du dialogue. Ce modèle permet à la fois d'intégrer proprement l'incertitude inhérente au traitement des données en provenance de l'utilisateur (par exemple la parole) mais aussi de pouvoir envisager l'optimisation automatique de la politique d'interaction à partir de données grâce à l'apprentissage par renforcement, en anglais *Reinforcement Learning* (RL).

Cependant, une des problématique liée à l'adoption d'approches statistiques est qu'elle nécessite le recours à un grand nombre de données d'apprentissage pour atteindre des niveaux de performance acceptables. Or, la collecte de telles données est un processus souvent long et couteux qui nécessite généralement pour le cas du dialogue la réalisation de prototypes fonctionnels avec l'intervention d'experts et/ou le développement de solutions alternatives comme le recours à la simulation d'utilisateurs.

Ainsi, nous étudierons la faisabilité d'un démarrage « à froid » d'un tel système par l'établissement d'un cadre d'apprentissage efficace et adaptatif capable de tirer parti des interactions que le système est en train de réaliser avec de vrais utilisateurs. Nous ferons également l'étude d'un système capable de prendre en compte dans ses décisions l'aspect situé que lui confère son incarnation physique dans le cadre de l'interaction Homme-Robot.

De ce fait, nous établirons plusieurs solutions pour accélérer l'apprentissage de la stratégie d'interaction mise en œuvre pour guider le dialogue. Ce faisant, nous avons pour idée de limiter le nombre d'interactions nécessaires pour atteindre un niveau de performance donnée, tout en conservant la propriété d'atteinte rapide de l'optimalité. A l'instar de récents travaux tels que (Gašić et al., 2010; Gašić et al., 2011; Sungjin et Eskenazi, 2012; Daubigney et al., 2012), nous ferons l'usage d'un algorithme dit « efficace par échantillon », ici KTD (Geist et Pietquin, 2010), capable d'apprendre une politique (quasi)-optimale de zéro en quelques centaines d'interactions (contre plusieurs dizaines de milliers dans des configurations antérieures). De plus, nous envisagerons l'intégration de deux sources d'informations supplémentaires pour accélérer cette apprentissage, à savoir des connaissances expertes initiales sur la tâche d'interaction et les jugements subjectifs émis au cours d'une interaction par l'utilisateur. Ces derniers seront également évalués dans leurs capacités à pouvoir aider l'adaptation de la stratégie d'interaction à de nouvelles conditions (nouveaux profils utilisateurs).

Nous nous intéresserons également à proposer une technique visant à réduire les coûts de développement d'un module de compréhension de la parole utilisé dans l'étiquetage sémantique d'un tour de dialogue. Pour cela, nous exploiterons les avancées récentes dans les techniques de projection de mots dans des espaces vectoriels continus conservant les propriétés syntactiques et sémantiques, comme l'approche word2vec. Elles offre un mécanisme permettant de généraliser les connaissances initiales de la

tâche (base de données). Nous nous attacherons aussi à proposer des solutions pour enrichir dynamiquement cette connaissance et étudier le rapport de cette technique avec les méthodes statistiques de l'état de l'art.

Enfin, nous appliquerons notre vision de l'apprentissage dans une situation concrète de développement d'un nouveau système de dialogue dans le domaine du dialogue Homme-Robot (tâche spécifique à cette thèse), dans lequel l'apprentissage et le tests du système seront faits par l'intermédiaire d'interactions avec de vrais utilisateurs. Nous profiterons de ce contexte applicatif spécifique pour étudier dans quelle mesure l'incarnation physique du système, au travers du robot, peut aider l'interaction et ce notamment grâce à la notion de prise de perspective. En effet, nous proposons dans cette thèse une extension de la méthode de prise de décision mise en œuvre jusqu'alors dans le gestionnaire de dialogue pour être capable de prendre en compte cette information située dans le mécanisme d'apprentissage de la politique d'interaction et ainsi permettre de désambiguïser en amont certaines situations complexes (par exemple les situations de fausses croyances entre le Robot et l'Homme, que nous expliciterons plus avant dans la thèse).

1.4 Organisation du manuscrit

Ce document est organisé en deux parties.

La première partie s'attache à établir le cadre théorique de nos travaux mais également à dresser un état de l'art des différentes techniques employées pour rendre possible l'interaction entre l'Homme et la Machine. Pour ce faire, le chapitre 2 présentera de façon générale les systèmes de dialogue au travers de la description de leurs principaux composants et des différentes méthodes employées pour permettre une interaction efficace. Nous profiterons de ce chapitre pour discuter du problème complexe de l'évaluation de leurs performances. Le chapitre 3 aura pour objectif de décrire le cadre de l'apprentissage par renforcement utilisé par le système pour progressivement améliorer sa capacité de conduite de l'interaction. Nous décrirons notamment le paradigme POMDP retenu dans notre étude ainsi que les réponses proposées dans la littérature pour l'appliquer au contexte du dialogue. Par la suite, nous discuterons tout particulièrement du contexte d'apprentissage pour justifier les choix faits dans nos travaux.

La seconde partie concerne nos contributions. Elle s'attardera à présenter nos différentes propositions pour établir un cadre permettant d'accélérer le déploiement d'un système de dialogue sur une nouvelle tâche. Pour cela, dans le chapitre 4 nous présenterons en détails les solutions envisagées pour permettre un apprentissage efficace de la politique d'interaction ainsi que leurs résultats en condition de simulation. L'objectif étant d'exploiter dans l'apprentissage des informations supplémentaires issues soit de l'expertise du domaine, soit directement de retours subjectifs émis par l'utilisateur (apprentissage socialement inspiré). Le chapitre 5 présentera une approche visant à limiter les coûts de développement d'un module de compréhension de la parole ainsi qu'une extension permettant son raffinement en ligne. Nous profiterons également de

ce chapitre pour présenter une technique visant à optimiser dynamiquement la stratégie de raffinement du modèle adoptée pour essayer de trouver un compromis entre coût et amélioration du modèle, mais également une solution permettant d'y introduire un mécanisme d'apprentissage supervisé plus classique. Enfin, le chapitre 6 sera dédié à la présentation de la plateforme de dialogue développée dans le cadre du projet *MaRDi*, et des premiers résultats obtenus avec des interactions réelles. Nous profiterons notamment de ces expériences pour valider sur un cas pratique, avec de vrais utilisateurs, l'apprentissage socialement inspiré introduit dans le chapitre 4. Pour conclure, nous présenterons une première tentative visant à intégrer dans l'apprentissage de la politique d'interaction des informations liées à l'aspect situé de l'interaction Homme-Robot, et plus exactement grâce à l'étude de situations de fausses croyances.

Tous les détails non strictement nécessaires à la compréhension de nos propositions mais présentant un intérêt pour qui veut reproduire nos expériences figurent en Annexe, ainsi quelques détails sur l'implémentation des différents systèmes, les métriques utilisées et les tâches considérées.

Première partie

Cadre théorique et état de l'art

Dans cette partie nous posons le cadre théorique nécessaire à la bonne compréhension du reste de ce manuscrit. Pour cela, nous dresserons un état de l'art de la recherche sur les systèmes de dialogue Homme-Machine. L' objectif de cette démarche est de pouvoir situer nos travaux dans le contexte qui leur a permis de voir le jour. Ainsi, nous nous intéresserons aux méthodes actuellement explorées par la communauté du dialogue, leurs éventuelles limitations et les enjeux scientifiques soulevés.

Le chapitre 2 présente quelques généralités sur les systèmes de dialogue Homme-Machine et leurs architectures dans la littérature. S'en suit une présentation des différents mécanismes de gestion de l'interaction dans la section 2.2, allant des méthodes expertes au approches statistiques qui seront employées dans nos travaux.

Le chapitre suivant décrit les méthodes d'apprentissage par renforcement et vient ainsi étayer l'adoption de ce paradigme pour la gestion du dialogue et tout particulièrement celui cadre formel du processus de décision de Markov partiellement observable (POMDP). Nous abordons ensuite la question de l'apprentissage et de l'évaluation de ces systèmes.

Chapitre 2

Présentation générale des systèmes de dialogue

Sommaire

Dans sa définition première, le dialogue est une « *conversation entre deux ou plusieurs personnes sur un sujet défini* »[1]. Derrière l'utilisation du terme « système de dialogue » se trouve l'idée qu'au moins un des participants à cette conversation est une Machine au sens large du terme (ordinateur, smartphone, robot, etc.). Néanmoins, cette interaction pouvant prendre de très nombreuses formes (chat textuel, serveur vocal, agent virtuel, etc.), il est difficile d'en dresser un état de l'art exhaustif. Dans cette thèse, nous nous intéresserons exclusivement aux dialogues s'effectuant entre deux participants, à savoir l'Homme (appelé également utilisateur par la suite) et la Machine, avec pour particularité d'avoir la parole pour moyen de communication dominant (Mori, 1997; McTear, 2004; Minker et Bennacef, 2004). De plus, nous nous concentrerons sur l'étude de dialogues dont le but est de satisfaire des demandes/tâches utilisateurs précises (réservation de billets de train, exécution d'une tâche de manipulation d'objet, etc.) dans des domaines applicatifs bien définis (recherche d'information touristique, assistant personnel sur smartphone, etc.).

1. Définition donnée par le dictionnaire Larousse

Si des applications de ce type de système existent dans l'industrie depuis quelques années, notamment pour réduire les coûts liés à l'emploi d'opérateurs humains dans les centres d'appels, nous assistons aujourd'hui aux prémices d'une réelle généralisation de ces approches dans notre quotidien. C'est par exemple le cas sur nos smartphones avec le développement d'applications d'assistant vocaux (Siri d'Apple, Cortana de Microsoft ou encore Google Now de Google) ou sur nos navigateurs web (Google Chrome par exemple) où il est désormais commun de pouvoir procéder à un certain nombre de tâches par le biais de la voix (recherche d'informations, écriture/envoi d'un e-mail, consultation de la météo, etc.). Cependant, malgré l'intérêt grandissant en termes applicatifs, l'expérience qui en est faite demeure assez frustrante pour les utilisateurs. En effet, les systèmes de dialogue actuels sont encore confrontés à un certain nombre de limites (capacité de compréhension du système restreinte, absence d'un réel système de gestion de l'historique d'interaction, limites a quelques domaines, peu d'adaptation à l'utilisateur, etc.) pour réellement envisager la tenue d'une conversation naturelle (c'est-à-dire plus proche de ce qui caractérise l'interaction Homme-Homme). Ainsi, augmenter l'efficacité (donner l'information pertinente au bon moment) et la qualité de l'interaction (avoir un comportement adéquat capable de surprendre positivement l'utilisateur) constituent deux des préoccupations principales de la recherche concernant les systèmes de dialogue.

Dans ce chapitre, nous présenterons les principales solutions proposées dans la littérature pour tendre vers ce but et ainsi dépasser les limites des approches actuellement utilisées dans la majorité des services déployés. Pour ce faire, avec la section 2.1 , nous commencerons par décrire les principaux composants nécessaire à la mise en œuvre d'un système de dialogue. Puis, dans la section 2.2, nous nous concentrerons sur les principales méthodes capables de répondre à la problématique de la gestion du cours de l'interaction. Nous distinguerons pour cela les approches déterministes (section 2.2.1) des approches statistiques (section 2.2.2). Et pour finir, la section 2.3 sera dédiée à la présentation des techniques qui peuvent être employées pour évaluer et comparer les différents systèmes de dialogue entre eux.

2.1 Les différents composants d'un système de dialogue

Un système de dialogue peut se décliner sous de nombreuses formes aux caractéristiques variées (centre d'appel, assistant personnel sur smartphone ou encore robot compagnon). Ceci explique notamment pourquoi il est difficile de trouver dans la littérature un standard architectural unifié entre les différents systèmes de dialogue. De nombreuses propositions existent et l'on peut par exemple faire référence aux approches dites à « domaine ouvert » (*open-domain* en anglais) telles que celles présentées dans (Strzalkowski et Harabagiu, 2006; Lcock, 2012). Ces dernières reposent principalement sur l'exploitation de sources de connaissances généralistes comme le web sémantique (DBpedia [2], Freebase [3], etc.), pour permettre au système de couvrir un grand champ

2. http ://wiki.dbpedia.org/
3. https ://www.freebase.com/

thématique. En ce sens, ces systèmes mettent en œuvre des techniques se rapprochant fortement de celles employées pour réaliser des systèmes de questions-réponses (Lopez et al., 2011) dans le domaine de la recherche d'informations. On peut également mentionner les systèmes de dialogue pouvant interagir avec plusieurs utilisateurs en même temps (Bohus et Horvitz, 2010). Dans ce manuscrit, nous nous limiterons cependant aux approches permettant à un système de réaliser un dialogue avec pour objectif principal de répondre à un besoin particulier de l'utilisateur dans un domaine applicatif bien identifié, on parlera alors de but utilisateur (*goal-oriented dialogue* en anglais). Procéder à la réservation d'un séjour dans un hôtel, consulter des horaires de trains, résoudre une panne, ou encore exécuter une commande de déplacement d'un objet dans un appartement sont autant d'exemples de buts utilisateur envisageables.

Toutes les interactions considérées dans ce manuscrit sont supposées pouvoir être modélisées par une alternance de « tours » entre les deux participants, c'est la notion de **cycle du dialogue**. Un tour représente le laps de temps pendant lequel un des participants peut s'adresser à l'autre pour faire progresser le dialogue (fin d'un tour généralement manifesté par l'apparition d'un silence prolongé). Le cycle du dialogue peut s'illustrer ainsi : lorsque c'est son tour, un des participants prend la parole (ou plus largement réalise une action) pendant que l'autre est à son écoute. Ce dernier, une fois la fin dudit tour atteinte, donne un sens aux dires ainsi perçus, prend une décision sur la suite à donner à l'interaction puis transmet sa réponse à l'autre participant. Le même processus d'écoute, interprétation, décision et génération de la prochaine action s'opère chez l'autre participant, et le cycle se poursuit jusqu'à la fin du dialogue (résolution ou échec du but utilisateur marqué par une rupture du cycle d'actions communicatives, par exemple raccrocher un téléphone ou s'éloigner).

Il est à noter que certains travaux, comme ceux présentés dans (Skantze et Schlangen, 2009; Khouzaimi et al., 2014), ont une vision plus souple de ce cycle, notamment par l'adoption de mécanismes dits incrémentaux pour le traitement des entrées/sorties du système. Ce type de technique peut par exemple être employé pour permettre au système de générer certaines actions pendant le tour de l'utilisateur afin de lui manifester l'état de sa compréhension courante (qu'elle soit bonne ou mauvaise). Selon la nature du système (service vocal, robot, etc.), cela peut prendre la forme d'une production de gestes ou de phénomènes vocaux d'acquiescements (hochement de tête, acte verbal comme « hum hum »). De la même façon, il est possible sous ce paradigme d'envisager la génération de tentatives de prise de parole de la part de la machine. Pour ce faire, ces approches adoptent généralement une vision plus atomique du tour de parole avec l'introduction de la notion de « micro-tour ».

Comme il en est fait mention dans la description de la notion de cycle du dialogue donnée plus haut, un système de dialogue doit être capable de :

— capturer et donner un sens au tour utilisateur ;
— prendre une décision pour poursuivre l'interaction ;
— restituer cette décision à l'utilisateur pour qu'il en fasse à son tour une bonne interprétation.

Dans la littérature, l'architecture des systèmes de dialogue généralement retenue pour répondre à la problématique posée comprend plusieurs composants. Avant d'al-

ler plus loin dans la description de ces derniers, il convient au préalable de faire la distinction entre les **systèmes vocaux** et les **systèmes multimodaux**. Les premiers font l'usage de la parole en tant que moyen exclusif de communication entre l'Homme et la Machine. On emploiera le terme de « modalité » dans la suite de ce manuscrit pour référer à un moyen de communication particulier. Les seconds font quant à eux également intervenir une à plusieurs autres modalités supplémentaires (gestes, saisie de texte, etc.). Dans ce manuscrit, nous parlerons également dans le chapitre 6 d'une troisième catégorie de systèmes, plus transverse, en liens avec les travaux réalisés dans le cadre du projet *MaRDi* auquel nous avons contribué, à savoir celle des **systèmes situés**.

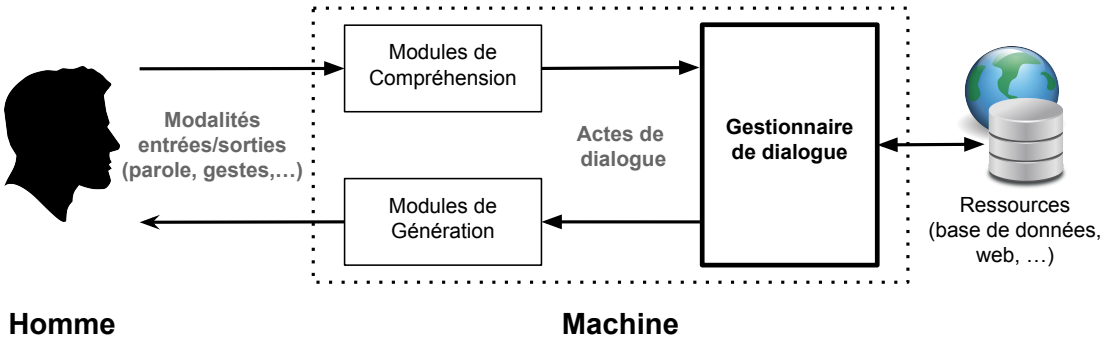


FIGURE 2.1 – *Architecture haut niveau d'un système de dialogue.*

La figure 2.1 donne un représentation « haut niveau » de l'architecture employée pour répondre à la problématique du dialogue (qu'il soit unimodal ou multimodal). Comme il est possible de le constater, un système de dialogue résulte du chaînage de divers modules, chaque module étant interfacé au suivant par la transmission de ses résultats.

Un premier ensemble de modules (ici représentés par l'encadré « Modules de Compréhension ») a pour rôle d'extraire le sens (la sémantique) des informations effectivement transmises par l'utilisateur au système par l'intermédiaire des différentes modalités considérées. Une représentation symbolique est alors nécessaire pour formaliser le sens d'un énoncé (ou plus largement de la combinaison des modalités dont l'utilisateur a fait l'usage) pour qu'il puisse être manipulé par le système. Dans nos études nous utilisons exclusivement le formalisme des actes de dialogue que nous détaillons dans la section 2.1.1.

Le gestionnaire de dialogue (*Dialogue Manager* - DM) est au cœur du système. Il a pour rôle principal de fournir le contenu de la réponse du système, là encore sous forme d'actes de dialogue. Pour ce faire, il met à jour son **état interne** grâce notamment au contenu sémantique du dernier énoncé utilisateur obtenu en sortie des modules de compréhension. Cet état contient les variables nécessaires au suivi de l'avancement de la tâche de dialogue (historique des tours précédents) ainsi que des informations permettant d'identifier les attentes de l'utilisateur dont une représentation formelle et structurée est faite au travers de la définition d'une **ontologie du domaine** (voir Annexe C). C'est également à ce niveau qu'un accès à des sources de connaissances est

réalisé dans le but d'obtenir des informations utiles à la résolution du but utilisateur. Cela se traduit généralement dans la pratique par l'interrogation d'une base de données (ou plus généralement du web). C'est donc sur la base de cet état mis à jour et enrichi par les données ainsi collectées que le système détermine la prochaine action à réaliser. Cette dernière est déterminée par la **politique** d'interaction employée par le DM. Nous reviendrons tout particulièrement sur le problème de la gestion de l'interaction dans la section 2.2.

L'action prise par le système est ensuite traduite dans une forme compréhensible pour l'utilisateur en exploitant pour cela les différentes modalités. Cette restitution est obtenue par l'intermédiaire d'un ensemble de modules (ici représenté par l'encadré « Modules de génération »). Cette étape est également importante car si les sorties effectives du système sont de mauvaises qualités, elles peuvent introduire de la confusion, une perte en efficacité ou pire, faire penser à l'utilisateur que le système n'est pas fonctionnel.

Étant donné que le DM raisonne à un niveau sémantique qui peut être considéré comme plus ou moins indépendant des modalités d'entrées/sorties choisies, les principales différences qui existent entre les systèmes purement vocaux et ceux multimodaux reposent principalement sur le choix des modules employés pour effectuer les étapes de compréhension et de génération.

Nous avons fait le choix de présenter en tout premier lieu les composants mis en œuvre dans un système de dialogue purement vocal. Ce choix s'explique principalement par le fait que la parole est la modalité de communication dominante dans l'ensemble de nos travaux, mais également par le fait qu'il s'agit d'une des formes de système de dialogue dont l'architecture est la plus « standardisée » dans la littérature. Enfin, nous étendrons cette description au cas plus complexe d'une architecture d'un système multimodal. Pour ce faire, nous nous concentrerons essentiellement sur la description des mécanismes de **fusion** et de **fission** employés pour la gestion des entrées et des sorties multimodales considérées (mécanismes communs à de nombreuses variantes d'un tel système). Comme nous l'avons déjà mentionné, nous détaillons la problématique centrale de la gestion de l'interaction dans la section 2.2.

2.1.1 Composants d'un système de dialogue vocal

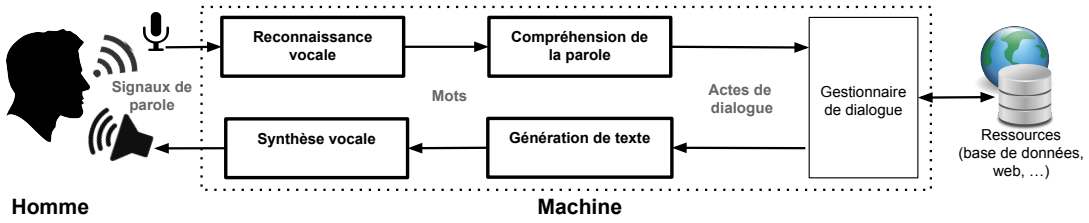


FIGURE 2.2 – *Architecture classique d'un système de dialogue oral*

La figure 2.2 représente les modules fonctionnels mis en œuvre dans un système

de dialogue oral. Cette architecture est employée dans la plupart des travaux sur les systèmes de dialogue parlé (*Spoken Dialogue System* - SDS) (Levin et al., 1997; Young et al., 2010; Thomson et Young, 2010).

Dans ce mode de fonctionnement, la tâche de compréhension de l'énoncé utilisateur repose sur l'utilisation consécutive d'un module de reconnaissance automatique de la parole (*Automatic Speech Recognition* - ASR), dont l'objectif est de transcrire le signal de parole en texte, et d'un module de compréhension automatique de la parole (*Spoken Language Understanding* - SLU) afin d'en extraire les concepts sémantiques associés.

De façon symétrique, le processus de génération exploite un module de génération en langage naturel (*Natural Language Generation* - NLG), visant à « traduire » les concepts sémantiques émis par le système en texte, ainsi qu'un module de synthèse vocale (*Text To Speech* - TTS), qui a pour rôle de transformer le texte ainsi obtenu en signal de parole.

Nous allons décrire ci-dessous plus en détail les différents modules d'entrées et de sorties employés dans un SDS.

Reconnaissance automatique de la parole

L'ASR est le premier module de la chaîne de traitements employée dans un système de dialogue vocal. Son rôle est de convertir en des temps proches du temps réel le signal de parole en provenance de l'utilisateur en texte (plus exactement sous la forme d'hypothèses de transcription). Pour ce faire, cet outil s'appuie sur un ensemble de modèles probabilistes représentant les caractéristiques acoustiques, sémantiques et grammaticales de la langue cible.

Le problème de la transcription automatique de parole peut s'exprimer mathématiquement de la façon suivante :

Soit x l'observation (signal) et w_i une séquence de mots possibles. On cherche à déterminer la meilleure hypothèse de transcription w^* à partir d'une observation, ce qui revient à vouloir calculer :

$$w^* = \underset{i}{\operatorname{argmax}}\, p(w_i|x) \tag{2.1}$$

D'après le théorème de Bayes, on peut réécrire 2.1 :

$$w^* = \underset{i}{\operatorname{argmax}}\, \frac{p(x|w_i)p(w_i)}{p(x)} \tag{2.2}$$

Or, il est possible d'émettre l'hypothèse que la probabilité d'obtenir une observation $p(x)$ est la même quelle que soit l'observation x et on en déduit la formule suivante :

$$w^* = \underset{i}{\operatorname{argmax}}(p(x|w_i)p(w_i)) \tag{2.3}$$

La probabilité d'émission du signal x sachant la séquence de mots w_i, notée $p(x|w_i)$, est obtenue grâce à un modèle acoustique. Ce dernier représente les caractéristiques propres du signal de parole dans une langue donnée (généralement en faisant usage d'une unité de son élémentaire d'un point de vue phonétique tel que le phonème ou la syllabe). Pour ce faire, les modèles de Markov cachés (*Hidden Markov Models* - HMM) font figures de standard dans la littérature. Le lecteur peut notamment trouver dans (Gales et Young, 2008) un panorama plus complet de ce type d'approches. À ce jour les performances état de l'art sont obtenues grâce à l'utilisation conjointe d'HMM et d'approches à base de réseaux de neurones profonds, comme dans (Hinton et al., 2012; Deng et al., 2013).

La probabilité de la séquence de mots $p(w_i)$ est quant à elle obtenue par un modèle de langue. Ce dernier est employé par le système pour guider le décodage vers des hypothèses de phrases cohérentes syntaxiquement et/ou grammaticalement. La cohérence linguistique est une donnée fondamentale pour un ASR, par exemple dans (Mariani, 1990) des expériences réalisées sur la langue française ont montré que l'observation d'une séquence de 9 phonèmes peut engendrer pas moins de 32000 segmentations en mots différentes si aucune contrainte linguistique n'est considérée. Le modèle de langue est communément basé sur un modèle n-grammes (Shannon, 1951) et représente à lui seul les contraintes syntaxiques et grammaticales de la langue cible. Ce modèle attribue des probabilités fortes à des suites de mots beaucoup observées dans les données d'apprentissage employées pour en faire l'estimation et des probabilités faibles pour des séquences peu ou pas rencontrées.

Il est à noter que certains travaux, tels que ceux présentés dans (Graves et Jaitly, 2014), ont montré que l'utilisation de réseaux de neurones récurrents (*Recurrent Neural Network* - RNN) permet d'envisager la construction d'un ASR de bout en bout avec une méthode unifiée (se substituant ainsi à l'utilisation des deux modèles décris ci-dessus) à partir de signaux de parole brut et de leur transcription.

Malgré une amélioration drastique des performances de ces systèmes au cours de ces dernières années, un ASR demeure un outil faillible qui est sujet aux erreurs. Ainsi, dans les conditions d'utilisation que sont celles des systèmes de dialogue (parole spontanée, auto-correction de l'utilisation, utilisation de phrase agrammaticale, bruit ambiant, etc.) un tel système, même optimisé pour la tâche, fera en moyenne une faute tous les cinq mots, comme le montre la plupart des travaux de référence sur le domaine (Young et al., 2010; Thomson et Young, 2010; Gašić et al., 2011; Black et al., 2011). Pour faire face à ce problème, il convient d'exploiter au mieux l'ensemble des hypothèses de transcription émises par le système plutôt que de se fier uniquement à la meilleure d'entre elles.

Les sorties du module ASR peuvent être exploitées dans la chaîne du dialogue sous différentes formes. Parmi elles, la **liste des N-meilleures hypothèses** de transcription est sans doute le format le plus utilisé dans la littérature. Il s'agit d'une approximation de la distribution complète sur l'ensemble des phrases possibles en ne conservant que les N plus probables avec leur score de confiance. Un exemple avec $N = 3$ est donné dans la figure 2.3a. Il faudra généralement trouver un bon réglage de N pour obtenir des

hypothèses sémantiques suffisamment variées pour permettre au système d'être plus tolérant aux erreurs. On pourra également mentionner deux autres formes de sorties communément exploitées dans la littérature, à savoir les **treillis de mots** (Murveit et al., 1993) et les **réseaux de confusions** (Mangu et al., 2000). Elles présentent toutes deux une approximation de la distribution complète moins limitée que celle faite par les listes des N-meilleurs hypothèses (les mots de faibles probabilités y étant généralement conservés).

Un treillis de mots est un graphe dirigé qui représente une portion du graphe de mots qui a été effectivement développée lors du processus de décodage de l'ASR. En effet, il intègre généralement l'ensemble des chemins complets (du début à la fin du signal de parole) mis en concurrences par l'ASR. Les nœuds de ce graphe représentent différents indices temporels et les arcs les hypothèses en termes de mots entre deux pas de temps. Un exemple de treillis est donné dans la figure 2.3b. Pour des raisons de simplifications on utilise aussi souvent des graphes de mots, en faisant disparaitre du treillis le marquage temporel des nœuds.

Un réseau de confusions peut être vu comme une représentation compacte d'un treillis de mots. Il s'agit là encore d'un graphe dirigé mais présentant une topologie particulière. En effet, les nœuds correspondent cette fois à des intervalle de temps du treillis et les arcs sont associés à un mot et sa probabilité à posteriori. Entre deux nœuds les arcs forment un jeu d'hypothèses mutuellement exclusives dont la somme des probabilités vaut 1. La création d'un réseau de confusion peut nécessiter la création d'arcs avec comme sortie *!NULL*, utilisés pour représenter des transitions ne faisant intervenir aucun mot. Contrairement au treillis de mots, les réseaux de confusions ont la particularité d'autoriser la création de séquences additionnelles (non issues du décodage de l'ASR). Par exemple sur la figure 2.3c la séquence « moderately price » est possible dans le réseau bien qu'il n'existe pas de chemin la représentant dans le treillis (voir figure 2.3b).

Dans nos études nous ferons cependant exclusivement usage de la liste des N-meilleures hypothèses pour sa facilité d'usage et sa capacité à être produite en temps réel par la plupart des systèmes ASR.

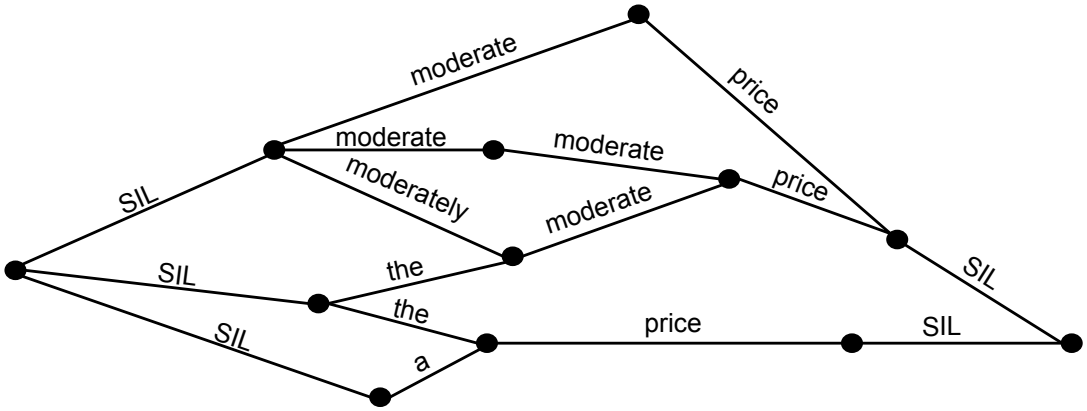
Compréhension automatique de la parole

Dans un système de dialogue Homme-Machine, le module de SLU joue un rôle intermédiaire entre le module ASR et le DM. Son rôle est d'extraire une représentation sémantique abstraite de l'entrée vocale de l'utilisateur à partir des hypothèses de transcription provenant de l'ASR, afin qu'elles puissent être traitées par le DM.

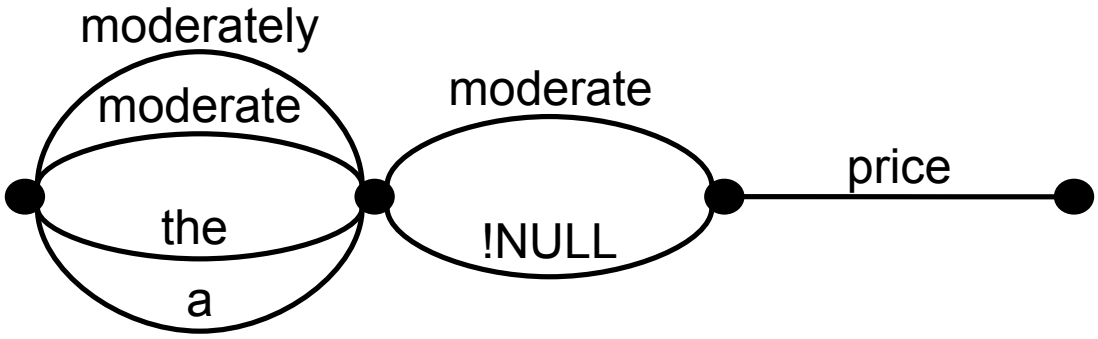
Représentation sémantique le choix d'une représentation sémantique assez riche pour capturer toute la richesse du langage humain constitue un réel défi scientifique en soit. En effet ce choix conditionne de façon importante les interprétations que le système pourra réaliser ainsi que sa capacité expressive (puisqu'un format identique est également employé pour les réponses du système). Généralement pour répondre à un besoin

Rang	Hypothèse	Probabilité logarithmique
1	moderate price	-1.3754
2	moderate moderate price	-1.3904
3	the price	-1.5640

(a) *Liste des 3-meilleures hypothèses.*



(b) *Treillis de mots (SIL correspondant au pause dans le signal de parole).*



(c) *Réseaux de confusions*

FIGURE 2.3 – *Exemples de liste des N-meilleures hypothèses (ici N=3), de treillis de mots et de réseaux de confusions sur la même phrase source.*

applicatif précis, des formes simplifiées, capturant uniquement l'information nécessaire à la conduite du dialogue, sont adoptées (De Mori et al., 2007). Le principal problème réside dans le fait que ces représentation sont sujettes à des variations d'une implémentation d'un système de dialogue à un autre. Cependant, beaucoup de travaux, notamment ceux utilisés dans ce manuscrit, ont en commun de placer leur représentation sémantique au niveau des **actes de dialogue** (Traum, 1999). Leur définition a pour origine la théorie des actes de langage établie par Searle (Searle, 1969) pour identifier le rôle fonctionnel d'un énoncé. Cette théorie a plus tard été étendue par Grice avec le concept d'intention du locuteur (Grice et al., 1975).

Dans la représentation adoptée dans nos travaux, à savoir celle proposée par le groupe CUED de l'université de Cambridge (Young, 2007), un acte de dialogue est défini comme étant la combinaison d'une étiquette identifiant l'intention dialogique portée par l'énoncé dont on veut extraire le sens (une demande d'information ou de confirmation, une affirmation d'un fait, etc.) et d'une séquence optionnelle d'arguments traduisant les informations sémantiques (concepts) également transmises par cet acte de dialogue :

$$acttype(\underbrace{a = x, b = y, ...}_{arguments})$$

acttype correspond au type de l'acte de dialogue (affirm, inform, confirm, request, etc.). Les arguments peuvent être des paires *concept=valeur* ou simplement des *concepts* et/ou *valeurs*. Une description plus détaillée de la taxonomie employée dans nos travaux est disponible dans l'annexe A de ce manuscrit.

Dans le contexte d'un système de dialogue ayant pour objectif de donner des informations sur les établissements d'une ville, les énoncés utilisateur suivants « je suis à la recherche d'un restaurant français » et « trouve moi un restaurant français » sont tous deux représentés par le même acte de dialogue « *inform(task=find,type=restaurant,food=french)* » [4]. Ce dernier se traduit par le fait que l'utilisateur cherche à informer le système (*inform*) qu'il est à la recherche (*task=find*) d'un restaurant (*type=restaurant*) français (*food=french*). De même, l'énoncé « je voudrais avoir le numéro de téléphone » est représenté par l'acte de dialogue « *request(phone)* » qui symbolise que l'utilisateur souhaite que le système lui transmette le numéro de téléphone.

Des représentations sémantiques plus riches peuvent être employées dans le cadre du dialogue. C'est notamment le cas des travaux présentés dans (Pinault et al., 2009; Pinault et Lefèvre, 2011a) qui, sur la base du paradigme FrameNet (Baker et al., 1998), utilisent une représentation hiérarchique s'appuyant sur la notion de *cadres sémantiques* (*frames* en anglais) pour représenter de façon plus structurée (graphes de frames) l'information sémantique contenue dans l'énoncé utilisateur (Meurs et al., 2009). Un exemple de cette représentation est donné sur la figure 2.1. Comme on peut le voir, il s'agit là d'une annotation sémantique selon deux niveaux. Le premier d'entre eux correspond à une modélisation « à plat » du sens énoncé par une séquence de triplets mode-concept-valeur. Le mode d'un concept peut être positif (+), négatif (-), interrogatif (?) ou option-

4. Il est à noter que la forme décomposée « *inform(task=find)* | *inform(type=restaurant)* | *inform(food=french)* » peut également être employée

nel (*). Le second niveau est celui des *frames*. Elles hiérarchisent ces triplets en groupe d'information plus haut niveaux (formulaires) en faisant le lien avec les informations en provenance de l'ontologie du domaine (date-début, date-fin, durée, etc.). Ces *frames* ont la particularité de pouvoir modéliser un niveau arbitraire d'abstraction dans leur capacité de pouvoir pointer vers d'autre *frames*. Par exemple, la frame *VOULOIR* pointe sur *RÉSERVATION* qui elle même pointe sur *HÔTEL* et *PÉRIODE*.

Au-délà des considérations liées à leurs qualités respectives de représentation, c'est principalement leur usage dans des corpus disponibles qui motive le choix entre ces alternatives. Par exemple, alors que des bases de données existent pour le français avec des annotations conforme à la représentation sémantique de type FrameNet (Lefèvre et al., 2012), nous allons utiliser principalement la première venant du CUED car elle permet la comparaison avec d'autres travaux en étant utilisée dans les corpus des challenges DSTC, présentés ci-après.

Transcription	je souhaiterais réserver un hôtel à Avignon pour 2 jours		
Concepts	**Nom du concept**	**Mode**	**Valeur**
	command-tâche	+	réservation
	objetBD	+	hôtel
	localisation-ville	+	Avignon
	séjour-nbNuit	+	2
Frames	**Nom de la *frame***	**Champs**	**Valeurs**
	F1 : *VOULOIR*	objet	F2
	F2 : *RÉSERVATION*	établissement période	F3 F4
	F3 : *HÔTEL*	ville nom	Avignon X
	F4 : *PÉRIODE*	date-début date-fin durée	X X 2 jours

TABLE 2.1 – *Exemple d'annotation en frames proposée dans (Meurs et al., 2009).*

Décodage sémantique Pour réaliser la tâche de décodage sémantique (extraction effective de la représentation abstraite adoptée sur les sorties ASR), des approches reposant sur la définition d'un ensemble de règles expertes ou de grammaires formelles plus ou moins complexes ont premièrement été adoptées dans de nombreux systèmes de dialogue (Young et Proctor, 1989; Ward, 1994; Wang et al., 2000). Bien que ces méthodes peuvent s'avérer efficaces en pratique, elles n'en demeurent pas moins très spécifiques au domaine applicatif pour lequel elles ont été développées initialement et sont souvent le fruit de couteuses phases d'itérations entre tests en déploiement et raffinements manuels pour faire face à l'ambiguïté liée aux spécificités de la parole (Young, 2002).

De nos jours, la majorité des méthodes état de l'art pour le SLU reposent sur des techniques issues de l'apprentissage automatique qui sont réputées plus flexibles et robustes aux aspects agrammaticaux du langage parlé ainsi que plus à même de faire

face aux erreurs faites par l'ASR. De plus, ces méthodes sont capables de fournir au DM une information plus riche en mettant à sa disposition des hypothèses scorées pertinentes (sous forme de listes de N-meilleures, treillis, etc.) qui permettront de transmettre les ambiguïtés au plus haut niveau décisionnel (si par exemple deux hypothèses ont des scores proches). Pourtant ces gains en termes de modélisation sont généralement obtenus au prix d'un besoin important en données annotées pour pouvoir réaliser un apprentissage de qualité. C'est pourquoi, les approches à base de règles représentent encore un intérêt dans des phases de prototypage où les données d'apprentissage viennent à manquer (Denecke, 2002). Nous discuterons tout particulièrement de cette problématique dans le chapitre 5.

Pour permettre l'application d'approches statistiques, la tâche de SLU est souvent assimilée dans la littérature à un problème d'étiquetage séquentiel de l'énoncé utilisateur mot à mot en exploitant des descripteurs extraits sur une fenêtre d'observation (et éventuellement sur la ou les décisions précédentes) dans le mécanisme de prise de décision. Il est à noter qu'une autre vision du problème consiste à raisonner sur des descripteurs extraits sur l'ensemble de l'énoncé puis de considérer l'utilisation de techniques capable de prédire plusieurs étiquettes sémantiques.

Le premier type de méthodes requiert généralement des données d'apprentissage alignées aux mots, ce qui n'est pas nécessaire pour le second type de méthode. La figure 2.4 donne un exemple pour illustrer la différence entre des données d'apprentissage alignées et non alignées. Nous profitons de cet exemple pour introduire le formalisme d'annotation segmentale BIO (pour Begin Inside Outside) (Ramshaw et Marcus, 1995). Dans ce dernier, les frontières entre les différentes étiquettes sont identifiées grâce à une convention d'annotation. Ainsi on fait précéder l'étiquette de **B-** pour identifier le premier mot qui lui est aligné ; pour tous les autres mots qui lui sont ensuite associés on fera précéder l'étiquette de **I-**. Les mots que ne sont associés à aucun concept particulier sont étiquetés par **O**.

Parmi les principales approches probabilistes employées dans la littérature, on pourra distinguer celles faisant appel à des modèles génératifs de celles utilisant des approches discriminantes.

Soit $x \in X$ et $y \in Y$ où X est l'espace des observations (descripteurs) et Y est celui des classes (étiquettes sémantiques) qu'il est possible d'associer aux observations. La distribution de probabilité conditionnelle, $p(y|x)$, est la distribution naturelle pour déterminer la bonne classe de sortie y sachant l'entrée x. Les approches qui visent à la modéliser directement sont dénommées les modèles discriminants dans la littérature. Les modèles génératifs s'attachent quant à eux à représenter la distribution de probabilité jointe, $p(x, y)$, qui grâce au théorème de Bayes peut être transformée en $p(y|x)$ pour effectuer la tâche de classification. Cependant la distribution $p(x, y)$ a l'avantage de pouvoir aussi être employée à d'autres fins, comme par exemple pour générer les couples (x, y) les plus probables.

Parmi les modèles génératifs on pourra faire mention des approches exploitant les réseaux bayésiens dynamiques et plus particulièrement celles adoptant les HMM. La première application de cadre formel au SLU a été proposée dans (Pieraccini et al.,

je cherche un restaurant français euh près du cinema
inform(task=find,type=restaurant,food=french,near=Cinema)

(a) *Annotation sémantique non alignée.*

je	B-inform(task=find)
cherche	I-inform(task=find)
un	B-inform(type=restaurant)
restaurant	I-inform(type=restaurant)
français	B-inform(food=french)
euh	O
près	B-infom(near=Cinema)
du	I-inform(near=Cinema)
cinéma	I-inform(near=Cinema)

(b) *Annotation sémantique alignée aux mots.*

FIGURE 2.4 – *Exemple d'annotation sémantique non alignée d'un énoncé utilisateur selon la représentation retenue dans nos travaux ainsi que sa version alignée aux mots.*

1992). Depuis, plusieurs travaux ont fait l'usage d'approches similaires comme ceux présentés dans (Pieraccini et Levin, 1995; Minker et al., 1996; Lefèvre et Bonneau-Maynard, 2002; Wang et al., 2005; He et Young, 2006; Lefèvre, 2007). On peut également mentionner des approches visant à assimiler le problème de compréhension à celui de la traduction en langage naturel (source) vers une autre « phrase » dans la représentation sémantique adoptée. Par exemple, dans (Macherey et al., 2009; Hahn et al., 2010) cet objectif est atteint en employant une approche standard de traduction automatique combinant un modèle de traduction automatique statistique à base de segments (*Phrase-Based Statistical Machine Translation* en anglais), un modèle de langage et un modèle de ré-ordonnancement.

Cependant, comme il a été montré dans (Wang et Acero, 2006), les approches discriminantes, de par leur plus grande flexibilité sur la prise en compte des observations, disposent d'un avantage applicatif notable sur les modèles génératifs pour la tâche SLU (meilleures performances). En effet, les approches discriminantes peuvent librement utiliser des fonctions arbitraires sur les descripteurs issus des observations, car elles n'ont pas besoin d'effectuer des hypothèses d'indépendances sur ces variables (contrairement aux approches génératives). De ce fait elles sont mieux équipées pour capturer les dépendances sur un énoncé lors du décodage sémantique. Ce constat a notamment été réitéré dans (Hahn et al., 2010).

Parmi les approches discriminantes état de l'art, on peut mentionner les approches reposant sur les champs conditionnels markoviens (*Conditional Random Fields* - CRF) (Lafferty et al., 2001) comme dans (Wang et Acero, 2006; Raymond et Riccardi, 2007) ou encore celles les utilisant conjointement avec des techniques issues de la traduction automatique à l'instar de (Jabaian et al., 2014) qui opère leur combinaison avec d'autres modèles par l'intermédiaire d'un transducteur à états finis (*Finite State Transducer* - FST). On peut également citer (Mairesse et al., 2009) dans lequel les auteurs font appel à

un ensemble de machines à vecteurs de support (*Support Vector Machine* - SVM) pour apprendre un SLU sur des données non alignées. De même, des techniques reposant sur des réseaux de neurones artificiels ont également été employées avec succès sur cette tâche. Parmi les architectures neuronales proposées, nous pouvons lister les réseaux profonds de croyance (*Deep Belief Network* - DBN) (Deoras et Sarikaya, 2013), les RNN (Yao et al., 2013; Mesnil et al., 2013) mais aussi les champs conditionnels markoviens récurrents (*Recurrent Conditional Random Fields* - R-CRF) (Yao et al., 2013, 2014) qui à l'instar des CRF introduisent une notion de séquentialité dans la fonction objectif des réseaux récurrents utilisés pour résoudre le problème de compréhension.

La génération en langue naturelle et synthèse vocale

Le module NLG a une fonction symétrique à celle du SLU. En effet son rôle est de transformer la réponse du DM qui est sous forme actes de dialogue en langue naturelle, c'est à dire dans une forme textuelle exploitable par le module de synthèse vocale. On parlera également de génération en langue naturelle pour faire référence à cette conversion. Il est à noter que là encore le formalisme de représentation du contenu sémantique des énoncés influe fortement sur les capacités de verbalisation de ce module.

Dans la plupart des travaux, et notamment ceux de référence en termes de système complet (Young et al., 2010; Thomson et Young, 2010), le processus de génération d'énoncés se limite souvent à l'application de patrons textuels prédéfinis. Quelques exemples de tels patrons sont regroupés dans le tableau 2.2. Bien que leur utilisation représente une solution relativement efficace dans les cas où le nombre d'actes de dialogue système considérés ne rend pas leur définition impossibles, des méthodes plus sophistiquées ont été étudiées dans la littérature pour mieux répondre à cette problématique (Walker et al., 2007; Mairesse et al., 2010; Mairesse et Young, 2014; Wen et al., 2015).

L'objectif visé par ces approches est d'utiliser des méthodes issues de l'apprentissage automatique, telles que les modèles de langage factorisés (*Factored Language Models* - FLM) ou les RNN, pour faciliter la création et la maintenance de tels systèmes. Cela peut se traduire par le fait d'introduire de la variabilité expressive dans les productions du système, ou encore d'offrir une plus grande tolérance à la nouveauté (séquences de concepts jamais rencontrées) ce qui permet notamment au système de dialogue d'être moins contraint sur la nature de ses sorties. De plus, l'emploi de telles techniques peut sensiblement améliorer les capacités de transfert d'un tel module (application sur une nouvelle tâche).

Enfin, certains travaux ont envisagé l'utilisation d'un cadre d'optimisation commun entre la gestion du dialogue et le NLG et ce par le biais de l'apprentissage par renforcement (*Reinforcement Learning* - RL) (Rieser et Lemon, 2010; Rieser et al., 2014; Lemon, 2011). Ceci est rendu possible par la modélisation du problème de génération par un processus de décision de Markov (*Markov Decision Process* - MDP) où l'agent apprenant réalise des choix de présentation de l'information sémantique haut niveau que souhaite transmettre le DM (modalités mises en jeu dans la restitution, façon de s'exprimer).

Actes de dialogue	Patrons associés
repeat()	Pouvez-vous répéter s'il vous plaît ?
request(area)	Pouvez-vous me donner une indication sur la zone de la ville qui vous intéresse ?
inform(phone=X)	Le numéro de téléphone est le X
confreq(type=bar,area=X,pricerange)	Ok un bar dans la partie X de la ville, mais dans quelle gamme de prix ?

TABLE 2.2 – *Exemples de patrons utilisés pour la génération. X représente ici une variable qui peut être remplacée par toutes les valeurs du concept auquel elle est associée dans le patron considéré*

La restitution vocale (synthèse vocale) de la réponse du système peut se faire de plusieurs manières. Tout d'abord il est possible d'envisager la sélection et la concaténation de segments élémentaires de parole naturelle pré-enregistrés. Ainsi on pourra mentionner les approches par sélection d'unités ou par corpus (Hunt et Black, 1996; Möbius, 2000). La synthèse vocale peut également se faire sur la base de modèles statistiques analogues à ceux utilisés pour l'ASR (qui est son processus symétrique dans la chaîne). Par exemple, les modèles HMM (Taylor, 2009) ou plus récemment les techniques à base de réseaux de neurones artificiels (Ling et al., 2015).

Certains systèmes de TTS permettent également de pouvoir spécifier des comportements vocaux (changements de rythme, de tonalité) pour jouer un ensemble (souvent réduit) d'émotions ou d'attitudes lors de la prononciation effective d'une phrase (Astrinaki et al., 2012). L'exploitation de réglages fins comme ceux visant à faire varier l'émotivité du timbre de voix (triste, heureux), le style de voix (portée, chuchotée) ou encore la vitesse d'élocution au cours de l'interaction présente un enjeu de taille pour tendre vers plus de naturel dans le dialogue. Nous verrons plus en détail l'intérêt de l'application de ce type d'approches dans le cadre du dialogue situé au chapitre 6.

2.1.2 Gestion de modalités multiples

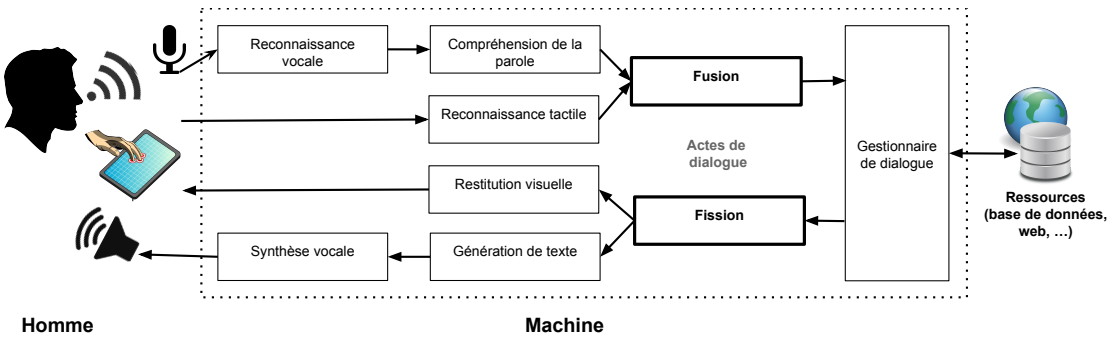


FIGURE 2.5 – *Architecture classique d'un système de dialogue multimodal.*

Plus généralement, le dialogue Homme-Machine peut faire usage d'une combinai-

son de modalités autres que celle de la parole. Par exemple, l'interprétation de gestes communicatifs émis par l'homme, l'utilisation d'entrées provenant d'un écran tactile ou encore une restitution visuelle sur un écran. Un système capable de gérer de multiples modalités entre dans la catégorie des systèmes de dialogue multimodaux (*Multimodal Dialogue System* - MDS) (Gibbon et al., 2001).

Un des tout premiers exemples d'un tel système est « *Put that there* » (Bolt, 1980). Dans ce dernier, les énoncés vocaux de l'utilisateur sont couplés à la position d'un curseur sur une interface graphique projetée sur un écran mural, déterminée via un capteur 3D interprétant ses gestes déictiques, pour réaliser des commandes utilisateur. Après cette tentative, de nombreux autres systèmes employant d'autres modalités ont été développés. On peut notamment lister (de façon non exhaustive) celui proposé dans (Koons et al., 1993) liant la parole, les regards et les gestes déictiques pour identifier des objets dans un environnement virtuel 3D ; MATIS (Nigay et Coutaz, 1995) proposant une interface multimodale liant la parole et une interface graphique usuelle (clavier/souris) pour effectuer une recherche d'information relative au trafic aérien ; MATCH (Johnston et al., 2002) offrant une interface multimodale parole/stylo sur un assistant numérique personnel donnant l'accès à des informations sur les restaurants et métros de la ville de New York ; SAMMIE (Becker et al., 2006) s'attachant à rendre un lecteur MP3, dont un affichage graphique est intégré dans le tableau de bord d'une voiture, accessible également par la voix ; ou encore le projet SmartKom (Wahlster, 2006) qui adresse une large gamme de tâches collaboratives sur des scénarios mettant des pièces intelligentes (smart room), des kiosques et ainsi des interfaces personnalisées aux utilisateurs combinant parole et des gestes. Concernant la robotique domestique, des interfaces similaires ont été développées dans (Stiefelhagen et al., 2004; Lucignano et al., 2013).

La figure 2.5 donne un exemple de ce type d'architecture, à savoir un système de dialogue multimodal faisant intervenir les modules en charge de la compréhension de la parole (ASR et SLU) et celui en charge de la reconnaissance tactile exploitant des données issues des capteurs présents sur un écran. On suppose également une restitution à la fois vocale et visuelle de la part du système. Selon le contexte et la nature de l'interaction, une utilisation différente des modalités peut être observée. Par exemple, dans un environnement bruyant il peut être préférable de favoriser temporairement les gestes à la parole.

Sachant que les modules employés en tant qu'entrées et sorties du système dépendent fortement de la tâche visée, nous consacrerons la suite de cette partie à la description des modules de fusion et de fission, qui sont les deux modules essentiels pour repositionner un MDS dans le paradigme du tour de dialogue. Il est à noter que nous donnerons dans le chapitre 6 un exemple concret d'un tel système.

Fusion

De nombreux canaux de communication multimodaux peuvent être employés par l'utilisateur (de façon intentionnelle ou non) pour permettre au système de dialogue

d'interpréter ses intentions. Dans la littérature on peut trouver de nombreuses combinaisons de modalités différentes, par exemple des situations où la reconnaissance vocale est assistée par un système de lecture labiale (Duchnowski et al., 1994), ou encore lorsqu'elle est employée avec un système de reconnaissance de gestes et d'écriture au stylo sur une interface graphique (Johnston et al., 2002).

Utilisation des modalités / **Type de fusion**	Séquentielle	Parallèle
Combinée	Alterné	Synergique
Indépendante	Exclusif	Concurrent

TABLE 2.3 – *Classification des mécanismes de gestion des entrées multimodales (Nigay et Coutaz, 1993).*

D'après la classification proposée dans (Nigay et Coutaz, 1993) et reportée dans le tableau 2.3), il est possible de distinguer les mécanismes de gestion des entrées multimodales d'une interface multimodale selon deux dimensions. La première colonne du tableau 2.3 représente la façon dont les modalités peuvent être utilisées par l' utilisateur du système. On distingue le mode séquentiel du mode parallèle où l'utilisateur peut employer plusieurs modalités simultanément. La seconde dimension (ligne) représente le fait que les informations fournies par plusieurs modalités peuvent être combinées ou considérées indépendantes et gardées en l'état.

Selon cette même classification il est donc possible d'identifier quatre modes :

— Alterné représente une situation où la fusion s'opère sur des données issues de plusieurs capteurs pour construire une sortie commune mais avec la contrainte que ces modalités soient utilisées de façon séquentielle. Par exemple une phrase utilisateur « ici » suivie d'un geste déictique après coup ;

— Synergique est analogue à la catégorie précédente, à la différence près que les modalités peuvent ici être employées simultanément par l'utilisateur ;

— Exclusif, dans ce mode l'utilisateur a le choix entre de multiples modalités, mais une seule peut être retenue à chaque tour pour transmettre son intention au système ;

— Concurrent, ce mode représente un contexte où l'utilisateur souhaite transmettre plusieurs intentions indépendantes au travers de plusieurs modalités. Par exemple sur une tâche de recherche d'information touristique avec carte interactive l'utilisateur peut demander oralement au système « peux-tu m'indiquer où se trouve le cinéma sur la carte ? » et cliquer simultanément sur l'icône d'un établissement pour avoir accès à des informations détaillées comme le numéro de téléphone.

Dans nos travaux nous nous concentrons tout particulièrement sur le mode synergique. Dans (Nigay et Coutaz, 1993), les auteurs mentionnent également une autre dimension, il s'agit du niveau de fusion que nous décrivons plus bas.

Dans le domaine de l'interaction homme-robot (*Human-Robot Interaction* - HRI) que

vise cette thèse, la prise en compte d'entrées multimodales et leur interprétation au niveau sémantique constituent un vecteur d'amélioration pour ce qui est de rendre toujours plus naturelle et qualitative la communication entre l'Homme et le Robot. Dans ce sens, un certain nombre de travaux se sont intéressés à la problématique de la fusion multimodale dans ce contexte (Holzapfel et al., 2004; Rossi et al., 2013), et s'intéressent notamment à la combinaison de la parole avec des gestes expressifs de l'utilisateur (déictiques, postures, etc.) pour interpréter au mieux des commandes (résolution des références).

De façon générale, le rôle du module de fusion est de combiner les informations multimodales au travers d'une représentation englobante, généralement sémantique (actes de dialogue), qui sera ensuite transmise au DM. Pour cela, il tient compte d'un certain nombre de facteurs propres aux diverses modalités à fusionner. Par exemple :

— Les entrées multimodales sont-elles complémentaires, corrélées ou indépendantes ?
— Les représentations employées (formats) sont elles compatibles ?
— Ces entrées doivent/peuvent-elles être alignées temporellement ?
— Est-il nécessaire d'effectuer des traitements sur une (ou plusieurs) d'entre elles ?
— Quel niveau de confiance leur accorder (qualité des capteurs employés, etc.) ?

Le module de Fusion pourra également s'appuyer sur les informations extraites du contexte de l'interaction pour par exemple désambiguïser au mieux une situation. Il est évident que l'ensemble de ces facteurs influencent directement sur le choix du mécanisme de fusion. Pour le concepteur ce choix se portera essentiellement sur deux critères : le niveau et la méthode de fusion. Même si nous détaillons ces notions ci-dessous, le lecteur pourra se référer à (Lalanne et al., 2009; Dumas et al., 2009; Atrey et al., 2010) pour plus de détails sur le sujet.

Choix du niveau de fusion on distingue généralement deux niveaux de fusion dans la littérature, la **fusion sur descripteurs** (également appelée fusion précoce) et la **fusion décisionnelle** (ou fusion tardive).

Le premier niveau se réfère aux techniques visant à fusionner les informations directement à la sortie des capteurs, et ce avant tout traitement. En d'autres termes, la fusion se fait au niveau des descripteurs bruts (tels qu'extraits par les capteurs multimodaux). Elle suppose généralement une très forte corrélation entre les données ainsi fusionnées tout comme des représentations initiales proches. Un bon exemple de ce type de fusion est celui mis en œuvre pour améliorer la qualité de la transcription vocale automatique par l'intermédiaire d'un système de lecture labiale (Duchnowski et al., 1994).

Le seconde consiste à considérer la fusion des informations unimodales uniquement après leur passage respectif dans une chaîne de post-traitements dédiée. Cette méthode permet une plus grande flexibilité dans le choix des capteurs employés du fait que la fusion s'opère cette fois sur la base d'informations dans une forme de représentation haut niveau, moins dépendante de l'implémentation des différents capteurs et de la nature des signaux considérés. Cependant, le passage à cette couche d'abstraction supplémentaire a un coût. En effet il peut introduire du bruit supplémentaire pouvant impacter sensiblement le résultat de la fusion (Atrey et al., 2010).

Afin de pouvoir contraster leurs aspects négatifs soulevés précédemment, il est également possible d'envisager des solutions hybrides, ces dernières proposant d'avoir recours aux deux types de fusion à différents niveaux de l'architecture (Wu et al., 2005).

Comme nous le verrons dans le chapitre 6, nous nous sommes particulièrement intéressés aux mécanismes de fusion au niveau décisionnel dans nos travaux relatifs à la plateforme de dialogue *MaRDi*. Dans notre cas, la parole est considérée comme la modalité principale, l'utilisateur peut au travers d'elle faire référence à des objets conceptuels qu'il faudra lier au niveau décisionnel à ses gestes déictiques ainsi qu'au contexte de l'interaction, acquis notamment grâce aux perceptions du robot.

Choix de la méthode de fusion trois grandes classes de méthodes peuvent être identifiées : les approches dites par **règles**, celles par **estimation** et enfin celles par **classification automatique**.

Les approches à base de règles regroupent une variété de techniques qui dépendent fortement des modalités considérées. On peut mentionner la fusion linéaire pondérée (Wang et al., 2003), le recours au vote à la majorité (Radová et Psutka, 1997), ou encore l'utilisation de règles expertes définies sur mesure pour la tâche comme dans (Pfleger, 2004). De façon générale ces différentes approches obtiennent de bonnes performances mais au prix d'un processus couteux d'établissement de règles efficaces qui reposent sur une bonne connaissance du domaine applicatif visé (difficulté de transfert).

Les approches dites par estimation, comme celles faisant usage des filtres de Kalman (Strobel et al., 2001) ou des filtres à particules (Pérez et al., 2004). Ces méthodes sont particulièrement adaptées dans un contexte de fusion précoce pour effectuer des tâches de suivi de l'état d'un objet en mouvement à partir de données multimodales (par exemple audio et vidéo).

Enfin, les approches dites par classification automatique regroupent quant à elles des approches allant de l'application de modèles génératifs comme dans (Pavlovic et Huang, 1999) où les auteurs font l'usage de réseaux bayésien dynamiques, à l'emploi d'approches discriminantes telles que les SVM comme dans (Rossi et al., 2013) ou les réseaux de neurones dans (Zou et Bhanu, 2005). On peut également faire entrer dans cette catégorie les techniques employant la logique floue à l'instar des travaux présentés dans (Reddy et Basir, 2010) qui utilisent dans leur mécanisme de fusion un modèle des croyances transférables (*Transferable Belief Model* - TBM) (Smets et Kennes, 1994). Chaque module d'entrée est vu comme un capteur assignant des croyances (niveau de confiance) sur un jeu fini de concepts. Elle propose donc des mécanismes robustes permettant d'exploiter ces confiances afin de renforcer une hypothèse ou un groupe d'hypothèses de combinaison (souvent guidé par une ontologie du domaine) lors du mécanisme de fusion. Ainsi, la notion d'incertitude est directement intégrée dans sa représentation, ce qui peut constituer un avantage par rapport au cadre probabiliste classique. Les approches par classification sont généralement employées sur des tâches de fusion où le système doit déterminer la nature d'une sortie parmi un jeu fixe de classes en fonction des observations multimodales.

Dans le domaine du dialogue multimodal et de l'interaction Homme-Robot cependant, la plupart des travaux utilisent un jeu de règles expertes spécifiques à la tâche et considèrent une des modalités comme étant dominante, généralement la parole. Sur cette ligne, on pourra citer les travaux de (Holzapfel et al., 2004; Pfleger, 2004; Corradini et al., 2005). Même si dans cette thèse une solution de cet ordre a été retenue pour les tests préliminaires de la tâche *MaRDi* (voir chapitre 6), il est clair que ses limitations théoriques et pratiques constituent pour nous un obstacle à son maintien dans la plateforme finale de dialogue. Les proposition faites dans (Rossi et al., 2013) et (Reddy et Basir, 2010) ont notamment particulièrement retenues notre attention. Cependant, il sera nécessaire au préalable de réaliser un important effort d'annotation sur les données collectées pour en garantir un niveau de performance acceptable.

Quel que soit le type de la méthode choisie, le temps est un élément important à prendre en considération dès lors que les différentes modalités employées peuvent être perçues de façon non-synchrones par le système. Une représentation du temps dans le mécanisme de fusion, autant dans sa dimension quantitative (observation sur une période temporelle) que qualitative (ordre des évènements) est généralement de rigueur.

De même, un autre aspect important du mécanisme de fusion est la façon dont est traitée l'ambiguïté. En effet, lorsque plusieurs entrées doivent être combinées plusieurs situations peuvent être identifiées :

— **renforcement** : les informations relatives aux deux modalités peuvent venir renforcer la même hypothèse. Par exemple, l'utilisateur dit « prends la tasse bleue » en désignant avec un geste de pointage non ambigu une tasse bleue posée devant lui ;
— **complémentarité** : les informations relatives aux deux modalités sont complémentaires. Par exemple, l'utilisateur dit « prends ça » tout en désignant du doigt une tasse bleue ;
— **incompatibilité** : les informations relatives aux deux modalités se contredisent ou ne se combinent pas. Par exemple, l'utilisateur dit « prends la tasse bleue » mais semble désigner un livre bleu du doigt. Dans ce cas, une estimation du niveau confiance (ou d'importance) des différents capteurs peut être très appréciable.

Lorsqu'il se place au niveau décisionnel le système doit parfois manipuler des entrées de natures différentes. Par exemple, des données probabilistes issues de la chaîne de compréhension de la parole (liste des N-meilleures hypothèses de compréhension) et déterministe (clic sur un bouton d'une interface graphique). Dans notre étude nous nous intéressons à un système capable de conserver un aspect probabiliste sur les sorties de la fusion avec la production de listes d'hypothèses pour gérer au mieux l'incertitude au niveau décisionnel.

Fission

Le module de Fission est responsable de la traduction d'un acte de dialogue système (abstrait) en une combinaison d'actions sur les différentes modalités et éventuellement

de la synchronisation de leur exécution. La tâche de fission est généralement décomposable en trois sous-tâches (Dumas et al., 2009) :

— Sélection et structuration du contenu à transmettre à l'utilisateur ;
— Sélection des modalités. Ces dernières sont généralement sélectionnées sur la base du contexte de l'interaction et du type d'information à transmettre. Il est également possible de tenir compte du profil utilisateur ;
— Coordination/synchronisation de l'exécution des sorties du système sur chaque modalités pour une présentation de l'action système cohérente.

Les approches déterministes sont encore à ce jour beaucoup employées pour mettre en œuvre ces mécanismes dans le cadre d'un système dialogue multimodal, notamment les approches par plan (Wahlster, 2002; Becker et al., 2006). Il est à noter que la nature même de l'action prise par le système peut sensiblement faciliter cette procédure (particulièrement en ce qui concerne les deux premières sous-tâches). En effet, l'action peut être enrichie avec des étiquettes visant à contrôler en partie la présentation multimodale. Là encore de telles stratégies de présentation multimodale peuvent être apprises sur des dialogues annotés (Rieser et Lemon, 2008) ou comme proposé dans (Lemon, 2011) de façon conjointe à la stratégie du dialogue grâce à sa modélisation par un MDP. Dans les travaux présentés au chapitre 6 nous avons fait usage d'une approche simple par règles pour réaliser cette tâche.

2.2 La gestion de l'interaction

Le DM est le cœur du système. Son rôle consiste à prendre les décisions (actions) pour faire progresser l'interaction jusqu'à son but (satisfaction utilisateur) en se basant sur le flux d'informations récupéré jusqu'alors. Pour ce faire, le DM maintient un état du dialogue qu'il connecte avec des sources extérieures (base de données, web, planificateur du Robot, etc.) pour déterminer la réponse adéquate. La performance du DM dépend donc d'une part de sa capacité à modéliser un état de dialogue cohérent et assez riche pour contenir l'information utile à sa prise de décision et d'autre part de la qualité de sa stratégie de choix d'action (politique) parmi l'ensemble de celles possibles.

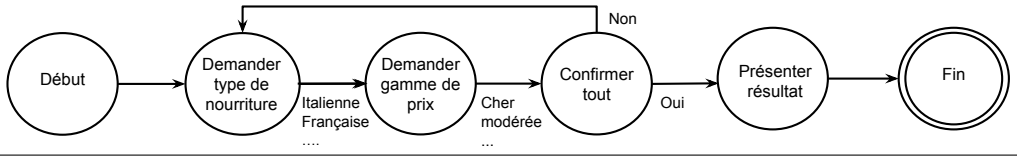
Plusieurs approches peuvent être utilisées pour l'implémentation d'un tel composant. On en distinguera principalement deux catégories. La première est celle des systèmes s'appuyant sur des approches déterministes généralement basées sur la définition d'un ensemble de règles. Ces dernières sont majoritairement privilégiées dans le cadre des systèmes industriels car malgré leur rigidité elles font généralement appel à des modèles simples dont le fonctionnement reste prédictible et améliorable par itérations. L'autre catégorie est celle des approches probabilistes. Ces dernières font l'objet d'études poussées dans la communauté scientifique pour leur capacité à mieux gérer l'incertitude et à apprendre par l'expérience des stratégies optimisées.

Dans la section suivante nous détaillerons donc ces deux catégories d'approches.

2.2.1 Approches déterministes

Les principaux paradigmes

Gestion par graphe le rôle principal du DM étant de décider quelle action le système doit prendre en chaque point du dialogue, une des approches les plus « simples » consiste à laisser le concepteur définir entièrement la structure de l'interaction sous la forme d'un graphe dirigé (Green, 1986; Sutton et al., 1996; McTear, 1998). Ce dernier représente la séquence des tours de parole système mais aussi les réponses possibles de la part de l'utilisateur à chaque point de l'interaction.



Sys_1	Bonjour et bienvenue sur le système *RestInfo*. Vous voulez que je cherche un restaurant servant quel type de nourriture ?
Usr_1	Française
Sys_2	Dans quelle gamme de prix ?
Usr_2	Bon marché
Sys_3	Je récapitule vous êtes à la recherche d'un restaurant français dans la gamme de prix bon marché, c'est bien ça ?
Usr_3	Oui
Sys_4	Le « Bistrot du coin » avenue Foch correspond à votre demande...

TABLE 2.4 – *Exemple simplifié de gestion du dialogue par un graphe sur une tâche de recherche d'information sur des restaurants.*

Un exemple d'une telle gestion est donné dans le tableau 2.4. Les nœuds du graphe représentent ici l'ensemble des états dans lesquels peut se trouver le système. Ces états sont associés à des actions systèmes (par exemple, poser une question ou accéder aux informations d'une base de données). Les arcs, eux, représentent les transitions qui existent (du point de vue du concepteur) entre ces états. Ces dernières sont conditionnées par des évènements tels qu'une réponse particulière de la part de l'utilisateur à une question système.

En suivant ce paradigme, le DM dispose d'un contrôle total sur le cours de l'interaction, le plus souvent en posant des questions dirigées à l'utilisateur pour limiter le champ de ses réponses. De ce fait, ce type d' approches est communément dénommé comme étant « à initiative du système » [5].

5. La notion d'initiative est liée au degré de liberté qui est laissé à celui qui va s'exprimer par celui qui vient de s'exprimer. On dit qu'un participant a l'initiative quand celui-ci guide l'interaction, par exemple en posant des questions précises. On parle d'initiative mixte quand les deux participants se laissent la possibilité de prendre temporairement l'initiative au cours du même dialogue.

Dans l'industrie, les systèmes déployés sont généralement basés sur des approches similaires et reposent sur des langages et *frameworks* de spécification haut niveau comme le standard W3C VoiceXML [6], ou encore l'AIML (Wallace, 2003).

Le principal inconvénient de ce mode de gestion réside dans sa rigidité qui rend le déroulement du dialogue peu naturel. En effet, dans ce paradigme les capacités de compréhension du système sont souvent réduites à celles requises dans l'état courant de l'interaction (pour effectuer une transition). De plus, dans ce mode de fonctionnement l'utilisateur ne peut pas donner les informations au système dans l'ordre qu'il souhaite ou encore poser des questions au système et prendre ainsi l'initiative.

Gestion par formulaire le paradigme dit du remplissage de formulaire (Ward et Issar, 1994; Goddeau et al., 1996) offre une vision plus flexible du problème. Dans ce paradigme, le système modélise au travers d'un formulaire les éléments informatifs utiles à la résolution de la tâche de dialogue. Ces derniers comprennent des informations que l'utilisateur peut vouloir exprimer ou demander au système (et inversement). Pour référer à ces éléments, on distinguera le champ (nom du concept) de sa valeur (information renseignant ce concept). Un exemple de formulaire est donné dans le tableau 2.5.

La gestion du dialogue consiste alors à gérer l'état de remplissage du formulaire avec les informations transmises par l'utilisateur mais aussi à exploiter son taux de remplissage (champs non spécifiés, etc.) et son rapport avec des données du domaine (base de données, etc.) pour déterminer la prochaine action système.

	Formulaire		
	Champs	**Valeurs**	
	Type de nourriture	française	
	Gamme de prix	bon marché	
Sys_1	Bonjour et bienvenue sur le système *RestInfo*. Comment puis-je vous aider ?		
Usr_2	Je cherche un restaurant français bon marché		
Sys_2	Confirmez-vous que vous êtes à la recherche d'un restaurant français dans la gamme de prix bon marché ?		
Usr_3	Oui		
Sys_3	Le « Bistrot du coin » avenue Foch correspond à votre demande		

TABLE 2.5 – *Exemple simplifié de gestion par formulaire du dialogue sur une tâche de recherche d'information sur des restaurants.*

Cette approche offre plus de liberté à l'utilisateur car il peut spécifier sa requête librement (ordre des informations données au système non contraint) et peut même prendre l'initiative localement (par exemple en demandant au système la valeur d'un

6. http ://www.w3.org/TR/voicexml30/

champ). Il est à noter que cette technique peut être combinée avec des approches de gestion par graphe pour pouvoir traiter plusieurs tâches connexes, chacune pouvant être représentée par un formulaire.

Gestion par plans si les approches à base de formulaire sont particulièrement bien adaptées pour des tâches relatives à la recherche d'information, elles sont difficilement applicables en l'état à des domaines sortant de cette problématique. En effet, pour une tâche telle que l'apprentissage d'une langue étrangère où les deux participants doivent œuvrer de paire pour atteindre le but visé (dialogues collaboratifs), des approches à base de plans peuvent être préférées.

Inspirée notamment des travaux présentés dans (Cohen et Perrault, 1979; Perrault et Allen, 1980), ces approches proposent de modéliser l'état mental de l'utilisateur (généralement sous forme de buts et de croyances) afin que le DM essaye de reconstruire le plan (possiblement partagé) que souhaite poursuivre actuellement l'utilisateur (qui supposé être rationnel) pour être à même d'y répondre de façon adéquate.

Dans ce formalisme, un plan est constitué d'un ensemble d'actions (ici les actes de dialogue système). Chaque action est définie par : un **entête**, représentant le nom et les paramètres de l'action, des **pré-conditions**, qui doivent être remplies pour que l'opération soit applicable, un **corps**, qui représente la réalisation concrète de l'action (sous-actions), et des **effets** qui décrivent l'impact de l'action sur l'état du monde. Ainsi, un plan est défini comme étant une séquence bien formée d'actions de telle sorte que leurs effets soient également les pré-conditions des actions suivantes. Un exemple est donné dans le tableau 2.6.

Entête	*Inform(Speaker,Hearer,LocalTime)*
Pré-conditions	*Knows(Speaker,LocalTime)* *Wants(Speaker,Inform(Speaker,Hearer,LocalTime))*
Corps	*Believes(Hearer,Wants(Speaker,Knows(Hearer,LocalTime)))*
Effets	*Knows(Hearer,LocalTime)*

TABLE 2.6 – *Exemple de définition d'une action employée dans la gestion de l'interaction par plan pour donner à l'utilisateur de l'heure locale.*

Une fois les opérations identifiées, la conversation peut alors être guidée par un planificateur/raisonneur qui va devoir construire en fonction de l'état courant de l'interaction un plan valide pour atteindre le but utilisateur grâce aux opérations mises à sa disposition. Parmi les systèmes utilisant ce paradigme, nous pouvons citer les systèmes *TRAINS* (Allen et al., 1995, 1996), *RAILTEL* (Bennacef et al., 1996), *TRIPS* (Ferguson et al., 1998) ou encore *ARISE* (Lamel et al., 2000).

Gestion par agendas cette approche a également été étendue à des approches dites basées sur les agendas, comme celle proposée par le gestionnaire de dialogue *Ravenclaw* (Bohus et Rudnicky, 2003). Ce mode de gestion repose sur une décomposition hiérarchique du dialogue en un ensemble de sous-dialogues. Chaque sous-dialogue vise

une sous-tâche précise et est lui-même géré par un agenda (Rudnicky et Xu, 1999). Généralement une structure arborescente est adoptée pour traduire les relations d'ordre entre les différentes sous-tâches et actions (voir figure 2.6).

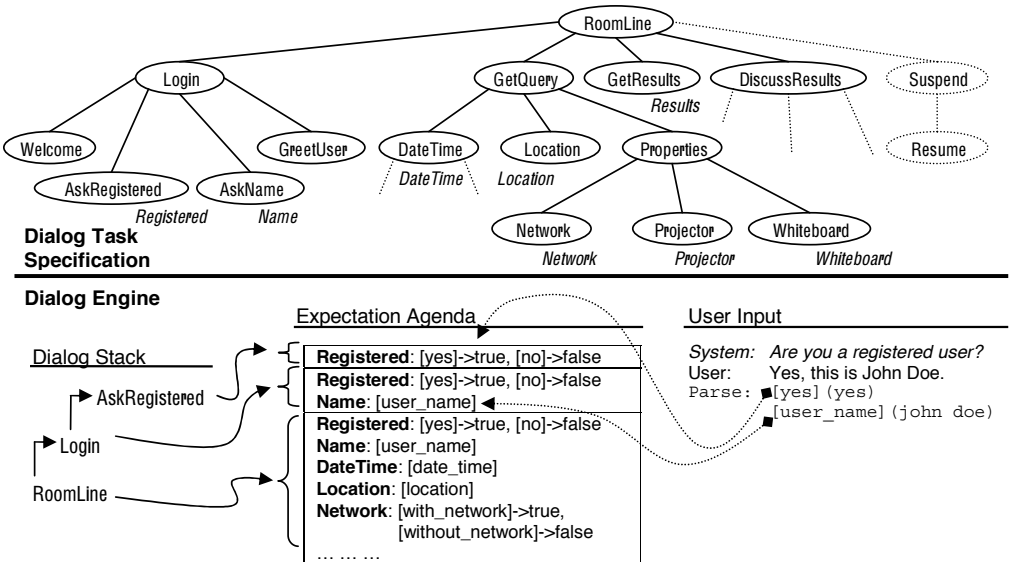


FIGURE 2.6 – *Figure représentant une approche de gestion de l'interaction par agendas, extrait de (Bohus et Rudnicky, 2003).*

Cette structure englobante est dynamiquement construite au cours du dialogue sur la base des actions système et utilisateur. Généralement, le parcours, l'extension et la détermination de la prochaine action système dans une telle structure sont effectués selon des heuristiques mais la généralisation selon un paradigme statistique est envisageable.

Paradigme du suivi de l'état d'information (ISU) une alternative consiste également à utiliser le paradigme du suivi de l'état d'information (*Information State Update* - ISU) (Traum, 1999; Larsson et Traum, 2000), employé notamment dans le système *DIPPER* (Bos et al., 2003). Ce modèle établit les principaux composants d'un système de dialogue au travers de la notion d'état d'information, des mécanismes visant à le mettre à jour au fil de l'interaction et d'une stratégie de dialogue s'appuyant dessus pour prendre la prochaine décision.

Dans ce formalisme, l'état d'information courant s_t correspond à une représentation synthétique des informations pertinentes au regard de l'ensemble des données collectées depuis le début de l'interaction (par exemple la confiance placée dans une information utilisateur, son intention supposée pour le tour courant, etc.). Les actes de dialogue système et utilisateur (resp. notés u_t et a_t), sont utilisés à chaque tour pour déclencher la mise à jour de cet état vers s_{t+1} selon des règles de la forme :

$$s_{t+1} = update(s_t, u_t, a_t) \tag{2.4}$$

Une des méthodes standards pour modéliser de tels mécanismes est l'utilisation de règles logiques (Traum et Larsson, 2003). Concernant la stratégie décisionnelle, elle doit sélectionner la prochaine action à prendre, compte tenu de l'état d'information courant parmi l'ensemble de celles possibles. Des stratégies simples peuvent être employées, comme prendre la première action applicable, de même que des techniques d'arbitrage plus complexes basées par exemple sur la théorie des jeux ou des méthodes statistiques.

Principales limites

Bien que les méthodes déterministes présentées ci-dessus ont la capacité d'atteindre de bons niveaux de performance en pratique, elles souffrent de nombreuses limites dont :

— un coût de développement très important puisque le concepteur doit modéliser de façon exhaustive le flux de l'interaction ;
— une forte dépendance au domaine applicatif ;
— de faibles capacités d'amélioration par des approches automatiques sur les données collectées, ce qui implique d'avoir recours à des raffinements manuels ;
— une faible tolérance aux erreurs de compréhension, et plus largement à l'incertitude, nécessitant l'utilisation de routines de corrections complexes (*error recovery*) et/ou d'heuristiques permettant de limiter la propagation des erreurs.

En effet, malgré leur progrès fulgurant au cours de ces quelques vingtaines d'années grâce à l'emploi des méthodes d'apprentissage automatique, les modules comme l'ASR ou encore le SLU sont toujours des sources d'erreurs. De ce fait, la prise en compte de l'incertitude sur les entrées dans le cadre des systèmes de dialogue est primordiale. Lorsque des approches déterministes sont considérées, ce rôle est généralement celui d'un gestionnaire d'erreurs qui est un module qui a pour objectif de détecter les erreurs et de proposer une stratégie pour y faire face (Bohus et Rudnicky, 2005). Une solution possible consiste à faire adopter au DM un comportement méfiant à l'égard de ses captations et a lui en faire la confirmation quasi systématique auprès de l'utilisateur. Cependant, une telle stratégie est de loin sous-optimale et lassante pour les utilisateurs du système (Paek et Pieraccini, 2008).

Une autre technique plus élaborée consiste à exploiter les scores de confiance associés aux hypothèses sémantiques transmises par la chaîne de compréhension. Le concepteur peut ainsi définir un seuil au dessus (resp. en dessous) duquel l'hypothèse sera automatiquement acceptée (resp. ignorée). Cette approche dépend donc de la qualité de la mesure de confiance adoptée. Parallèlement, le DM peut également suivre ce qu'on appelle un état d'ancrage (en anglais *grounding*) (Traum, 1994) sur les différentes informations (concepts) que peut transmettre l'utilisateur.

Cette vision s'oppose à une vision binaire (rempli ou vide) plus classique dans ce type d'approche. En effet, cet état traduit le niveau de certitude sur la compréhension de ce que l'utilisateur désire, par exemple en distinguant des situations telle que « l'utilisateur vient juste de mentionner cette valeur », « utilisateur vient de la confirmer/nier ». Le concepteur du système peut alors spécifier pour chaque élément informatif distinct un niveau attendu de certitude. Par exemple, pour les informations sensibles ce dernier

peut être requis comme devant être élevé avant d'envisager d'en faire l'usage. Le DM peut alors utiliser l'information relative à l'ancrage (*grounding* en anglais) pour contrôler au mieux le cours du dialogue. Par exemple, s'il est peu confiant dans la valeur d'un concept jugée sensible il peut alors procéder à sa confirmation dans le prochain acte de dialogue système (Roque et Traum, 2008). Il est à noter que cette approche est employée dans le paradigme ISU mentionné ci-dessus puisque l'état d'ancrage fait partie intégrante de l'état du dialogue modélisé.

2.2.2 Approches statistiques

Grâce à leurs très bon résultats dans le domaine du traitement automatique de la langue naturelle (*Natural Language Processing* - NLP), l'adoption d'approches statistiques pour la gestion de l'interaction présente de nombreux avantages. En effet, elles permettent d'envisager la conception d'un outil de gestion moins rigide capable à la fois d'optimiser la politique d'interaction du système sur la base de données d'apprentissage mais aussi d'intégrer plus proprement l'information liée à l'incertitude inhérente au problème du dialogue (erreurs des modules d'entrées).

Optimisation de la politique

Même si les méthodes d'apprentissage supervisé (nécessitant des données avec des sorties de référence) sont beaucoup employées dans la littérature du NLP, elles sont malheureusement difficilement applicables en l'état au problème de la gestion de l'interaction. La première raison qui explique cela est que ces méthodes ont généralement un besoin d'une quantité de données d'apprentissage bien supérieure au nombre des différentes instances que la tâche peut générer. Or, même au prix de la mise à disposition d'un grand corpus d'interactions annotées, ces dernières ne pourront représenter qu'une infime partie de l'ensemble des dialogues réalisables. De ce fait, seules des méthodes capables de généraliser à partir de peu d'exemples pourront être appliquées.

De plus, elles devront faire face à un autre problème plus critique. Ce dernier réside dans la nature des comportements de référence que ces méthodes vont essayer de reproduire. En effet, les comportements observés dans les données d'apprentissage, même si récoltés à l'aide d'un magicien d'Oz (*Wizard of Oz* - WoZ), dans lequel un expert joue le rôle de la Machine, ne sont pas obligatoirement ceux optimaux (Levin et Pieraccini, 2000). Quand bien même il serait possible de déterminer l'action optimale que le système aurait dû prendre a posteriori, la nature même du dialogue fait que toute modification de la prise de décision à ce point serait à même de modifier l'intégralité du cours de l'interaction.

C'est pourquoi dans la littérature, l'utilisation du RL est de loin la plus commune. Dans cette approche, la problématique de la gestion de l'interaction y est vue comme un processus de prise de décision séquentielle dans lequel la politique du DM est optimisée au regard d'une métrique reflétant la qualité de ses choix (récompenses). Ainsi le DM va apprendre à partir d'expériences (essais-erreurs), ce qu'il convient de faire

en différentes situations, de façon à récolter plus de récompenses au cours du temps. Contrairement aux approches supervisées qui reposent sur un modèle estimé à partir d'exemples vus dans un corpus d'apprentissage, un gestionnaire de dialogue basé sur RL peut explorer de nouveaux comportements et en tirer profit. Nous détaillerons tout particulièrement cette approche de référence dans le chapitre 3.

Gestion de l'incertitude

L'incertitude est une composante fondamentale dans le processus de gestion de l'interaction. En effet, les modules de compréhension mis en jeu dans la chaîne sont faillibles et leurs erreurs doivent impérativement être prises en compte par le DM. De par l'adoption de modèles probabilistes sur les modules d'entrées du système de dialogue, la tolérance aux erreurs peut être grandement améliorée par l'enrichissement des canaux de transmission entre les différents modules rendant possible la transmission de l'incertitude au plus haut niveau décisionnel (notion de *fat-pipeline*). Par exemple, cela peut se faire en transmettant une liste des N-meilleures hypothèses scorées d'un module à un autre. Dans le cadre du RL, il sera possible de prendre en compte cette incertitude dans un mécanisme d'optimisation de la politique d'interaction par l'intermédiaire d'un processus de décision de Markov partiellement observable (*Partially Observable Markov Decision Process* - POMDP). Plus de détails seront donnés dans la section 3.3.

D'autres solutions permettent également de prendre en compte cette notion dans le mécanisme de décision, notamment en la faisant intervenir lors du raisonnement des mécanismes issus de la logique floue et inspirés de la théorie de Dempster-Shafer (Shafer et al., 1976), comme la solution proposée dans (Laroche et al., 2008).

Limites

Pour dépasser la rigidité et limiter les coûts de développement et de mise à jour d'un système reposant sur des approches déterministes, de même qu'introduire une plus grande robustesse aux conditions bruitées du système, les approches statistiques doivent pour cela avoir recours à des données d'apprentissage sans quoi elles sont inexploitables. Bien que réel lors de la mise en service « de zéro » du système, ce problème est tout de même à relativiser du fait que le raffinement des approches déterministes passe également par l'étude des données (logs) par un expert (Pieraccini et al., 2009). Pour autant, nous étudierons dans cette thèse différentes techniques visant à réduire ce besoin en données mais aussi à exploiter les connaissances expertes initiales pour accélérer l'apprentissage de telles politiques.

L'autre frein à l'application de ces méthodes dans l'industrie est le manque de contrôle qu'il peut y avoir sur la stratégie finale (respect de la *VUI-completeness*[7]). En effet, la nature statistique des modèles introduits fait que le concepteur ne peut plus garantir

7. Complétude de l'interface vocale utilisateur (Voice User Interface).

que son système ne prendra jamais une action totalement aberrante dans une situation non rencontrée (Pieraccini et Huerta, 2005). De ce fait, l'industrie est hésitante à adopter cette technologie dont elle ne maîtrise pas les risques. Cependant, des techniques comme celle proposée dans (Williams, 2008a) permettent un plus grand contrôle sur la marche de manœuvre d'un DM statistique grâce à une présélection des actions réalisables par heuristiques expertes pour éviter des comportements aberrants. Nous reviendrons sur ce point plus en détail dans le chapitre 3.

2.3 L'évaluation des systèmes de dialogue

Dans le cadre des systèmes de dialogue, l'intérêt de l'évaluation est double car elle sert non seulement à pouvoir améliorer/guider le développement d'un système particulier par le biais de retours expérimentaux (mesures, diagnostics, questionnaires de satisfaction, etc.), mais aussi à des fins de comparaison entre différentes configurations, implémentations et méthodologies.

Comme nous l'avons vu précédemment, un système de dialogue résulte du chaînage (plus ou moins complexe) de plusieurs modules fonctionnels tels que l'ASR, le SLU, le NLG ou encore la base de données externe employée par le DM. De ce fait, son évaluation peut se faire soit unitairement sur chaque module (évaluation « boîte transparente », glassbox), soit de façon jointe pour évaluer le fonctionnement global de l'ensemble du système (évaluation « boîte noire », blackbox).

La premier type d'évaluation permet d'obtenir des performances précises sur le niveau de fonctionnement d'un module selon des métriques et des méthodes plus standardisées. Ce type d'évaluation est généralement plus facilement automatisable mais ne permet pas d'établir avec précision l'impact de l'amélioration d'un module sur les performances du système dans son ensemble.

Le seconde type d'évaluation consiste à établir un niveau de performance global du système. Ce dernier est généralement fortement corrélé avec l'objectif principal du système, à savoir celui de satisfaire les requêtes utilisateurs, mais il peut inclure des métriques additionnelles comme celles traduisant l'efficacité du système (temps de réponse, nombre de tours pour résoudre des tâches). Pour autant, il n'existe pas à ce jour de réel consensus dans la littérature quant aux métriques et méthodes à employer pour permettre d'en envisager l'automatisation à grande échelle.

2.3.1 Évaluation unitaire

Comme nous l'avons mentionné précédemment, il existe pour la plupart des composants d'un système de dialogue, des métriques et des méthodes bien établies pour en réaliser leur évaluation individuelle (ou unitaire) hors contexte. C'est particulièrement le cas des technologies employées pour le traitement automatique de la parole et de la langue (Hirschman et Thompson, 1997; Mariani et Paroubek, 1999) intervenant généralement en début de la chaîne du dialogue (ASR et SLU). Pour ces approches, l'évalua-

tion consiste en la comparaison des sorties du module évalué à des sorties de référence (*gold standard* en anglais). Généralement une mesure traduisant la déviation par rapport à la référence est alors extraite. On pourra donner comme exemple le taux d'erreur de mots (*Word Error Rate* - WER) pour le module ASR, ou encore le taux d'erreur en concepts (*Concept Error Rate* - CER) et la *F-mesure* pour le module de compréhension (pour plus de détails sur ces métriques le lecteur peut se référer à l'annexe B).

Plus récemment, un ensemble de métriques standards (précision, norme $L2$, les courbes ROC pour *Receiver Operating Characteristic* en anglais, etc.) a été employé avec succès pour comparer la qualité des sorties de divers systèmes visant à maintenir l'état du dialogue (estimé sur un corpus de dialogues annotés) dans le cadre des trois premières éditions du *Dialogue State Tracking Challenge* (DSTC) (Williams et al., 2013; Henderson et al., 2014a,b).

Ce type d'approche n'est pour autant pas forcement applicable avec la même pertinence à tous les composants intervenant dans la chaîne du dialogue. En effet, plus on avance dans cette dernière, plus il est difficile de restreindre les comportements des différents modules à une petite gamme acceptable. C'est notamment le cas des décisions prises par le DM (il peut y avoir un grand nombre de sorties valides à chaque point du dialogue) ou des sorties du NLG (il y plusieurs manières de traduire le même acte de dialogue système). De même, pour le TTS il n'existe bien sûr pas de références atteignables. C'est pourquoi il est nécessaire d'employer d'autres techniques d'évaluation.

En ce qui concerne le DM, et tout particulièrement la politique d'interaction, on peut par exemple procéder à son évaluation en ayant recours à la simulation ce qui présente l'« avantage » de ne pas faire intervenir les autres composants de la chaîne. Nous verrons cela plus en détail dans la section 3.5.1. Pour le NLG, à l'instar des corpus parallèles [8] utilisés en traduction, on peut mettre en correspondance les actes de dialogue système avec leur version « traduite » sous forme textuelle par un expert et évaluer les sorties en utilisant des métriques standards dans ce domaine telle que le BLEU (*BiLingual Evaluation Understudy*) (Papineni et al., 2002). Pour le TTS on peut faire l'usage de métriques subjectives telles que les notes moyennes d'opinion (*Mean Opinion Score* - MOS) qui caractérise la qualité de la restitution sonore en moyennant des notes obtenues lors d'un sondage effectué sur panel représentatif d'utilisateurs.

Un autre aspect important non couvert par les évaluations hors contexte est l'impact des erreurs d'un module particulier sur les performances du système dans son ensemble. S'il est connu que l'ASR est plus sujet aux erreurs que le SLU du simple fait de la complexité de la tâche, les méthodes d'évaluation unitaires ne permettent pas à elles seules d'expliquer les performances d'un système. Il faudra pour cela les corréler au performance du système dans son ensemble et/ou avoir recours à des études contrastives en ne faisant varier qu'un élément de la chaîne de dialogue (comme la configuration ou l'implémentation d'un module).

Certaines études se sont pourtant attachées à l'évaluation unitaire d'un module

8. Corpus dont chaque phrase est écrite à la fois dans une langue cible et dans une langue source et utilisée pour identifier des correspondances entre les unités textuelles (mots, groupes de mots, phrases) de ces deux langues.

dans le contexte de son utilisation au sein du dialogue. Par exemple dans le paradigme d'évaluation PEACE (pour Paradigme d'Evaluation Automatique de la Compréhension hors et En contexte dialogique) proposé dans (Devillers et al., 2002) le contexte de l'interaction (historique du dialogue) est intégré au travers de la définition d'une paraphrase modélisant les tours systèmes et utilisateurs précédents pour assister localement la tâche de compréhension de la parole et son évaluation contextuelle. Cependant, une des limitations de cette approche réside justement dans la création de ces dites paraphrases qui sont généralement obtenues au prix de l'adoption d'un processus semi-automatique nécessitant des corrections expertes coûteuses pour être exploitées. De plus la complexité de cette tâche de l'établissement de ces paraphrases dépend fortement de la nature du système. Par exemple dans le cas d'un système multimodal de type robot assistant, cette dernière doit également contenir toutes les informations relatives au contexte physique du monde.

2.3.2 Évaluation jointe

Le problème de l'évaluation de bout en bout d'un système de dialogue est toujours à ce jour considéré comme une tâche complexe dans la littérature. En effet, dans ce cadre il n'est pas pertinent, comme c'est le cas pour la majorité des approches standards du NLP, d'envisager l'établissement d'un comportement de référence pour le système d'un bout à l'autre de la chaîne (oracle) du fait du très grand nombre de comportements acceptables à chaque étape de l'interaction. De plus, de par sa nature, un dialogue est le fruit d'un processus liant le système à un utilisateur ; ainsi toute réponse du système qui serait différente de celle présente dans les données de référence serait à même de changer le cours global de l'interaction dans la pratique. C'est pourquoi la problématique de la conduite du dialogue ne peut pas simplement être réduite à l'étude de réactions isolées face à une nouvelle entrée utilisateur.

Ainsi, la seule méthode qui semble donc appropriée est de procéder à une mise en situation de ces systèmes, autrement dit à leur utilisation face à de vrais utilisateurs. Ce qui n'est pas sans poser de problèmes (Gandhe et Traum, 2008) : choix des sujets, coût de mise en œuvre, difficulté de recruter suffisamment de sujets pour faire des études contrastives sur différents configurations/systèmes pour que les résultats puissent être statistiquement significatifs, etc.

Cette mise en situation peut se faire sous deux formes :

— la première consiste à avoir recours à un nombre limité de sujets en conditions expérimentales. Généralement cette phase est appelée l'évaluation en condition de laboratoire (au sens large). Dans ce cas de figure l'utilisateur « réel » interagit avec le système selon des scénarios pré-établis par l'expérimentateur. Il peut soit avoir des instructions précises à suivre tout au long du dialogue, soit simplement une description générale du but qu'il doit poursuivre. Après la réalisation du scénario, l'utilisateur doit noter le système au travers d'un questionnaire (voir l'exemple du tableau 2.7). Les réponses sont soit des réponses binaires (oui/non), soit une note mise sur une échelle de Likert allant généralement de 1 à 5, 5 étant le score maximal. Le choix de ces questions est généralement complexe

car elles doivent être claires (facile à interpréter), ne cibler qu'un nombre limité d'éléments de la chaîne (résultats séparables permettant d'identifier les lacunes du système), et leur nombre doit être restreint (sinon cela risque de lasser l'utilisateur). Certains études ont proposé l'utilisation d'acteurs lors de cette phase pour s'assurer d'évaluer des comportements utilisateurs plus proche de ceux qui seront réellement observés lors du déploiement du système (Hoey et al., 2007).

— la seconde méthode consiste à réaliser une étude du système sur la base d'une solution déployée. Ici l'utilisateur est généralement un client avec de vrais besoins. Il n'y a dans cette phase qu'un contrôle très limité [9] (voire inexistant) sur les conditions expérimentales et l'évaluation se fait généralement sur la base d'analyses a posteriori sur les données collectées.

Pour effectuer l'évaluation, un certain nombre de critères ont été proposés dans la littérature et mis en relation entre eux (Walker et al., 1998; Hone et Graham, 2000; Paek, 2001; Turunen et al., 2006; Dybkjaer et al., 2004) sans qu'un réel consensus soit trouvé à ce jour du fait de la grande spécificité des systèmes étudiés (généralement ces derniers sont développés pour répondre à des besoins applicatifs précis et souvent uniques). Le constat que l'on peut faire est qu'il existe de nombreuses métriques potentiellement utilisables. Par exemple le système peut être évalué dans sa capacité à atteindre les buts de ses utilisateurs ou encore dans son aptitude à détecter et à récupérer ses erreurs de compréhension.

Globalement les métriques que l'on peut employer sont classées dans deux grandes catégories : celles dites **objectives** et celles dites **subjectives**. Les métriques objectives peut être obtenues sans le recours au jugement humain et dans la plupart des cas les informations utiles peuvent être collectées automatiquement lors de l'utilisation du système de dialogue. On pourra donner comme exemple, le nombre de tours de dialogue, la longueur moyenne des phrases système/utilisateur, le temps moyen de réponse système/utilisateur. Les métriques subjectives requièrent quant à elles une évaluation humaine pour catégoriser le dialogue ou les tours selon des critères qualitatifs. On pourra donner comme exemple, le pourcentage de réponses appropriées/non appropriées de la part du système ou encore la satisfaction de l'utilisateur.

Bien qu'il n'y ait ni métriques ni méthodes d'évaluation standards, on peut tout de même identifier dans la littérature quelques tentatives de standardisation notables. Parmi elles figure le paradigme PARADISE (pour *PARAdigm for DIalogue System Evaluation* en anglais) (Walker et al., 1997, 1998). Ce dernier propose de lier par régression linéaire la performance d'un système de dialogue, exprimée en termes de satisfaction utilisateur et obtenue par l'intermédiaire de notes attribuées par un questionnaire (le tableau 2.7 en donne un exemple) par les utilisateurs du système (critères subjectifs), à un ensemble de critère objectifs extraits d'un corpus de dialogue. Ainsi, la performance du système est exprimée sous la forme une fonction pondérée faisant intervenir l'information sur la réussite (ou l'échec) du dialogue mais aussi un ensemble de mesures traduisant des coûts (le nombre de tours, le nombre de fois où le système a procédé à une confirmation, etc.). Ces informations présentent l'avantage d'être plus facilement

9. Dans certains cas, un opérateur humain peut orienter des utilisateurs avec de vrais besoins vers une configuration particulière du système ou encore lui donner des instructions spécifiques avant l'interaction.

accessibles (extraction automatique) sur de nouvelles données et donc permette d'avoir une estimation du niveau de performance du système sans avoir recours aux questionnaires (ou une version plus simple). Il est à noter que ce paradigme a notamment été étendu au cadre multimodal dans (Beringer et al., 2002) avec la définition de PROMISE (pour *Procedure for Multimodal Interactive System Evaluation* en anglais).

Plusieurs critiques, dont celle de Larsen (Larsen, 2003), ont cependant été émises quant au fondement théorique et expérimental du choix de cette représentation linéaire de la performance. De plus, mis à part dans des situations bien particulières (but utilisateur très bien identifié), l'information sur l'accomplissement effectif de la tâche par le système n'est pas toujours quelque chose d'accessible par traitement automatique, d'autant plus lorsqu'un système est réellement déployé. En effet, cela requiert généralement que l'utilisateur donne cette information, ce qui n'est pas forcement acquis en conditions réelles (les utilisateurs ayant tendance à raccrocher en fin d'interaction). De plus, la nature objective de l'accomplissement de la tâche est automatiquement remise en cause si c'est l'utilisateur qui donne cette information. Par exemple il a été montré dans (Gašić et al., 2011) que les évaluations obtenues sur ce critère par des sujets réels (en l'occurrence recrutés via Amazon Mechanical Turk [10] pour jouer des scénarios artificiels) peuvent être sujettes à des biais préjudiciables dans le cadre d'apprentissage automatique.

Critères évalués	**Questions associées**
Performance génération	les réponses du système étaient-elles facilement compréhensibles ?
Performance compréhension	le système comprenait-il ce que vous disiez ?
Facilité de la tâche	était-il facile de trouver l'information dont vous aviez besoin ?
Rythme de l'interaction	la cadence de l'interaction vous a-t-elle paru appropriée ?
Expertise de l'utilisateur	Saviez-vous quoi dire à chaque étape de l'interaction ?
Temps de réponse système	Avez-vous ressenti des latences dans les réponses du système ?
Comportement du système	Le système s'est-il comporté de la façon dont vous l'imaginiez ?
Comparaison des interfaces	Comment jugeriez-vous l'interface vocale proposée par ce système par rapport à celle fonctionnant avec les touches clavier du téléphone ?
Utilisation future	D'après cette expérience d'utilisation, pensez-vous utiliser à nouveau ce système ?

TABLE 2.7 – *Exemple de questionnaire utilisateur proposé dans (Walker et al., 1998) (traduction)*

Malgré le fait que quelques campagnes d'évaluation aient été organisées récemment, notamment les deux éditions du Spoken Dialogue Challenge (Black et Eskenazi,

10. https ://www.mturk.com/mturk/welcome

2009; Black et al., 2011), on constate que celles-ci n'ont pas encore la maturité ni l'ampleur des celles proposées dans le cadre de NIST [11] pour le traitement de la parole. Par exemple dans la dernière évaluation en date (Black et al., 2011) quatre systèmes ont été évalués en conditions de laboratoire sur une tâche de consultation d'horaires de bus de la ville de Pittsburgh et trois d'entre eux (les plus stables) ont été comparés en conditions réelles d'utilisation.

2.4 Bilan

Dans ce chapitre nous avons introduit les systèmes de dialogue homme-machine de façon générale au travers de la définition des systèmes vocaux et multimodaux. Nous avons pour cela décrit les principaux composants mis en œuvre dans la boucle du dialogue et tout particulièrement celui du gestionnaire de dialogue, module qui a un rôle central dans la chaîne du dialogue et qui constitue un des objets principaux d'étude de cette thèse. Le développement de ce module peut être réalisé par des approches déterministes ou des approches issues de l'apprentissage automatique, et plus particulièrement de l'apprentissage par renforcement qui à ce jour fait office d'état de l'art dans ce domaine. Le prochain chapitre sera consacré à son application au contexte du dialogue.

Nous avons également discuté du problème complexe de l'évaluation d'un système de dialogue. Nous avons pour cela pris soin de distinguer les approches d'évaluation unitaires de celles jointes. Dans notre étude nos évaluations se feront dans un premiers temps de façon unitaire dans les chapitres 4 (simulateur) et 5 (corpus) puis de façon jointe dans le chapitre 6 (interactions avec de vrais utilisateurs).

11. National Institute of Standards and Technology

Chapitre 3

Apprentissage par renforcement pour la gestion de l'interaction

Sommaire

Le RL (Sutton et Barto, 1998) est un type d'apprentissage issu de la théorie du contrôle optimal (Thorndike, 1932; Bellman, 1957a). Dans ce paradigme, un système, dénommé **agent**, apprend par essais-erreurs à contrôler un **environnement** par le biais d'**actions** choisie de façon séquentielle et d'une optimisation réalisée sur la base de **récompenses** indicatrices de la qualité de ses décisions.

Comme le montre la figure 3.1, à chaque instant t, l'agent perçoit son environnement au travers d'un **état**, noté s_t. Sur cette base, l'agent devra prendre une action a_t parmi un jeu d'actions possibles. Cette prise de décision aura pour conséquence la modification de l'état courant de l'environnement, qui transitera alors vers un nouvel état s_{t+1}

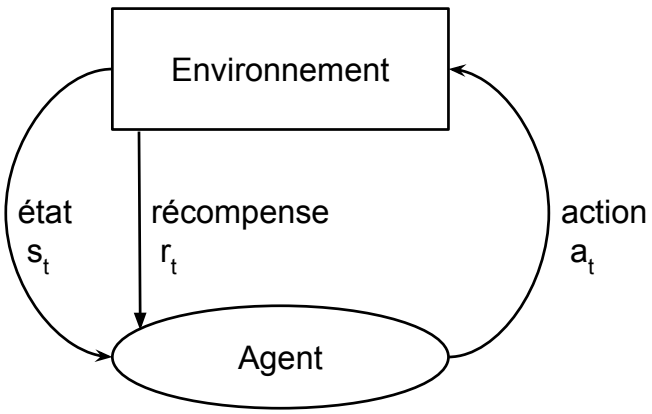


FIGURE 3.1 – *Principe du RL*

mais aussi l'émission d'une récompense r_{t+1} par l'environnement qui viendra donner un jugement de valeur sur la qualité de cette décision. Au fil des interactions, l'agent apprendra à associer aux différentes situations auxquelles il fait face (états spécifiques) les actions les plus appropriées pour maximiser les **récompenses** ainsi cumulées.

Dans le cadre du dialogue Homme-Machine, l'agent est assimilé au DM dont le but est de prendre des décisions de façon séquentielle pour conduire l'interaction vers son but souhaité (l'accomplissement de la tâche visée par l'utilisateur). Les informations permettant de modéliser l'utilisateur et celles décrivant l'état d'avancement de l'interaction courante (historique du dialogue) constituent donc l'environnement du système apprenant. Ces informations sont mises à jour par les informations en provenance des modules de compréhension mais aussi en tenant compte des réponses du système.

Parmi les solutions proposées en RL, beaucoup de travaux utilisent le cadre formel du MDP (Bellman, 1957b; Puterman, 1994). C'est d'ailleurs sous cette forme que la première application du RL pour la gestion du dialogue a été réalisée (Levin et al., 1997). Dans ce cadre formel l'état est défini de telle manière à pouvoir satisfaire la **propriété de Markov**. Cette dernière permet de considérer que la probabilité de transition vers un état futur, étant donnés les états passés et l'état présent, ne dépend que de l'état présent (absence de « mémoire »). Nous consacrons la section 3.1 à décrire ce modèle ainsi que ses principales techniques de résolution. Puis, dans la section 3.2 nous discutons tout particulièrement de son application au contexte dialogue ainsi que de ses limites pratiques et théoriques. Ensuite, dans la section 3.3 nous présentons le modèle POMDP, qui constitue une de ses extensions capable de modéliser l'incertitude inhérente au contexte applicatif visé (erreurs de compréhension, etc.), et tout particulièrement son application à la gestion du dialogue dans la section 3.4. Enfin nous conclurons ce chapitre par une présentation du cadre d'apprentissage retenu dans ce manuscrit, à savoir celui de l'apprentissage, dit en ligne, de la politique lors de sa mise en confrontation avec des utilisateurs.

3.1 Processus de Décision Markovien (MDP)

3.1.1 Définition

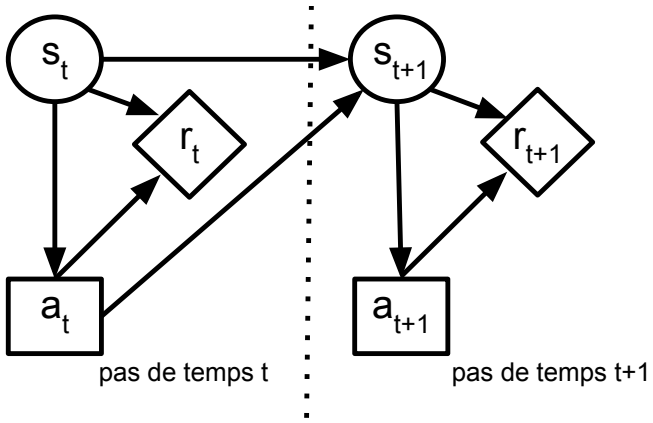


FIGURE 3.2 – *Diagramme d'influence d'un MDP. Dans cette figure les cercles représentent les variables aléatoires observables, les carrés sont les actions prises par système, les losanges sont les récompenses à valeur dans l'ensemble des réels et les flèches montrent les relations de causalités qui existent entre les différentes variables du modèle.*

Un MDP est un modèle permettant de décrire le comportement d'un agent décisionnel. Il est définit comme un quintuplet de la forme $\{S, A, T, R, \gamma\}$, où :

— S est l'espace d'état qui permet de décrire l'environnement de l'agent décisionnel. Du point de vue mathématique cet espace peut être discret, continu ou mixte ;

— A est l'espace d'action à disposition de l'agent pour contrôler son environnement. Cet espace est généralement considéré comme étant discret bien qu'il puisse être continu ;[1]

— T est la fonction de transition markovienne définit comme $T : S \times A \times S \to [0, 1]$. Sachant un état et une action courants, s_t et a_t, la probabilité du prochain état, s_{t+1}, est donnée par :

$$T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t) \tag{3.1}$$

avec $\sum_{s_{t+1} \in S} P(s_{t+1}|s_t, a_t) = 1$ pour tout couple (s_t, a_t) ;

— R est la fonction de récompense immédiate définie par $R : S \times A \to \mathbb{R}$[2]. Sachant l'état et l'action courants, s_t et a_t, la récompense immédiate que l'agent reçoit de l'environnement est donnée par :

$$r_t = R(s_t, a_t) \tag{3.2}$$

1. Certains travaux considèrent le jeu des actions comme étant fonction de l'état s_t, $A(s_t)$, pour traduire le fait que les actions ne sont pas toutes disponibles sur l'intégralité des états

2. La fonction de récompense immédiate peut être aussi définie dans la littérature par $(R : S \to \mathbb{R})$ ou $(R : S \times A \times S \to \mathbb{R})$. Cependant ce choix n'affecte en rien les propriétés fondamentales décrites dans ce manuscrit.

Généralement on considère cette fonction comme étant bornée, c'est à dire que pour tout couple (s_t, a_t) :

$$R_{min} < R(s_t, a_t) < R_{max} \tag{3.3}$$

— $\gamma \in [0, 1]$ le facteur d'escompte (ou d'actualisation) qui comme nous le verrons plus bas permet de régler l'influence des récompenses à long terme dans le mécanisme d'apprentissage.

Comme le montre la figure 3.2, à partir d'un état de l'environnement (supposé être parfaitement observable) au pas de temps t, s_t, l'agent prend une nouvelle action a_t. Cette prise de décision se fait par le biais d'une **politique**, π, qui associe à chaque état une action parmi le jeu d'actions possibles. Elle est définie [3] telle que $\pi : S \rightarrow A$, avec :

$$a_t = \pi(s_t) \tag{3.4}$$

Suite à cette action, l'environnement transite de l'état s_t à l'état s_{t+1} selon la probabilité de transition markovienne $P(s_{t+1}|s_t, a_t)$ donnée par T. L'agent apprenant reçoit alors la récompense r_t de l'environnement qui préjuge de la qualité de la transition ainsi effectuée.

La résolution d'un MDP consiste à trouver une politique optimale, notée π^*, qui maximise l'espérance des récompenses cumulées sur le long terme (fonction objectif). Généralement, une somme pondérée des récompenses sur un horizon potentiellement infini est employée comme critère d'optimisation. Elle est définie pour une politique π donnée comme suit :

$$\Re_t^\pi = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \tag{3.5}$$

Il s'agit d'une somme pondérée des récompenses obtenues en suivant la politique π depuis le pas de temps t. La pondération par une puissance de γ permet de régler l'importance des récompenses immédiates par rapport à celles qui seront collectées ensuite. Ainsi défini proche de 1 [4], γ permet de tenir compte du futur dans l'optimisation du choix de l'action.

3.1.2 Techniques de résolution d'un MDP

Bien qu'une approche exhaustive de type « force brute » (*brute force* en anglais), qui teste l'ensemble des politiques possibles et prend celle ayant la meilleure espérance sur $\Re$, est envisageable sur des problèmes très simples, le nombre extrêmement grand de politiques envisageables (voire infini pour certains problèmes) fait que la résolution d'un MDP requiert généralement l'adoption de techniques de résolution d'une tout

3. Bien qu'il existe d'autres manières de représenter cette politique, par exemple en adoptant une vue stochastique du processus décisionnel, dans cette thèse nous nous concentrerons sur une définition déterministe.

4. Le fait de définir γ strictement inférieur à 1 permet dans des cas où l'horizon est infini de garantir que l'équation 3.5 est bornée (sachant que la fonction de récompense immédiate l'est).

autre nature. Parmi elles on pourra distinguer quelques grandes familles, comme les approches basées sur l'estimation d'une fonction de valeur ou encore celles faisant une recherche directe de politique.

Dans la suite de ce manuscrit les techniques que nous emploierons reposent sur l'estimation d'une **fonction de valeur** (et ses dérivées) afin de résoudre le problème du MDP. C'est pourquoi nous les détaillerons tout particulièrement ci-dessous. Cependant il faut savoir qu'une des stratégies alternatives est de voir le problème comme celui d'une optimisation stochastique (Spall, 2005) où l'agent est à la recherche d'une politique dans (une sous-partie) l'espace complet des politiques (comme par exemple (Daubigney et al., 2013) voir ci-après).

Méthodes basées sur l'estimation d'une fonction de valeur

La fonction de valeur est définie par $V^\pi : S \rightarrow \mathbb{R}$. Elle représente l'espérance moyenne sur $\Re$ (voir l'équation 3.5) en suivant une politique donnée π [5]. Ainsi, pour une stratégie π donnée et un état initial s_0, cette fonction est définie comme suit :

$$V^\pi(s_0) = E_\pi[\Re_t^\pi | s_t = s_0] = E_\pi[\sum_{i \geq 0} \gamma^t r_{t+i} | s_t = s_0, a_i = \pi(s_i)] \tag{3.6}$$

La fonction de valeur peut également être définie récursivement par :

$$V^\pi(s) = E[r_t + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = \pi(s)] \tag{3.7}$$

Trouver une politique optimale π^* pour le MDP consiste à déterminer une politique qui maximise la fonction de valeur. Soit :

$$\pi^*(s) = \arg\max_\pi V^\pi(s) \tag{3.8}$$

Par convention on notera V^* la fonction de valeur associée à la stratégies optimale π^* [6] où :

$$V^*(s) = \max_a E_{s_{t+1}|s_t,a_t}[r_t + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] \tag{3.9}$$

L'équation 3.9 correspond à celle de l'optimalité de Bellman (Bellman, 1957a).

De façon similaire, on peut définir la **fonction de qualité** par $Q^\pi : S \times A \rightarrow \mathbb{R}$. Cette fonction ajoute un degré de liberté sur la première action sélectionnée :

$$Q^\pi(s, a) = E[r_t + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1})) | s_t = s, a_t = a] \tag{3.10}$$

5. La fonction V^π est également dénommée comme étant l'équation d'évaluation de Bellman (Bellman, 1957a) dans la littérature.

6. Pour une fonction de valeur donnée il peut exister plusieurs politiques optimales et vice versa (Sutton et Barto, 1998).

Il est à noter qu'entre les équations 3.7 et 3.10 il existe la relation suivante :

$$V^{\pi}(s) = Q^{\pi}(s, \pi(s)) \tag{3.11}$$

Q^* correspond à la fonction qualité de la stratégie optimale π^*. Elle est définie comme suit :

$$Q^*(s,a) = E_{s_{t+1}|s_t,a_t}[r_t + \gamma \max_{a'} Q^{\pi}(s_{t+1}, a')|s_t = s, a_t = a] \tag{3.12}$$

Si cette fonction est connue, la stratégie optimale consiste à suivre systématiquement la meilleure action, choix de l'action gloutonne (greedy en anglais), selon Q^*. C'est à dire :

$$\pi^*(s) = \arg\max_{a} Q^*(s,a) \tag{3.13}$$

Le fait de chercher à déterminer π^* au travers de l'équation 3.9 ou 3.12 est appelé « résoudre » ou « optimiser » le MDP. Un des principaux débats scientifiques dans la littérature du RL porte sur l'utilisation ou non d'un modèle pour effectuer cette résolution.

Dans le cas des approches dites **sur modèle** (model-based en anglais), les dynamiques du MDP, à savoir les probabilités de transitions définies par T et la fonction de récompense immédiate R, sont supposées connues et représentées au travers d'un modèle. Dans ces conditions la politique optimale peut alors être déterminée **hors-ligne** (off-line en anglais), c'est-à-dire sans interaction directe avec l'environnement. Les techniques issues de la programmation dynamique (Bertsekas, 1995) sont notamment capables de résoudre le problème de façon exacte à condition que les espaces S et A considérés soient de petites dimensions. Nous pouvons citer deux algorithmes de ce type, celui de l'*itération de la politique* et celui de l'*itération de la valeur*. Malgré leur efficacité notable, leur application reste limitée. Premièrement du fait qu'ils sont difficilement applicables dans des cas où l'espace espace d'état est grand. Deuxièmement, à cause du fait qu'ils requièrent la connaissance des dynamiques du système, ce qui est rarement le cas pour des problèmes réels.

À contrario, les approches dites **sans modèle** (model-free en anglais) ne requièrent quant à elles aucune connaissance a priori sur les dynamiques de l'environnement et ne cherchent pas à les représenter explicitement. Elles reposent exclusivement sur l'estimation d'une forme approximée de la fonction de valeur (ou de la fonction de qualité) sur la base des trajectoires observées (quadruplets $\{s_t, a_t, r_t, s_{t+1}\}$). Ces méthodes permettent d'obtenir des solutions (quasi-)optimales à condition de faire une exploration de l'espace suffisante (tests des couples état-action). On peut faire mention de la méthode de *Monte-Carlo* mais aussi de celles basées sur les *différences temporelles* (e.g. SARSA, Q-learning, TD(λ), etc.). Un panorama plus complet de ces techniques est disponible dans (Sutton et Barto, 1998).

Il est également utile de faire une distinction supplémentaire entre les techniques dites **sur politique** (on-policy en anglais) et celles dites **hors politique** (off-policy en anglais).

Le méthodes sur politique sont des méthodes itératives alternant des phase d'évaluation et d'amélioration de la politique. Elles se basent sur l'estimation courante de la fonction valeur (resp. qualité) en l'état courant pour déterminer le choix de la prochaine action (contrôle), après observation du nouvel état courant et du signal de renforcement reçu, le modèle qui a été employé est mis à jour. Un exemple classique de ce type de méthode est l'algorithme SARSA.

Les méthodes hors politique ne sont quant à elles pas sensibles à la manière dont les actions sont sélectionnées à chaque instant mais seulement au fait d'observer une politique de contrôle présentant un niveau d'exploration suffisant. De ce fait elles peuvent librement observer une politique de contrôle différente (pouvant être sous-optimale). Une exemple classique d'algorithme hors politique est l'algorithme Q-learning.

De façon générale, les méthodes sur politique obtiennent de meilleures récompenses durant la phase d'apprentissage en-ligne et convergent plus rapidement vers la politique optimale. Cependant elles présentent la limite théorique de se baser uniquement sur les trajectoires réellement observées, contrairement aux méthodes hors politique qui elles sont capables de généraliser. Ceci est dû au fait que ces méthodes utilisent l'équation d'optimalité de Bellman (voir équation 3.12) et donc l'opérateur non-linéaire max dans leur processus d'optimisation. Ces dernières sont très utiles dans les situations où un corpus d'apprentissage obtenu avec une politique sous-optimale est à disposition, car elles pourront tout de même espérer apprendre une politique optimale. Le lecteur pourra se référer au problème de « la traversée de la falaise » (Cliff Walking problem en anglais) tel que présenté dans (Sutton et Barto, 1998) pour avoir une illustration du fonctionnement de ces deux techniques sur un cas concret d'apprentissage.

Nous reviendrons plus en détail sur ces notions dans leur application au contexte du dialogue dans la section 3.5.

3.2 Limites théoriques du MDP pour le problème du dialogue

La première application du modèle MDP à la problématique du dialogue a été présentée dans (Levin et al., 1997). Par la suite de nombreux travaux ont également adopté ce formalisme pour optimiser la gestion de l'interaction sur la base de données (Singh et al., 1999; Levin et Pieraccini, 2000; Litman et al., 2000; Goddeau et Pineau, 2000; Young, 2000; Singh et al., 2002). La plupart des travaux avaient alors recours soit à des technique d'apprentissage sur modèle soit à la simulation d'utilisateurs (voir section 3.5.1). Grâce à l'adoption de cette modélisation du problème, de nombreux travaux ont montré qu'il était possible d'obtenir de meilleurs résultats (sur des critères objectifs et subjectifs) que ceux obtenus avec des approches déterministes faisant l'usage d'heuristiques expertes (Litman et al., 2000; Singh et al., 2002).

Cependant, la mise en œuvre d'un tel formalisme n'est pas sans difficulté dans la pratique. En effet, une bonne définition des éléments de base du formalisme MDP est essentielle. En effet, les espaces S, A et R doivent être choisis avec précaution pour permettre l'apprentissage.

Concernant S, la contrainte est que l'état du dialogue doit à la fois pouvoir respecter la propriété de Markov (condition sine qua none pour l'utilisation du modèle MDP) tout en restant assez compact pour pouvoir permettre l'apprentissage d'une politique de qualité en des temps raisonnables. Dans la littérature, de nombreuses techniques ont été employées pour définir un espace d'état de taille réduite sur la base de corpus d'interactions. Dans (Singh et al., 1999), les informations utiles à la poursuite du dialogue sont conservées au travers d'un ensemble d'indicateurs booléens au prix d'une perte en expressivité de la modélisation. D'autres travaux, comme (Young, 2000), ont considéré l'utilisation d'heuristiques pour passer d'un état très informatif à un état réduit pour rendre possible l'apprentissage. Dans (Denecke et al., 2004) une méthode de regroupement (*clustering* en anglais) est employée pour agréger les états jugés « similaires ».

De façon similaire, A doit lui aussi avoir une taille raisonnable pour permettre l'apprentissage. Pour ce faire, dans la plupart des travaux (dont ceux mentionnés ci-dessus), des actions haut niveau (ou méta-actions) ont été employées conjointement à des heuristiques pour les faire correspondre à de vraies actions système. Des méthodes d'approximation de la fonction de valeur sont également employées pour interpoler l'espace de recherche et ainsi réduire le nombre de paramètres à apprendre. Nous détaillons ces techniques dans la section 3.4.3.

Pour ce qui est de R, la plupart des études font usage d'une définition experte et fortement dépendante du domaine. Généralement cette fonction repose sur des informations objectives telles que le critère réussite de la tâche, le nombre de tours système (traduisant son efficacité) ou encore le nombre d'accès à la base de données. Dans (Paek, 2006), l'auteur considère même qu'il s'agit là de l'aspect le plus artisanal du modèle. Outre l'aspect sous-optimal que cela peut représenter (puisqu'elle représente à elle seule le critère d'optimisation du RL), il peut se poser également le **problème de l'affectation temporaire de valeur**, en anglais *temporal credit assignment problem*, dans le mécanisme d'apprentissage. En effet, bien souvent en employant de telles fonctions de récompenses, l'agent apprenant n'aura accès aux retours les plus pertinents que tardivement dans le processus d'apprentissage (temporellement parlant). Pour le cas particulier du dialogue, avant que l'agent conduise le dialogue jusqu'à la fin de l'interaction, étape à laquelle il recevra la récompense la plus importante liée à la réussite ou à l'échec de la tâche utilisateur, il va d'abord devoir passer par un ensemble d'états pour lesquels ses décisions seront récompensées de façon identique (pénalité d'un tour de parole). La conséquence de cet éloignement temporel (les premiers dialogues étant généralement longs) est que le signal de récompense final ne sera que faiblement répercuté sur les décisions locales qui l'ont précédé.

Ainsi, l'influence d'une récompense devient de plus en plus faible au fil du temps, ce qui peut conduire à de mauvaises propriétés de convergence du mécanisme de RL lors de l'initiation du processus d'apprentissage. De ce fait, la conduite (ou l'observation) par l'agent de nombreux épisodes d'apprentissage (dialogues) est généralement de rigueur pour réussir à propager de façon itérative l'influence du signal de renforcement tardif à tous les couples état-action qui ont contribué à sa génération. Certaines études se sont attachées à déterminer des récompenses plus diffuses pour accélérer l'apprentissage notamment à l'aide de technique comme le *reward shaping* (El Asri et al., 2013;

Su et al., 2015) ou proposer des techniques comme celle de l'apprentissage par renforcement inverse (*Inverse Reinforcement Learning* - IRL) (Russell, 1998) pour apprendre une fonction de récompense sur la base de données (Paek et Pieraccini, 2008; Boularias et al., 2010; El Asri et al., 2012) ou encore par régression linéaire (Rieser et Lemon, 2011). Cependant toutes ces techniques supposent que des exemples d'interactions de qualité sont disponibles. Ces données sont obtenues en ayant recours à des collectes par WoZ ou par l'utilisation de système pré-existant avant d'être annotées par des experts, ou encore en utilisant un outil de simulation d'utilisateurs pour collecter des dialogues artificiels.

Pour autant, la principale limite de l'application de la modélisation par MDP à la problématique du DM n'est pas pratique mais théorique. En effet ce modèle suppose l'observabilité totale de l'état du dialogue. Or, l'état mental de l'utilisateur n'est pas quelque chose d'accessible pour la Machine mais quelque chose de perçu au travers de ses actes. De plus, les modules de compréhension sont eux-même sujets aux erreurs. De ce fait, l'agent apprenant n'a à sa disposition qu'une observation incertaine de l'état sous-jacent dans lequel se trouve l'environnement. Bien que la définition de l'état du MDP peut en pratique inclure des informations relatives à l'incertitude (Bohus et Rudnicky, 2005), une approche qui semble plus fondée théoriquement est d'avoir recours à un POMDP (Sondik, 1971). Ce modèle, décrit plus en détail dans la prochaine section, modélise l'état du dialogue comme une variable latente dont la distribution est estimée sur la base des observations du système. Par cette prise en compte explicite de l'incertitude, ce formalisme dispose de caractéristiques capables d'améliorer la robustesse d'un système de dialogue.

Dans la littérature des approches alternatives au POMDP ont été proposées. Une d'entre elle repose sur l'utilisation de techniques « boîte noire », comme dans (Daubigney et al., 2013) où un algorithme d'optimisation par essaims particulaires est employé pour résoudre le problème du contrôle optimal sous incertitude du DM. On pourra également mentionner l'étude réalisée dans (Cuayáhuitl et al., 2007) qui fait usage d'un mécanisme de RL hiérarchique (Barto et Mahadevan, 2003) en ayant recours au découpage du problème MDP initial en sous-tâches par l'adoption d'une hiérarchie de processus décisionnels semi-Markoviens[7].

3.3 Processus de Décision Markovien Partiellement Observable (POMDP)

3.3.1 Définition

Un POMDP est défini par le n-uplet $\{S, A, T, R, O, Z, \gamma, b_0\}$, où :
— $\{S, A, T, R, \gamma\}$ est un MDP

7. Un modèle semi-Markovien est une généralisation du modèle Markovien classique obtenue par l'ajout du facteur temps dans la probabilité de transition du modèle ; ainsi, le temps de séjour dans un état sera modélisé et pris en compte.

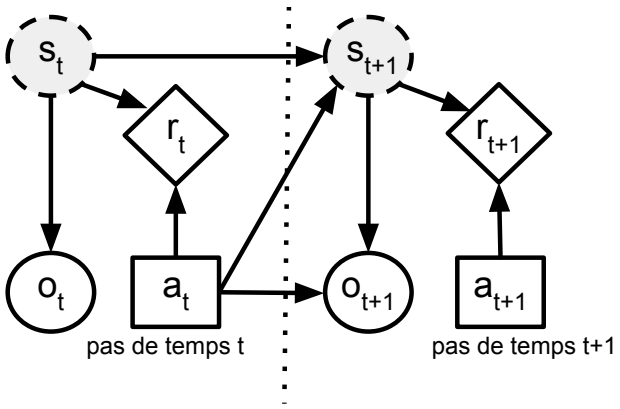


FIGURE 3.3 – *Diagramme d'influence d'un POMDP. Dans cette figure les cercles grisés correspondent aux variables aléatoires non-observables, les cercles clairs sont les variables aléatoires observables, les carrés sont les actions du système, les losanges sont les récompenses à valeurs réelles et les flèches montrent les relations de causalités qui existent entre ces variables.*

— O est l'espace d'observation
— Z est un jeu de probabilité sur les observations tel que :

$$Z = \{P(o_{t+1}|s_{t+1}, a_t), \forall (o_{t+1}, s_{t+1}, a_t) \in O \times S \times A\} \tag{3.14}$$

— b_0 est l'état de croyance initial (avant que la première action système soit prise)

Dans ce cadre formel, à un instant t, le monde est dans un état non observable s_t. Du fait de cette non observabilité, le système maintient une distribution sur l'ensemble des états possibles appelé **état de croyance** et noté b_t, où $b_t(s_t)$ correspond à la probabilité d'être dans l'état de dialogue s_t. Sur la base de b_t, le système choisit l'action a_t, reçoit une récompense immédiate r_t et transite vers un nouvel état non observable s_{t+1} selon un loi de transition markovienne (dépendance se limitant à s_t et a_t). La Machine reçoit comme indice de cette transition une observation o_{t+1} qui dépend uniquement de s_{t+1} et a_t, et met à jour son état de croyance selon l'équation suivante :

$$b_{t+1}(s_{t+1}) = \eta P(o_{t+1}|s_{t+1}, a_t) \sum_{s_t \in S} P(s_{t+1}|s_t, a_t) b_t(s_t) \tag{3.15}$$

où

$$\eta = \frac{1}{(\sum_{s_{t+1} \in S} P(o_{t+1}|s_{t+1}, a_t) \sum_{s_t \in S} P(s_{t+1}|s_t, a_t) b_t(s_t)} \tag{3.16}$$

est une constante de normalisation.

Ce processus est illustré sur la figure 3.3 grâce à un diagramme d'influence. Il est à noter que dans ce modèle les états sous-jacents respectent la propriété de Markov, mais pas les observations.

3.3.2 Techniques de résolution d'un POMDP

Dans ce formalisme la fonction de valeur peut être réécrite de la façon suivante :

$$V^{\pi}(b_t) = r(b_t, \pi(b_t)) + \gamma \sum_{o_{t+1}} P(o_{t+1}|b_t, \pi(b_t)) V^{\pi}(b_{t+1}) \tag{3.17}$$

Si théoriquement la résolution exacte d'un POMDP est possible (Cassandra et al., 1997; Kaelbling et al., 1998), la complexité du modèle (dimension des espaces considérés) fait qu'elle ne l'est que pour des tâches ne faisant intervenir qu'un nombre très limité d'états et d'actions (Littman et al., 1995). Au prix d'approximations, certaines méthodes permettent cependant d'envisager l'utilisation d'un modèle POMDP sur des tâches plus complexes. On peut mentionner les approches de résolution exploitant les dynamiques du modèle (généralement apprises sur un corpus) telles que SPOVA (Parr et Russell, 1995), PBVI (Pineau et al., 2003) ou encore Perseus (Spaan et Vlassis, 2005). Cependant dans cette thèse nous nous concentrerons sur des approches sans modèle qui nécessitent une définition compacte de l'espace d'état et l'utilisation de techniques élaborées de représentation de la politique.

D'un point de vue mathématique, un POMDP peut être assimilé à un MDP pour lequel l'espace d'état correspond à celui d'un état continu sur l'espace d'état de croyances (noté B) avec le même espace d'action A. Ainsi, ce qui a été écrit pour les MDP dans la section 3.1.2 s'applique donc directement aux POMDP. Il est à noter que des méthodes comme celle proposée dans (Dutech et Samuelides, 2003) (fenêtre glissante sur les observations) ou encore dans (Szita et al., 2006) (RNN) peuvent être employées pour définir un état « étendu » de dimension finie composée d'éléments non ambigus afin de s'assurer du respect de la propriété de Markov de l'état, et ainsi garantir théoriquement l'atteinte à l'horizon infini d'une solution optimale par les approches RL standards. Cependant la complexité de la tâche du dialogue rend difficile l'application de telles techniques et explique pourquoi nous aurons recours à des approximations du modèle.

3.4 Application au dialogue du POMDP

Depuis sa première application à la problématique du dialogue dans (Roy et al., 2000), le modèle POMDP s'est progressivement imposé comme le modèle état de l'art dans la littérature. Cependant, bien qu'ayant l'avantage de pouvoir modéliser de manière explicite l'incertitude et l'ambiguïté inhérentes au processus d'interaction entre l'Homme et la Machine, son application sur des tâches concrètes n'en est pas pour autant triviale. Le principal problème réside dans l'explosion combinatoire empêchant le calcul de l'équation 3.15 ainsi que l'apprentissage de la politique optimale. En effet, même pour un système de dialogue de taille raisonnable, le nombre total d'états, d'actions et d'observations possibles peut facilement atteindre un ordre de grandeur de 10^{10}.

C'est pourquoi dans la littérature de nombreuses solutions ont été envisagées pour exploiter les propriétés spécifiques liées à la tache du dialogue pour permettre de dé-

finir des représentations compactes du modèle (espaces manipulés) et de la politique. L'objectif visé étant de permettre à des algorithmes de mettre à jour l'état de croyance du système et l'optimisation de la politique avec moins de ressources et en des temps raisonnables.

3.4.1 Représentation et maintien de l'état de croyance

Une des premières solutions pour obtenir une représentation plus efficace est d'effectuer une décomposition de l'état non observable représentant l'utilisateur et le dialogue. Nous détaillerons tout particulièrement celle proposée dans (Poupart, 2005) du fait de son utilisation dans de nombreux travaux de référence (Williams et al., 2005; Young et al., 2010; Thomson et Young, 2010).

Ainsi dans cette proposition l'état est décomposé en trois composants :

$$s_t = (g_t, u_t, h_t) \tag{3.18}$$

g_t représente le but actuellement poursuivi par l'utilisateur (par exemple l'état de remplissage du formulaire représentant le but utilisateur), u_t représente ce qu'il a effectivement dit à ce tour de dialogue (acte de dialogue utilisateur du tour courant) et h_t est l'historique du dialogue qui conserve une trace des informations pertinentes issues des tours précédents (par exemple l'état d'ancrage sur les champs du formulaire).

En suivant cette décomposition et en posant des hypothèses d'indépendances raisonnables, le modèle POMDP peut être représenté par le diagramme d'influence de la figure 3.4.

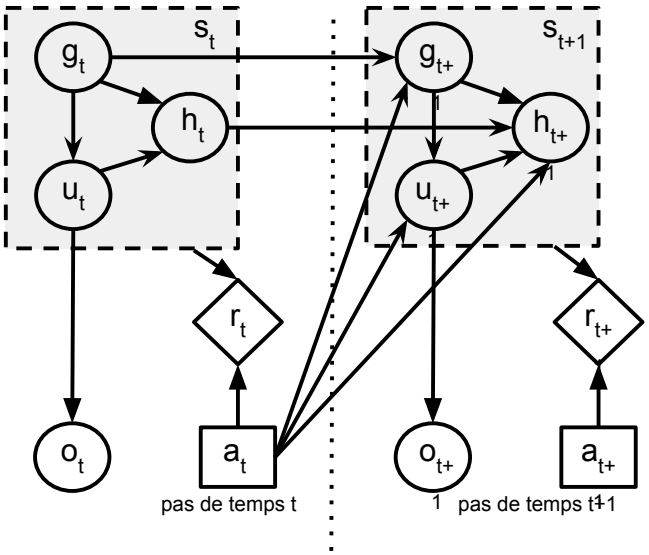


FIGURE 3.4 – *Diagramme d'influence d'un POMDP dont l'état est factorisé en trois éléments g_t, u_t et h_t.*

L'équation 3.15 peut alors être simplifiée de la façon suivante :

$$b_{t+1}(g_{t+1}, u_{t+1}, h_{t+1}) = \eta . \underbrace{P(o_{t+1}|u_{t+1})}_{\substack{\text{modèle} \\ \text{d'observation}}} \underbrace{P(u_{t+1}|g_{t+1}, a_t)}_{\substack{\text{modèle d'action} \\ \text{utilisateur}}} \sum_{g_t} \underbrace{P(g_{t+1}|g_t, a_t)}_{\substack{\text{modèle de but} \\ \text{utilisateur}}} \sum_{h_t} \underbrace{P(h_{t+1}|g_{t+1}, u_{t+1}, h_t, a_t)}_{\substack{\text{modèle d'historique} \\ \text{du dialogue}}} b_t(g_t, h_t) \tag{3.19}$$

Dans cette formulation il est possible de distinguer 4 modèles :

— le **modèle d'observation** permet déterminer la probabilité de l'observation courante sachant l'énoncé utilisateur. Ce modèle tient compte des éventuelles erreurs de compréhension ;

— le **modèle d'action utilisateur** estime la vraisemblance de l'acte de dialogue utilisateur tel que perçu par le système sachant l'action précédente du système et le but poursuivi par l'utilisateur ;

— le **modèle de but utilisateur** donne la probabilité de transition d'un but à un autre (changement de but) ;

— le **modèle d'historique du dialogue** met à jour les informations liées à l'historique de l'interaction depuis le premier tour du dialogue.

En pratique ces modèles sont soit définis de façon *ad hoc* soit en ayant recours à des techniques issues de l'apprentissage automatique. Nous en verrons dans la section 3.4.4 un exemple concret d'implémentation. Bien que cette décomposition diminue de façon significative la complexité du modèle POMDP le recours à des approximations supplémentaires est généralement indispensable pour rendre possible la résolution.

Parmi les pistes envisagées dans la littérature, une d'entre elle consiste à modéliser l'état de croyance complet du système par une distribution limitée aux hypothèses les plus probables au regard des observations. Dans le paradigme de l'état de l'information caché (*Hidden Information State* - HIS) (Young et al., 2010), cela est rendu possible par un regroupement des états en classe équivalente (indistinguable au regard des observations). Ainsi, les buts utilisateur (l'ensemble des g_t possibles) sont modélisés au travers de **partitions** avec comme hypothèse que tous les buts au sein d'une même partition sont équiprobables. Ces dernières sont construites au fil de l'interaction en exploitant à la fois l'ontologie du domaine (relation entre les concepts), la dernière action du système et les observations en provenance des modules de compréhension. Il est à noter que nous donnons une description plus détaillée de ce paradigme dans la section 3.4.4, vu que nous l'avons employé et même enrichi dans nos travaux. Dans (Henderson et al., 2008), les auteurs proposent une approche similaire où l'état de croyance est représenté comme une distribution sur des états MDP obtenus par l'application de règles expertes sur les observations. Cette approche utilise ensuite les probabilités des observations pour déterminer la probabilité de chaque état du MDP ainsi construit.

Une approche alternative aux deux propositions précédentes consiste à procéder à une décomposition supplémentaire du but utilisateur en éléments conditionnellement indépendants. Généralement une décomposition au niveau des attributs mis en jeu dans l'ontologie du domaine est employée. Cette technique peut notamment per-

mettre de modéliser un réseau bayésien dynamique sur la base de connaissances expertes. Cette idée a notamment été employée dans (Bui et al., 2009) dans un scénario où les attributs du domaine sont tous conditionnellement indépendants ; dans ce cadre particulier un algorithme standard de propagation des croyances (*belief propagation* en anglais) peut être employé pour mettre à jour l'état de croyance. Dans un formalisme semblable, l'approche BUDS (*Bayesian Update of Dialogue State*) (Thomson et Young, 2010) permet de modéliser également un jeu limité de dépendances en faisant l'usage d'une approximation itérative de l'algorithme de propagation des croyances (*loopy belief propagation* en anglais). Une approche comparable a également fait l'usage de filtres à particules (Williams, 2007) pour approximer l'état de croyance et le mécanisme de sa mise à jour.

Généralement ce type de techniques exploite de nombreux paramètres que le concepteur peut en tout premier lieu initialiser par intuitions (par exemple pour accélérer leur déploiement) mais aussi optimiser en exploitant des données (Thomson et al., 2010; Jurcicek et al., 2010; Lee et al., 2014).

D'autres approches, comme celles discriminantes, font l'usage des données préalablement collectées pour apprendre des classifieurs capables d'estimer l'état de croyance (distribution jointe sur les valeurs de chaque concepts - généralement approximée par la combinaison des différentes probabilités marginales) à chaque tour de dialogue. La première approche de ce type pour le dialogue est celle faite dans (Bohus et Rudnicky, 2006), avec l'emploi d'un modèle linéaire de maximum d'entropie (*MaxEnt* dans la littérature). Ces méthodes ont notamment rencontrées un franc succès sur les trois premières éditions du DSTC (Williams et al., 2013; Henderson et al., 2014a,b) où les performances de divers systèmes permettant de maintenir l'état de croyance étaient évaluées sur la base de données collectées et annotées. Parmi ces méthodes on pourra mentionner celles faisant usage des CRF (Lee, 2013; Kim et Banchs, 2014), des réseaux de neurones profonds et récurrents (Henderson et al., 2013, 2014b,a) ou encore celles ayant recours à des techniques discriminantes utilisées en recherche d'information pour ordonner les résultats d'une requête web comme dans (Williams, 2014) où l'algorithme *LambdaMART* (Burges, 2010) est employé pour déterminer l'état de croyance du système.

Cependant même si ces techniques ont des caractéristiques intéressantes, le fait qu'elles requièrent des données d'apprentissage constitue quelque chose de limitant lorsqu'on considère une nouvelle tâche où ces données ne sont pas disponibles. Comme nous l'avons déjà mentionné, dans ce manuscrit nous ferons l'usage exclusif de l'approche HIS. Cette dernière nous permet de tenir compte de l'ensemble des dépendances qu'il existe entre les concepts (défini au travers d'une ontologie du domaine) au prix du maintien d'une distribution partielle de l'état de croyance.

3.4.2 Réduction des tailles des espaces considérés

La réduction des tailles des espaces considérés (S, A, O, B) dans un POMDP est une piste de recherche essentielle. Elle vise à permettre l'utilisation d'une représentation et d'une optimisation de politique plus compactes et plus simples.

Dans (Lefèvre et de Mori, 2007) et (Pinault et al., 2009; Pinault et Lefèvre, 2011b), les auteurs proposent par exemple d'utiliser des méthodes de *clustering* automatique pour obtenir un POMDP plus simple dont il est possible d'estimer les dynamiques sur la base de données pour envisager sa résolution par une approche sur modèles.

Dans l'approche CSPBVI présentée dans (Williams et Young, 2006), le POMDP est décomposé en de multiples MDP résumés. La décision du système étant déterminée sur la base d'heuristiques sélectionnant la meilleure action maître au regard des décisions prises par les différents MDP.

Certains travaux exploitent un résumé de l'état de croyance. Ce résumé est généralement obtenu par l'intermédiaire des descripteurs hétérogènes (valeurs réelles, binaires et catégorielles). On peut donner comme exemple la valeur croyance des N meilleures hypothèses sur l'état du dialogue, la valeur de croyance marginale sur les différents concepts, le nombre de résultats de la base de données associés à la meilleure hypothèse du but utilisateur, l'état d'ancrage des différents concepts, voir des combinaisons de ces paramètres (Thomson et al., 2008; Williams et al., 2005; Williams, 2008a; Young et al., 2010; Li et al., 2009). Ces descripteurs sont généralement choisis par le concepteur du système. Certains travaux ont cependant envisagé leur sélection automatique sur la base de données comme dans (Li et al., 2009; Pietquin et al., 2011).

Certaines études font également l'usage d'un espace résumé des actions système. Une méthode classique consiste alors à faire correspondre la liste des actions résumées à celle des différents types d'acte de dialogue (*inform*, *confirm*, etc.) applicables aux concepts de la meilleure hypothèse (Williams et al., 2005; Thomson et Young, 2010). Bien qu'ayant l'avantage d'être entièrement automatisable, cette méthode autorise le système à sélectionner des actions que l'on sait erronées dans le contexte courant de l'interaction, comme par exemple le fait de saluer l'utilisateur en plein milieu de l'interaction. Afin de contourner ce problème, une alternative consiste à utiliser des heuristiques expertes (Williams, 2008a) ou des réseaux de Markov logiques (*Markov Logic Networks* - MLN) (Lison, 2010) pour contraindre l'espace de recherche aux actions possibles. Ce mécanisme peut sensiblement accélérer la convergence de l'apprentissage (Williams, 2008a). Cependant l'ajout de telles règles a un coût et l'action optimale peut être exclue par erreur. Une méthode intermédiaire, notamment employée dans le paradigme HIS (Young et al., 2010), consiste cette fois à autoriser toutes les actions résumées à tous les instant et à appliquer des heuristiques expertes pour contraindre les concepts sur lesquels vont s'appliquer ses actions et permettre éventuellement le repli sur une autre action moins optimale mais réalisable (Gašić et al., 2009).

3.4.3 Représentation de la politique

Étant donné que l'état résumé de croyance employé dans la plupart des travaux est souvent de nature hétérogène (variables continues, booléenne ou encore catégorielles) et de grande cardinalité (voire infini lorsque des variables continues sont considérées, par exemple des probabilités), il convient d'employer des techniques visant à faire une approximation de la fonction de valeur.

Une des méthodes souvent employée dans la littérature consiste à avoir recours à la **discrétisation** de l'espace de croyance. Par exemple en procédant à un pavage de l'espace par un jeu de points connus et définissant une métrique de distance capable d'estimer la valeur en de nouveaux points (Young et al., 2010). Des techniques telles que celle des k plus proches voisins (*k-Nearest Neighbors* - k-NN) (Lefèvre et al., 2009) peuvent être employées pour mieux tenir compte du voisinage et améliorer l'estimation lorsque la quantité de données est limitée.

Une autre approche consiste à adopter une **représentation paramétrique** de la fonction de valeur. L'objectif de l'apprentissage RL est alors d'estimer un jeu de paramètres permettant de déterminer la fonction valeur en tout point de l'espace.

Cette paramétrisation peut être **linéaire**. Dans ce cas la fonction de qualité approximée peut être représentée par :

$$\hat{Q}_\theta(s_t, a_t) = \theta^T \phi(s_t, a_t) \tag{3.20}$$

Dans ce cas de figure, les fonctions de base radiales (*Radial Basis Functions* - RBF) sont connues dans la littérature pour offrir un cadre formel permettant l'obtention d'estimations de qualité (Powell, 1987). Des approches **non linéaires**, comme par exemple les réseaux de neurones, peuvent également être employées à cette fin, comme par exemple dans (Dutech, 2012). Dans ce cas de figure, la fonction de qualité approximée a pour définition :

$$\hat{Q}_\theta(s_t, a_t) = f_\theta(s_t, a_t) \tag{3.21}$$

où f est une fonction non-linéaire sur les paramètres.

L'avantage de ces approches sur celles linéaires est qu'elles sont moins sensibles au problème du nombre de dimensions considérées. En effet, elles sont capables de représenter une quantité équivalente d'information avec beaucoup moins de paramètres à estimer (ce qui peut sensiblement faciliter l'apprentissage). Le problème est que la convergence d'une telle approximation n'est pas mathématiquement assurée (Tsitsiklis et Van Roy, 1997). Une comparaison de ces deux types de paramétrisation sur la problématique du dialogue a été faite dans (Daubigney et al., 2012).

Certaines approches dans la littérature font également l'usage de représentations non-paramétriques de la fonction de valeur. C'est le cas notamment de la méthode des différences temporelles par processus Gaussien (*Gaussian Process Temporal Differences* - GPTD) (Engel et al., 2003) qui a notamment été appliquée dans le cadre du dialogue (Gašić et al., 2010; Gašić et al., 2011). Cette approche représente la fonction de valeur par un dictionnaire de points où cette dernière est quantifiée et l'utilisation de noyaux pour déterminer sa valeur en n'importe quel point de l'espace selon une modélisation par un processus gaussien. En utilisant des noyaux adéquats, ce type d'approche peut également faciliter le transfert de la politique entre les domaines (Gašic et al., 2013).

Enfin des travaux ont également envisagé la recherche directe dans l'espace des politiques (c'est à dire sans modélisation de la fonction de valeur). C'est le cas de (Jurcicek et al., 2010) où la politique est représentée par des paramètres qui sont estimés par descente de gradient.

3.4.4 Paradigme de l'état de l'information caché (HIS)

Dans cette section, nous détaillons tout particulièrement le paradigme POMDP HIS (Young et al., 2010) et les choix faits dans ce modèle pour en réduire la complexité. Dans le chapitre 6 nous présenterons une extension de ce même paradigme pour prendre en compte l'information située.

Ce modèle propose la combinaison de 4 procédés de simplification :

— la décomposition de l'état de dialogue (voir l'équation 3.18) ;
— le regroupement dynamique des buts utilisateurs en partitions ;
— l'exploitation de versions approximées des modèles mis en jeu dans la mise à jour de l'état de croyance (voir l'équation 3.19) ;
— la projection du MDP continu maître dans une version résumée (espace d'état et d'action) conjointement à l'emploi d'heuristiques pour appliquer en retour dans l'espace maître les décisions prises dans l'espace résumé.

Les principaux mécanisme de ce paradigme sont illustrés dans la figure 3.5.

À chaque tour le modèle prend en entrée l'action système précédente et les nouvelles hypothèses sémantiques faites par les modules de compréhension. Ces dernières sont représentées sous la forme d'une liste des N-meilleures hypothèses sémantiques associées à leur score de confiance respectif.

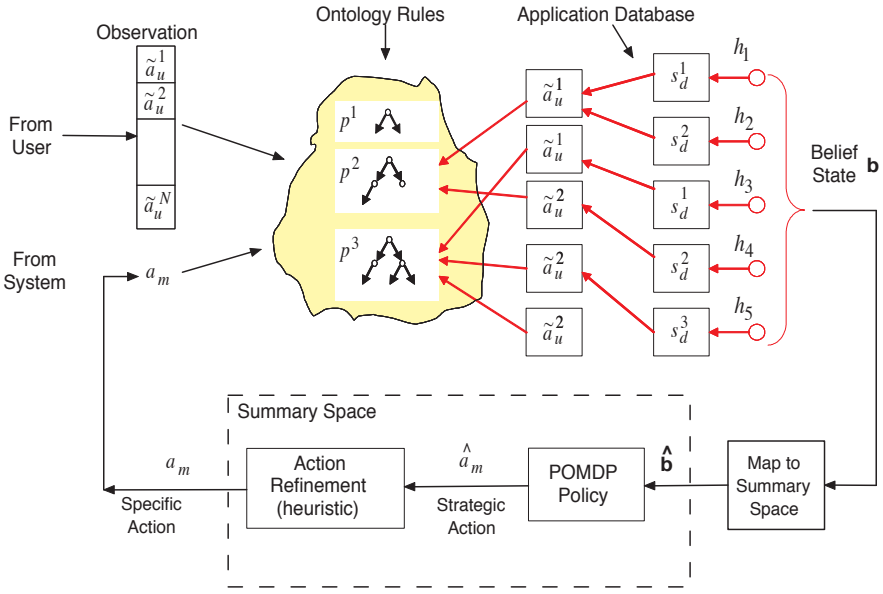


FIGURE 3.5 – *Le paradigme HIS - Extrait de (Young et al., 2010))*

Partitionnement dynamique de l'espace d'état de croyance

L'idée de la notion de partition vient du constat qu'à chaque tour du dialogue de nombreux états sous-jacents du modèle POMDP partagent le même niveau de croyance car aucune des observations faites par le système jusqu'alors ne conduit à les distinguer. Par exemple, si le système perçoit l'acte de dialogue utilisateur *inform(food=french)*, il peut en toute légitimité porter plus de crédit au fait que l'utilisateur ait bel et bien parlé de nourriture française. Rien ne lui permet par contre, sans indices supplémentaires (ou connaissances a priori), de faire une distinction entre tous les autres types de nourriture (italienne, chinoise, etc.). Suivant cette logique, le modèle HIS propose de regrouper l'ensemble de tous les états en classes d'équivalences (états indistinguables au regard des observations) pour simplifier le calcul de la mise à jour de l'état de croyance.

Ainsi donc, à un instant t, l'ensemble des buts utilisateur envisageables est représenté sous la forme d'un ensemble de partitions $p_t \in P$ regroupant les buts équivalents. Chaque partition partage ainsi la même valeur de croyance $b_t(p_t)$. Mais des buts ayant la même valeur de croyance ne font pas forcement partie de la même partition. La combinaison d'une partition, d'une action utilisateur et d'un historique du dialogue (état d'ancrage sur les différents concepts, etc.) forme une hypothèse.

Suivant ce formalisme, de même qu'en supposant que le but utilisateur reste assez stable durant l'interaction (soumis uniquement à des changements explicites, par exemple la détection d'un acte de dialogue utilisateur $reqalts()$) l'équation 3.19 peut encore être simplifiée en :

$$b_{t+1}(p_{t+1}, u_{t+1}, h_{t+1}) = \eta . \underbrace{P(o_{t+1}|u_{t+1})}_{\substack{\text{modèle} \\ \text{d'observation}}} \underbrace{P(u_{t+1}|p_{t+1}, a_t)}_{\substack{\text{modèle d'action} \\ \text{utilisateur}}} \sum_{h_t} \underbrace{P(h_{t+1}|p_{t+1}, u_{t+1}, h_t, a_t)}_{\substack{\text{modèle d'historique} \\ \text{du dialogue}}} \underbrace{P(p_{t+1}|p_t) b_t(p_t, h_t)}_{\substack{\text{raffinement de la} \\ \text{croyance}}} \quad (3.22)$$

où p_t est une partition et le terme $P(p_{t+1}|p_t)$ représente la probabilité que la partition p_t soit décomposée en deux sous-partitions $p_t \rightarrow \{p_{t+1}, p_t - p_{t+1}\}$.

Dans ce contexte, nous pouvons donner une définition plus précise des différents modèles employés.

Raffinement de la croyance dans ce paradigme, un but utilisateur est représenté sous la forme d'une structure arborescente similaire à celle représentée sur la figure 3.6. Cette dernière est composée de classes abstraites, d'instances (subtypes), de concepts (ou classes terminales) et de valeurs atomiques. Les classes abstraites sont des structures qui regroupent d'autres classes abstraites ou concepts liés. Les concepts sont des classes terminales associés à des valeurs atomiques. Les instances correspondent quant à elles aux différentes variantes d'une même classe abstraite et conditionnent les concepts et les classes abstraites qui y sont effectivement associés.

L'espace de tous les buts possibles (tous les arbres possibles) est défini au travers d'une ontologie du domaine, dont un exemple est donné dans le tableau 3.1. Dans

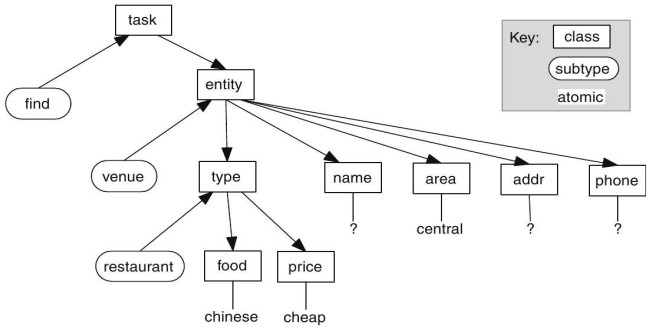


FIGURE 3.6 – *Exemple de partition pour le but utilisateur « trouver un restaurant chinois bon marché en centre ville » - Extrait de (Young et al., 2010) .*

task	$\rightarrow$	find(entity)	{0.4}
entity	$\rightarrow$	venue(name, type, area)	{0.8}
venue	$\rightarrow$	bar(drinks, music)	{0.4}
venue	$\rightarrow$	restaurant(food, pricerange)	{0.3}
area	=	(south | north | east | west | central)	
food	=	(french | italian | chinese | ...)	
...			

TABLE 3.1 – *Exemple d'ontologie du domaine telle qu'exploitée dans le paradigme HIS.*

cette ontologie sont listées toutes les instances possibles pour chaque classe abstraite (ligne avec le symbole $\rightarrow$). Comme on peut le voir, la classe abstraite *venue* peut se décliner sous les deux formes *bar* et *restaurant* selon une certaine probabilité a priori (donnée entre crochet). Selon l'instance effectivement mentionnée par l'utilisateur au cours de l'interaction on peut voir que ce ne sont pas les mêmes concepts qui vont être mis en jeu, par exemple le concept *food* est ici spécifique à l'instance *restaurant* de la classe abstraite *venue*. De même dans cette ontologie sont identifiées toutes les valeurs atomiques pouvant être assignées à un même concept (ligne avec le =).

Un exemple du processus permettant de maintenir des partitions sur ces buts est donné sur la figure 3.7. À l'état initial, tous les buts possibles sont regroupés dans une seule et même partition p_0 avec $b_0(p_0) = 1$ symbolisée par le nœud *task* dans la figure 3.7. Au fil de l'avancement du dialogue, les partitions vont se diviser en partitions de plus en plus fines (identifiant des états de plus en plus précis), et ce en exploitant à la fois l'ontologie du domaine, la dernière action du système et les dernières observations en provenance des modules de compréhension. Cette division en sous-partitions se fera de pair avec une redistribution de la masse de probabilité attribuée aux différentes partitions sur les partitions enfants ainsi créées. Le mécanisme mis en œuvre pour assigner ces probabilités correspond à celui du raffinement de la croyance.

La limite principale de cette approche est que le nombre de partitions maintenues va croître exponentiellement avec le nombre de tours de dialogue considérés. C'est pourquoi des techniques d'élagage sont généralement adoptées pour garantir une mise à

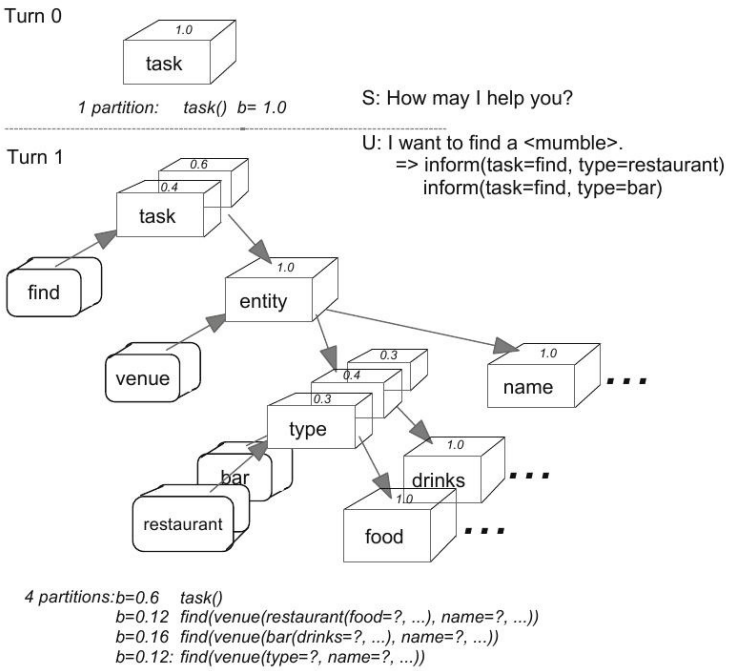


FIGURE 3.7 – *Mécanisme de raffinement successif des partitions - Extrait de (Young et al., 2010)*

jour en des temps raisonnables. Dans (Young et al., 2010) les partitions de faibles probabilités sont recombinées avec leur parent. Une méthode utilisant la notion de complément, au sens ensembliste du terme, a été proposée dans (Gašić et Young, 2011) pour pouvoir gérer des dialogues de longueur arbitraire.

Modèle d'observation les scores associés à la liste des N-meilleures hypothèse issue de la chaîne de compréhension sont ici exploités pour faire une approximation du modèle d'observation :

$$P(o_{t+1}|u_{t+1} = \tilde{u}^i_{t+1}) \approx k^0.p_i \tag{3.23}$$

où k^0 est une constante, $\tilde{u}^i_{t+1}$ est la $i^{\text{ème}}$ hypothèse de compréhension et p_i son score.

Modèle d'action utilisateur le modèle d'action utilisateur employé est le suivant :

$$P(u_{t+1}|p_{t+1}, a_t) \approx \underbrace{P(\mathcal{T}(u_{t+1})|\mathcal{T}(a_t))}_{\text{modèle bigramme}} . \underbrace{P(\mathcal{M}(u_{t+1})|p_{t+1}, a_t)}_{\substack{\text{modèle de} \\ \text{correspondance}}} \tag{3.24}$$

où $\mathcal{T}(.)$ correspond au type de l'acte de dialogue (*acttype*), $\mathcal{M}$ permet de définir si oui ou non l'acte de dialogue est pertinent au regard de la partition p_{t+1} et de la dernière action système a_t. Cette fonction repose exclusivement sur des heuristiques expertes qui lui permettent de déterminer de façon binaire s'il y a correspondance entre ces informations. Par exemple, si la nouvelle partition modélise un but dans lequel apparait

une référence à de la nourriture chinoise et que l'action utilisateur parle de nourriture française on peut considérer que l'acte utilisateur ne correspond pas.

Le modèle bigramme ainsi employé permet d'intégrer dans le modèle la notion de paires adjacentes (Schegloff et Sacks, 1973). Par exemple, le fait qu'une question est souvent suivie d'une réponse. Ce modèle est généralement appris sur des données de dialogue. Cependant il peut être réutilisé d'une tâche à l'autre si le concepteur estime qu'il n'y a pas trop de variation.

Le modèle de correspondance retourne des probabilités optimisées empiriquement sur l'état binaire de (non-)correspondance déterminé par la fonction $\mathcal{M}$.

Modèle d'historique du dialogue le modèle d'historique du dialogue cherche à déterminer le nouvel état de chaque nœud terminal de la partition en considérant la dernière action utilisateur en exploitant un modèle d'ancrage (Traum, 1999). Les états d'ancrage considérés sont listés dans le tableau 3.2 et les transitions entre ces états sont modélisées par une machine à états finis dont les transitions dépendent des actions système et utilisateur. Il est important de noter que pour un nœud terminal d'une partition, le système peut maintenir plusieurs états d'ancrage. En effet chaque nœud peut avoir un jeu d'historiques possibles qui ont conduit à l'établissement de la partition courante.

La probabilité actuellement employée est obtenue de façon déterministe. Ainsi, si l'on estime que le nouvel historique h_{t+1} est conforme au nouveau but p_{t+1} (par exemple en constatant qu'un élément du but a été confirmé par l'utilisateur), alors $P(h_{t+1}|p_{t+1}, u_{t+1}, h_t, a_t) \approx 1$ sinon $P(h_{t+1}|p_{t+1}, u_{t+1}, h_t, a_t) \approx 0$

États	**Significations**
Init	État initial
UReq	La valeur de l'attribut a été demandée par l'utilisateur
UInfo	La valeur de l'attribut a été donnée par l'utilisateur
SInfo	La valeur de l'attribut a été donnée par le système
SQry	La valeur de l'attribut a été demandée par le système
Deny	Valeur de l'attribut a été niée
Grnd	Valeur de l'attribut a été validée (considérée comme étant ancrée)

TABLE 3.2 – *Liste des états du modèle d'ancrage.*

Résumés des espaces d'état de croyance et d'action

Dans HIS, une version résumée des espace de croyance (hypothèses) et d'action est considérée pour permettre aux algorithmes de RL d'être plus efficaces.

Espace d'état résumé il est composé de cinq variables qui reprennent en partie les informations présentes dans l'état de croyance sous-jacent (N-meilleures hypothèses d'état du dialogue). C'est à partir de cet espace de taille raisonnable que le système

pourra prendre ses décisions (qu'elles soient obtenues par règles expertes ou par apprentissage). Les deux premières variables sont respectivement les scores de confiance [8] (valeur réelle entre zéro et un) de la première et de la seconde hypothèse sur l'état du dialogue, et notées **b(hyp1)** et **b(hyp2)**. Les trois autres variables considérées seront discrètes et sont extraites uniquement de la meilleure hypothèse. La première d'entre elles, **p-status**, contient une estimation du rapport à la base de données du but (partition) associée à la meilleure hypothèse (voir tableau 3.3). La seconde variable discrète, **last-uact**, contient une estimation d'une dernier type d'acte de dialogue employé par l'utilisateur. Il est à noter que les différents types d'actes sont dépendants de la tâche visée. En annexe le tableau A.1 donne un exemple concret des actes de dialogue pouvant être employés. Enfin, la dernière variable considérée est le **h-status** qui contient une estimation de l'état d'ancrage (historique) global de la meilleure l'hypothèse (voir tableau 3.4).

États	**Significations**
Initial	la partition n'a pas été mise en relation avec la base de données
LargeGroup	la partition correspond à plus de 3 résultats dans la base de donnés
Group	la partition correspond à 3 résultats ou moins dans la base de données
Unique	la partition correspond à un seul résultat dans la base de données
Unknown	la partition n'a aucun résultat correspondant dans la base de données

TABLE 3.3 – *Liste des états dans lesquels peut se trouver la partition (p-status).*

États	**Significations**
Initial	état dans lequel se trouve l'hypothèse à sa création
Supported	l'hypothèse a au moins un attribut dont la valeur est validée
Offered	l'hypothèse a été proposée à l'utilisateur
Accepted	l'hypothèse a été acceptée par l'utilisateur
Rejected	l'hypothèse a au moins un attribut dont la valeur a été niée par l'utilisateur
Completed	l'hypothèse correspond à un but utilisateur qui semble résolu

TABLE 3.4 – *Liste des états dans lesquels peut se trouver l'hypothèse (h-status).*

Espace d'action résumé à chaque tour du dialogue, la politique d'interaction employée doit choisir une action sur les 11 actions résumées (ou meta-actions) proposées (voir tableau 3.5). Le passage de l'action ainsi sélectionnée à celle qui va se retrouver effectivement en sortie du DM (action maître) se fait en appliquant un ensemble d'heuristiques expertes sur les informations contenues dans la ou les meilleures hypothèses de l'état du dialogue. Par exemple, si l'action *Offer* est prise et que la meilleure hypothèse ne correspond qu'à un seul établissement dans la base de données, alors le système pourra créer un acte de dialogue de type *inform* contenant les informations le concernant (par exemple son nom). C'est notamment à ce niveau que le concepteur du système peut en partie répondre au problème de complétude de la solution, en empêchant la réalisation d'actions que l'on sait inutiles (par exemple saluer l'utilisateur en

8. Il est à noter que l'usage que l'on pourra faire de ces scores dépendra intimement de leur fiabilité, autrement dit des modèles employés.

plein milieu de l'interaction) en se rabattant sur l'application du prochain choix de la politique.

Actions résumées	**Significations**	**Exemple d'actions maîtres associées**
Greet	Le système accueille l'utilisateur	hello()
Request	Le système demande la valeur d'un attribut	request(area)
Inform	Le système donne une information	inform(phone=728-64-32)
Confirm	Le système demande explicitement confirmation sur la valeur d'un ou plusieurs attributs à l'utilisateur	confirm(type=restaurant, pricerange=cheap)
ConfReq	Le système fait une confirmation implicite de la valeur d'un ou plusieurs attributs et demande la valeur d'un autre attribut	confreq(type=restaurant, pricerange=cheap, area)
Select	Le système demande à l'utilisateur de faire un choix entre deux valeurs du même attribut	select(pricerange=expensive, pricerange=cheap)
Offer	Le système propose une entité (ici un établissement correspondant à la demande dans la base de données)	inform(name="Char Sue", type=restaurant, pricerange=cheap)
OfferAlt	Le système propose une autre entité que celle(s) déjà proposée(s)	inform(name="Peking", type=restaurant, pricerange=cheap)
QueryMore	Le système demande à l'utilisateur si il n'a pas une autre chose qu'il veut savoir	reqmore()
UserRepeat	Le système demande à l'utilisateur de répéter	repeat()
Bye	Le système clôt le dialogue	bye()

TABLE 3.5 – *Liste des actions résumées.*

Fonction de récompense la fonction récompense immédiate que nous utiliserons dans tous nos travaux repose exclusivement sur deux critères objectifs, à savoir la réussite (ou l'échec) de la tâche et le nombre de tours. Elle est définie pour pénaliser chaque tour de dialogue par une récompense négative de -1. À l'issue de l'interaction, si l'objectif a été atteint (réussite de la tâche utilisateur), une récompense de $+20$ est attribuée au système contre une de 0 le cas échéant.

Algorithme d'apprentissage plusieurs algorithmes de RL ont été employés avec ce paradigme avec succès pour déterminer des politiques d"interaction capables notamment de dépasser en termes de performance des politiques déterministes expertes. Si

l'approche originale faisait état de l'algorithme de *Monte Carlo* (Young et al., 2010; Lefèvre et al., 2009) des approches plus efficaces ont depuis été employées dont GPTD (Gašić et al., 2010) ainsi que celui des différences temporelles de Kalman (*Kalman Temporal Differences* - KTD) (Daubigney et al., 2012). Ces approches ont notamment permis de montrer qu'il était possible d'apprendre en quelques centaines d'interactions une politique (quasi-)optimale contre plusieurs centaines de milliers dans les configurations antérieures.

3.5 Vers l'apprentissage en ligne des politiques

De nombreuses approches ont considéré l'utilisation d'algorithmes RL sur modèle pour résoudre le POMDP dans le cadre du dialogue. C'est notamment le cas dans la première application du modèle à la problématique du dialogue (Roy et al., 2000) qui montre l'avantage de considérer une approche POMDP résolue de façon exacte (grâce à l'algorithme d'amélioration incrémentale proposée dans (Cassandra et al., 1997)) par rapport à un MDP sur une tâche de dialogue très simple où la résolution est possible. Afin d'augmenter la taille des espaces manipulés et donc le champ applicatif, des travaux ont notamment proposé l'emploi de techniques de compression (clustering) des espaces d'états, d'observations et de croyance, comme dans (Lefèvre, 2007) ou encore fait l'usage d'espaces résumés (Pinault et Lefèvre, 2011b; Chinaei et al., 2012). Cependant estimer les dynamiques du modèle représente un coût très important puisqu'il faudra se reposer sur beaucoup de données pour les estimer correctement.

Dans notre perspective de proposer une solution de démarrage à froid d'un nouveau système, nous nous concentrons exclusivement sur les approches RL sans modèle. Selon ce paradigme d'apprentissage il est possible de distinguer trois types d'approche :

— sur corpus (hors-ligne)
— avec des utilisateurs simulés
— avec de vrais utilisateurs

Les approches sur corpus (hors-line) sont d'un grand intérêt quand des données ont déjà été collectées, par exemple par le biais d'un WoZ ou d'une architecture prototypique. Notamment par l'utilisation d'algorithmes hors politiques (généralisation) très efficaces comme LSPI [9] (Li et al., 2009; Pietquin et al., 2011). Cependant, de telles données ne sont pas disponibles pour toutes les tâches (surtout pour de nouvelles applications). De plus leur collecte est un processus coûteux qui nécessite également un grand effort de post-traitements (annotations) pour rendre ces données exploitables par la machine. Un autre problème réside à la potentielle sous-optimalité de la politique mise en œuvre pour effectuer cette collecte. Bien que les techniques d'apprentissage RL hors politique sont capables de dépasser les performances des politiques qu'elles observent, l'exploration de l'espace de recherche sera tout de même contrainte à celle présente dans les données. Or, rien ne peut réellement garantir qu'il soit suffisant en l'état, et cela peut grandement impacter la qualité de la solution optimale déterminée par un apprentissage.

9. Least-Squares Policy Iteration

De ce fait dans nos travaux, nous privilégierons des techniques permettant d'envisager un apprentissage en-ligne (par le biais d'interactions) de la politique. Dans cette optique, nous détaillons par la suite les approches faisant usage d'utilisateurs simulés mais également celles ayant recours à de véritables interactions (vrais utilisateurs) et qui constituent ce vers quoi nous désirons tendre par nos propositions.

3.5.1 Simulation

Le régime d'apprentissage et de test qui est souvent considéré comme le plus simple et le plus efficace consiste à avoir recours à la simulation. En effet, la construction et l'exploitation d'environnements simulés permet d'automatiser l'exploration d'une large couverture de l'espace des dialogues possibles, selon une gamme variée de scénarios tout en offrant la capacité de pouvoir modifier les conditions expérimentales telle que le niveau d'erreurs des modules de compréhension (Watanabe et al., 1998; Ai et Weng, 2008). Ainsi, ces systèmes permettent la conduite d'interactions fictives à même de générer des comportements utilisateur, auxquels le concepteur du système n'a pas forcement pensé (Pietquin et Hastie, 2013). De plus, la simulation a longtemps été considérée comme une étape incontournable pour initier l'optimisation de la stratégie d'interaction en ligne par RL (Schatzmann et al., 2007b; Young et al., 2010; Thomson et Young, 2010) avant de pouvoir procéder à des raffinements en interagissant avec de vrais utilisateurs. Ceci s'explique notamment par le fait que les techniques d'apprentissage employées jusqu'alors avaient des propriétés lentes de convergence (plusieurs milliers d'interactions étaient généralement nécessaires), ce qui proscrivait leur utilisation directe (apprentissage de zéro).

Un simulateur repose sur la définition d'un **modèle utilisateur** (on parlera également d'utilisateur simulé) et d'un **modèle d'erreurs** (on parlera également de simulateur d'erreurs).

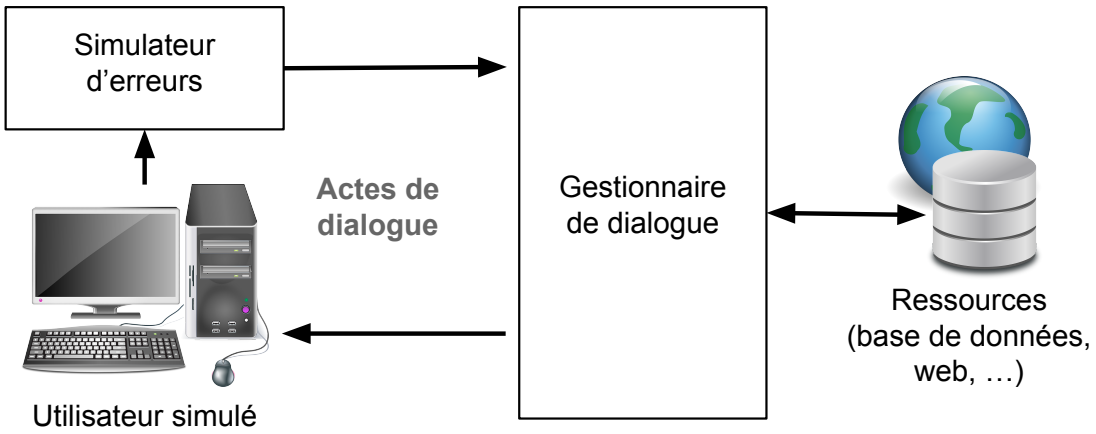


FIGURE 3.8 – *Cycle d'interaction au niveau intentionnel entre un utilisateur simulé et le gestionnaire de dialogue.*

La figure 3.8 illustre le cycle du dialogue tel que généralement mis en œuvre lorsqu'un simulateur est considéré. Il s'agit là d'une représentation usuelle de la simulation qui vise à reproduire le comportement d'un utilisateur au niveau intentionnel (actes de dialogue). En suivant ce paradigme, l'utilisateur simulé génère sa réponse courante sous forme d'actes de dialogue en tenant compte de l'historique de l'interaction. Ces actes sont ensuite transmis à un simulateur d'erreurs qui génère des confusions semblables à celle que ferait la chaîne de compréhension et produit des scores de confiances adéquats. Puis le système répond à l'utilisateur sans avoir recours à la chaîne de génération.

Il à noter qu'il existe également des environnements de simulation qui ne se contente pas du simple niveau intentionnel mais vont également jusqu'au niveau des mots (Schatzmann et al., 2007a; Jung et al., 2009; Khouzaimi et al., 2015). Dans ce cas les actions utilisateur (actes de dialogue) telles que sélectionnées par le simulateur servent en tout premier lieu à générer des formes de surface. Ces dernières sont ensuite traitées par un simulateur d'erreurs capable de reproduire, à l'instar des erreurs que ferait le module ASR, des confusions au niveau des mots. Ceci permet notamment d'employer sur ces formes bruitées le même module SLU qui est employé lorsque le système se trouve face à de vrais utilisateurs.

Le modèle utilisateur

Un modèle vise à reproduire le comportement d'un utilisateur au niveau intentionnel (actes de dialogue). En se plaçant à ce niveau plus abstrait, l'apprentissage en est facilité. Pour ce faire, le modèle utilisateur détermine la distribution sur l'ensemble des réponses utilisateur possibles sachant l'historique de l'interaction (actes de dialogue utilisateur et système), notée :

$$p(u_t|a_t, u_{t-1}, a_{t-1}, u_{t-2}, ...) \tag{3.25}$$

Cette distribution est généralement apprise sur un corpus de dialogue. Ainsi, on retrouve dans la littérature plusieurs techniques pour estimer cette distribution sur la base de données. Parmi les plus anciennes, on peut mentionner l'approche N-grammes (Eckert et al., 1997; Levin et al., 1997) qui consiste à estimer la réponse utilisateur la plus probable compte tenu d'un historique réduit (N=2 ou N=3) . Cependant, une des limitations de cette approche et qu'en pratique N doit être grand pour pouvoir générer des comportements cohérents. Or du fait du manque de données (quelques centaines de dialogues seulement) considérer un N trop grand (N=4, N=5) conduirait nécessairement à des modèles peu fiables.

Une réponse à cette problématique consiste à employer des solutions garantissant le suivi d'un but tout au long de l'interaction et reposant principalement sur la mise en place d'heuristiques de contrôle dans lesquelles certains paramètres décisionnels sont initialisés par expertise, puis éventuellement optimisés à l'aide de données. On pourra donner comme exemple de ce type de techniques celles reposant sur le maintien d'un agenda (Schatzmann et al., 2007b; Keizer et al., 2010). Nous détaillons d'ailleurs tout

particulièrement cette solution dans la section 4.2.2 pour en avoir fait l'usage dans nos travaux préliminaires. Si en pratique de telles techniques obtiennent de bonnes performances elles nécessitent généralement un gros travail de raffinement des heuristiques sur lesquelles elles reposent pour générer des comportements cohérents.

D'autres approches utilisent les HMM (Pietquin, 2004; Cuayáhuitl et al., 2005) ou encore les réseaux bayésiens (Pietquin et Dutoit, 2006; Rossignol et al., 2011) pour garantir un cadre formel incorporant explicitement le but utilisateur (structure), modélisant des dépendances conditionnelles riches et ayant de nombreux paramètres estimables via des techniques d'apprentissage. Certains travaux ont également envisagé l'utilisation de méthodes discriminantes telles que les CRF (Jung et al., 2009) qui présentent l'avantage de pouvoir modéliser plus efficacement de grandes séquences d'observations.

Une autre technique consiste à développer un simulateur défini de façon symétrique au DM, à savoir via un modèle de type MDP ou POMDP. Dans cette vision du problème le DM et le simulateur peuvent interagir ensemble tout en optimisant leur politique respective via l'observation de leurs propres récompenses. Le principal frein à cette idée réside dans la définition d'une fonction de récompense immédiate pour permettre l'optimisation de la politique d'interaction de l'utilisateur. Parmi les solutions envisagées, des techniques telles que celles de l'IRL (Ng et al., 2000) ont été proposées dans (Chandramohan et al., 2011) pour estimer cette fonction sur la base d'un corpus d'interactions.

La simulation des erreurs

Comme dans des conditions d'interaction réelles, le système de dialogue n'a à sa disposition que des observations bruitées des vraies réponses de l'utilisateur, un modèle d'erreur est également employé pour améliorer le réalisme de l'outil de simulation.

Ce dernier doit pouvoir être à même de reproduire les erreurs faites par la chaîne de compréhension (ASR et SLU) sur l'énoncé tel qu'émis par l'utilisateur tout en étant capable de produire des scores de confiances pertinents. En effet, les scores de confiances attribués à la distribution complète des hypothèses de compréhension ont un rôle primordial dans la modélisation de l'état de croyance comme le montre l'étude dans (Thomson et al., 2008). Il peut également être intéressant que ledit module soit paramétrable, permettant par exemple de pouvoir jouer sur le niveau de performance simulé de la chaîne de compréhension. Ceci permet de pouvoir faciliter la tenue d'évaluations contrastives sur des cas d'utilisations limites (pour par exemple étudier la tolérance aux bruits).

Pour ce faire, plusieurs techniques ont été étudiées dans la littérature. La plupart d'entre elles se sont concentrées sur la modélisation des erreurs faites par l'ASR et ce en ne considérant que la première hypothèse de transcription. Certains travaux ont fait l'usage de taux d'erreurs fixes dépendant de la tâche réalisée par le système (reconnaissance de numéro, de date, parole libre) (Pietquin et Renals, 2002), du profil utilisateur (Prommer et al., 2006) ou d'une estimation établie sur des données (Georgila et al., 2005;

Lemon et al., 2006). D'autres approches se sont quant à elle concentrées sur une modélisation plus fine des erreurs au travers de la modélisation de possibles confusions phonétiques (Deng et al., 2003; Pietquin, 2004; Stuttle et al., 2004; Pietquin et Dutoit, 2006; Schatzmann et al., 2007a; Jung et al., 2009) et de l'étude de leurs incidences sur la compréhension (Schatzmann et al., 2007a).

Limites de la simulation

Même si le recours à la simulation présente certains avantages une fois le simulateur en place (facilité de mise en place des expériences), leur utilisation pour l'évaluation pose problème puisque les performances obtenues en simulation ne peuvent être uniquement interprétées que comme le résultat de l'adéquation entre les conditions de simulation et la politique de dialogue, que l'on sait très sensible aux performances du simulateur (Schatzmann et al., 2005; Ai et al., 2007; Pietquin et Hastie, 2013). Afin de s'assurer de l'adéquation des politiques apprises face à de vrais utilisateurs certaines études ont proposé de les tester en conditions réelles (Schatzmann et al., 2007b).

Cependant quand le système doit être développé de zéro, les conditions nécessaires à l'établissement d'un simulateur ne sont pas toujours réunies, du fait du manque de données pour modéliser l'utilisateur et/ou les erreurs (d'autant plus si la nature de l'interaction est complexe - cas d'un système multimodal). Plutôt que d'envisager une collecte de données coûteuses avec WoZ, dans cette thèse nous prenons le parti de dire que dans la mesure où l'apprentissage RL est rendu suffisamment efficace, il peut être intéressant d'envisager un apprentissage direct face à de vrais utilisateurs (éventuellement des utilisateurs moins sensibles aux conditions difficiles - concepteur du système / panel réduit).

3.5.2 Apprendre efficacement face à de vrais utilisateurs

Comme nous l'avons mentionné précédemment, même si le recours à la simulation permet d'automatiser l'apprentissage et de tester des politique d'interaction, leur limitations théoriques (adéquation des comportements en simulation avec ceux observés en interactions réelles) de même que leur besoin en données pour raffiner leur modèle font qu'il peut être intéressant d'avoir recours directement à de véritables interactions (plus proche de l'idée d'origine du RL (Sutton et Barto, 1998)).

Longtemps limité au raffinement des politiques préalablement apprises sur corpus ou en simulation, l'apprentissage de la politique selon ce paradigme présente l'avantage de pouvoir offrir un cadre permettant une optimisation sur les comportements des utilisateurs finaux, voire même capable de s'adapter à des nouveaux usages. Dans ce manuscrit nous soutenons que si cet apprentissage est suffisamment bien mené il peut même se substituer au besoin de développer un simulateur utilisateur ou d'avoir recours à une collecte préalable de données (WoZ, prototype, etc.). Pour ce faire, il conviendra de prendre quelques précautions sur le choix des conditions d'apprentissage, choix que nous décrivons plus en détail ci-après.

Récupération des récompenses de l'environnement

Lorsqu'un apprentissage en ligne est adopté, une des contraintes importantes consiste en la nécessité du système de récupérer des retours (récompenses) pertinents au cours de l'interaction pour optimiser sa politique efficacement. Comme nous l'avons vu précédemment, la plupart des travaux utilisent une fonction de récompense définie grâce à des métriques objectives (nombre de tours, etc.). Mais certaines d'entre elles, comme le critère de réussite de la tâche requiert la coopération de l'utilisateur qui, à la fin de l'interaction doit transmettre cette information à l'agent apprenant pour qu'il puisse raffiner sa politique en fonction.

En condition réelle cependant, les travaux de (Gašić et Young, 2011) ont montré que ces retours utilisateurs pouvaient parfois s'avérer erronés (biaisés) dans certaines conditions. Deux raisons sont avancées pour expliquer ce constat. La première est que les retours utilisateurs ne peuvent pas être considérés comme réellement objectifs. Par exemple, malgré le fait que le système ait réussi de façon objective la tâche utilisateur, ce dernier peut vouloir, consciemment ou non, pénaliser le système pour des comportements incohérents manifestés durant le cours de l'interaction. Une autre explication peut venir du fait que les utilisateurs généralement recrutés à des fins d'expérience ne sont pas réellement concernés par la tâche qu'ils « jouent » (puisqu'elle leur est généralement imposée sans lien avec leur "vraie" vie et leurs besoins).

Une première solution à cette problématique consiste à adopter une fonction de récompense plus diffuse, moins sensible à un retour final erroné. Ceci peut notamment être vu comme une réponse au problème d'affectation temporelle de valeur décrit dans la section 3.2. Parmi les différentes techniques envisagées on peut mentionner le *reward shaping* (Ng et al., 1999), des techniques comme celle de l'IRL (Russell, 1998) pour apprendre une fonction de récompense sur la base de données (Paek et Pieraccini, 2008; Boularias et al., 2010; El Asri et al., 2012) ou encore par régression linéaire (Rieser et Lemon, 2011). Cependant toutes ces techniques supposent que des exemples d'interactions de qualité sont disponibles (capables de couvrir l'espace de recherche effectivement exploré en ligne lors de vraies interactions). Dans nos travaux, nous proposons d'exploiter des signaux de récompenses additionnels « génériques » issus de ressources disponibles dès le démarrage du système telles que les connaissances expertes initiales ou encore les jugements subjectifs émis tout au long de l'interaction par l'utilisateur.

Une autre solution (qui peut être adoptée conjointement avec la première) consiste à faire interagir des utilisateurs avertis (expert, panel d'utilisateurs avancés) dans les premières phases de l'apprentissage (phase durant laquelle le système aura les comportements les plus incohérents) ou encore à susciter l'engagement des utilisateurs finaux en valorisant plus avant les bénéfices potentiels de ce mécanisme sur leur expérience (surtout pour des applications où l'utilisateur peut être amené à interagir à nouveau avec le système).

Exploration efficace

Pour qu'un agent RL améliore sa politique en ligne il a besoin d'explorer suffisamment son espace de recherche pour espérer atteindre la solution optimale. Pour cela il doit tester des actions autres que celles préconisées par sa politique de contrôle courante. Cependant il ne peut se contenter d'explorer car il doit également satisfaire la tâche de façon efficace. Se pose donc alors le problème de choisir entre explorer de nouvelle actions ou exploiter l'estimation courante de sa politique. Ce problème est communément appelé le **dilemme exploration/exploitation**.

Dans la littérature, des techniques à base d'heuristiques sont généralement employées pour répondre à cette problématique. Une des plus simple est la stratégie d'exploration ϵ**-glouton**. Elle consiste à prendre une action aléatoire selon une probabilité ϵ et à exploiter la politique de contrôle le reste du temps. Une extension classique consiste à combiner cette approche avec une technique de type recuit simulé. De cette façon, l'agent commence par explorer son espace avec un ϵ très grand, puis ϵ diminue progressivement au fil du temps de façon proportionnelle à une métrique (par exemple le nombre des tâches réussies) pour permettre à l'agent de progressivement exploiter sa politique.

Cependant, lorsque le système se retrouve face à de vrais utilisateurs, le recours à l'aléatoire est très problématique. En effet, procéder de la sorte peut sensiblement perturber le cours de l'interaction. Il convient donc de donc de considérer des approches plus sûres. Parmi les approches les plus efficaces, se trouve l'approche de *bonus-glouton* proposée initialement dans (Geist et Pietquin, 2011) et dont la formule est rappelée ici :

$$a_t = \underset{a}{\operatorname{argmax}}\, \mu_Q(s_t, a) + \beta \frac{\sigma^2_Q(s_t, a)}{\beta_0 + \sigma^2_Q(s_t, a)} \tag{3.26}$$

où β et β_0 sont deux constantes et $\mu_Q(s_t, a)$ et $\sigma^2_Q(s_t, a)$ sont respectivement la moyenne et la variance de l'estimation courante de la fonction de qualité Q. L'équation 3.26 définit, pour chaque instant t, la politique permettant de sélectionner la réponse du système a_t.

Cette technique présente l'avantage de permettre une exploration plus subtile en se basant sur la moyenne courante estimée pondérée par l'incertitude de la fonction de qualité pour le couple (s, a). En effet, plus l'incertitude sur l'estimation courante de fonction de qualité pour le couple état-action (s_t, a) est grande, plus la variance qui lui est associée est grande. Ainsi, suivant l'équation 3.26, l'attribution de ce bonus β favorise l'exploration des actions dont l'incertitude sur l'estimation de la fonction de valeur est forte. Lorsque l'espace de recherche est suffisamment exploré, l'incertitude décroît et la variance diminue de fait. Le choix de l'action gloutonne (maximisant la moyenne) est alors favorisé.

Compte tenu de son efficacité sur des tâches de référence RL (Geist et Pietquin, 2011) mais aussi sur la problématique particulière du dialogue (Daubigney et al., 2011, 2012), la technique bonus-glouton constitue notre technique d'exploration de référence tout au long de ce manuscrit lorsque nous considérons l'apprentissage en ligne. Cependant,

il est a noter que cette méthode présuppose que l'algorithme RL employé soit capable de nous fournir une estimation sur la variance de ses estimations courantes.

Algorithme efficace par échantillon

Lorsqu'un apprentissage avec de vrais utilisateurs est envisagé, l'efficacité de l'apprentissage est primordiale. Certaines techniques d'apprentissage RL telles que l'approche *Monte-Carlo*, notamment employée dans la version originale de HIS (Young et al., 2010), ne sont en fait applicables que si le système est en mesure d'interagir un grand nombre de fois avec son environnement afin d'estimer les probabilités de transitions et leur apprentissage reste relativement lent. Il est en est de même pour les approches classiques du RL comme SARSA ou Q-learning, qui sont des algorithmes dits du premier ordre (direction de la descente de gradient déterminée à partir de la dérivée). Bien que cela n'est en soi pas une problématique fondamentale dès lors que le système peut interagir avec un environnement simulé, elle l'est pourtant dès lors qu'un apprentissage en ligne avec de vrais utilisateurs est cette fois considéré. C'est pourquoi, dans la littérature, des approches du seconde ordre (direction de la descente de gradient déterminée à partir de la dérivée seconde) dites efficaces en termes d'échantillons sont généralement préférées dans des situations de cette nature.

Des études récentes ont par exemple montré qu'il était possible d'apprendre de zéro une politique de dialogue en quelques centaines d'interactions (Gašić et al., 2010; Sungjin et Eskenazi, 2012; Daubigney et al., 2012) contre plusieurs milliers dans les configurations antérieures (Young et al., 2010; Thomson et Young, 2010). Parmi ces algorithmes efficaces on peut mentionner l'approche GPTD (Engel et al., 2003; Gašić et al., 2010) et plus particulièrement GP-SARSA qui a notamment été utilisée avec succès dans le cadre du dialogue (Gašić et al., 2010; Gašić et al., 2011; Gašic et al., 2013, 2014).

Cette approche modélise la fonction de qualité Q de façon non paramétrique par un processus gaussien de moyenne nulle avec un noyau représentant les corrélations entre l'espace d'état de croyance et l'espace d'action. Le processus gaussien ainsi employé ne se contente pas seulement de l'estimation de la moyenne de la fonction de Q, mais aussi de sa variance, ce qui donne une estimation de l'incertitude de l'approximation. Comme nous l'avons vu précédemment, lorsqu'on considère une approche en ligne, cette estimation de l'incertitude peut être exploitée pour rendre l'exploration plus efficace et ainsi accélérer l'apprentissage (Gašić et al., 2011).

Parallèlement, l'approche KTD (Geist et Pietquin, 2010; Daubigney et al., 2012) a également était proposée dans la littérature. Cette dernière présente de nombreux avantages sur GPTD qui ont notamment motivé son choix dans nos travaux. Plus de détails sont donnés sur cette technique dans la section 3.5.3.

Faire face à la non-stationnarité

Lorsque le système est dans les conditions d'apprentissage en ligne, il est plus sujet à des phénomènes de non-stationnarité. Il s'agit là d'une problématique importante en

RL. En effet, dans de nombreuses situations d'apprentissage, l'adaptation à des environnements non stationnaires (dont les dynamiques changent au court du temps) est une caractéristique souhaitée.

Pour le DM, cela peut se traduire par le fait que des utilisateurs avec différents niveaux d'expertise (de novice à avancé par exemple) et aux caractéristiques propres (patience, qualité d'audition, etc.) peuvent interagir avec le système. Cela peut aussi être le cas lorsque le système aura fait un premier apprentissage de sa politique sur simulateur et qu'ils souhaiteront ensuite la raffiner en ligne. De plus, le système doit également savoir faire face à des phénomènes de co-adaptation, où l'Homme, symétriquement à la Machine, va progressivement s'adapter aux comportements de l'agent décisionnel pour contourner ses limitations et mieux exploiter ses capacités (Chandramohan et al., 2014).

Ainsi, le système apprenant doit être en mesure de faire face à un large éventail de comportements dont les dynamiques peuvent changer au cours du temps (pour peu que ces changements soient stationnaires par morceaux). Il est à noter qu'une autre source de non-stationnarité survient lorsque le régime d'itération sur politique (Sutton et Barto, 1998) est adopté (ce qui est souvent le cas dans le cas de l'apprentissage en ligne). En effet dans ce paradigme d'apprentissage la fonction de valeur change en même temps que la politique (puisque c'est l'estimation courante de la fonction de valeur qui est utilisée pour le contrôle) ce qui rend le processus non-stationnaire.

Quel que soit le type de non-stationnarité (liée à l'environnement et/ou à la méthode d'optimisation) une solution de poursuite (*tracking* en anglais) de la solution optimale semble préférable à utiliser une solution plus classique permettant de converger vers elle (du fait de la vision dynamique de la politique qu'elle introduit). Une discussion plus détaillée sur les avantages du *tracking* par rapport à la convergence (et ce même pour des situations où l'environnement présente des dynamiques stationnaires) peut être trouvée dans (Sutton et al., 2007). La plupart des algorithmes de RL supposent la stationnarité du problème (dynamiques du système stationnaires) et visent à converger vers une solution fixe. Peu de tentatives pour traiter la non-stationnarité peuvent être trouvées dans la littérature. On pourra citer l'algorithme Dyna-Q (Sutton et Barto, 1998) combinant des méthodes de RL et de planification ou encore l'approche KTD (Geist et Pietquin, 2010). Généralement cela passe par l'intégration de mécanismes permettant la remise en cause des observations passées (capacité d'oubli) pour s'adapter aux nouvelles dynamiques sur des fenêtres temporelles glissantes.

3.5.3 Cadre des différences temporelles de Kalman (KTD)

Le cadre formel des différences temporelles de Kalman, KTD (Geist et Pietquin, 2010), applique le mécanisme du filtre de Kalman (Kalman, 1960) au formalisme des différences temporelles pour estimer les paramètres utilisés pour représenter la fonction de qualité (vue comme des variables cachées), à partir d'une fenêtre d'observation sur les récompenses collectées. Le principe de base est de tenter une prédiction de la fonction de récompense.

Dans nos travaux nous nous intéresserons principalement à une représentation linéaire et paramétrique de la fonction de qualité, telle que :

$$\hat{Q}_\theta = \theta^T \phi(s,a) \tag{3.27}$$

où le $\phi(s,a)$ est un vecteur de caractéristiques défini par un ensemble de n fonctions de base conçues par un expert et $\theta \in \Re^n$ le vecteur de paramètres à estimer. Il est à noter qu'une représentation non linéaire de la fonction Q est tout à fait envisageable dans ce formalisme.

Les composantes du vecteur θ sont les variables cachées du modèle et θ est vu comme un vecteur aléatoire. Ce vecteur de paramètres évolue en suivant une marche aléatoire définie par l'équation d'évolution suivante :

$$\theta_t = \theta_{t-1} + v_t \tag{3.28}$$

où v_t est un bruit blanc de matrice de covariance P_{v_t}. Ce bruit permet notamment de tenir compte de la possible non-stationnarité de la fonction de qualité. Cette capacité du modèle sera étudiée dans le prochain chapitre.

Les observations considérées correspondent aux récompenses de l'environnement et sont liées au vecteur de paramètres cachés par l'une des équations d'échantillonnage de Bellman $g_t(\theta_t)$ définie ci-dessous selon le schéma RL employé (à savoir celle d'évaluation pour l'apprentissage sur politique ou d'optimalité pour celui hors politique) :

$$g_t(\theta_t) = \begin{cases} \hat{Q}_{\theta_t}(s_t,a_t) - \gamma \hat{Q}_{\theta_t}(s_{t+1},a_{t+1}) & \text{(evaluation)} \\ \hat{Q}_{\theta_t}(s_t,a_t) - \gamma \max\limits_a \hat{Q}_{\theta_t}(s_{t+1},a) & \text{(optimalité)} \end{cases} \tag{3.29}$$

Dans ce formalisme, on suppose que les récompenses suivent l'équation d'observation suivante :

$$r_t = g_t(\theta_t) + n_t \tag{3.30}$$

où un bruit blanc n_t avec matrice de covariance P_{n_t} est également considéré .

Entre autres, deux algorithmes peuvent être définis :
— KTD-SARSA qui utilise l' équation d'évaluation de Bellman (algorithme sur politique) ;
— KTD-Q qui exploite celle de l'optimalité (algorithme hors politique).

Dans (Geist et Pietquin, 2010; Daubigney et al., 2012), les auteurs ont montré que KTD possède de nombreuses propriétés qui permettent de répondre à la majorité des critères utiles pour envisager l'apprentissage de la politique d'un DM. Comme nous l'avons déjà mentionné, c'est un algorithme efficace par échantillon du fait qu'il est basé sur l'estimation des moments du second ordre. De plus, il permet de réaliser à la fois des apprentissages sur ou hors politique et en ligne ou hors ligne. Il est également à même de produire une mesure sur l'incertitude de ses estimations, ce qui peut notamment être exploité pour établir une stratégie d'exploration plus efficace (voir section 3.5.2). Il peut fonctionner avec une représentation paramétrique de la fonction de qualité autant linéaire que non-linéaire. Enfin, il a été remarqué pour ses capacités de

suivi de la politique optimale, et d'adaptation à un changement de dynamiques de l'environnement, dans (Geist et al., 2009)

Enfin, dans (Daubigney et al., 2012), les deux algorithmes KTD-Q et KTD-SARSA ont été favorablement comparés à différents algorithmes de l'état de l'art mais qui ne disposent que d'un sous-ensemble des propriétés mentionnées ci-dessus, tels que Q-learning (Watkins et Dayan, 1992) , LSPI (Lagoudakis et Parr, 2003) ou GP-SARSA (Engel et al., 2003). Pour plus de détails sur cette technique et ses performances le lecteur peut se référer à (Geist et Pietquin, 2010; Daubigney et al., 2012) .

3.6 Bilan

Ce chapitre nous a permis de dresser un état de l'art sur les techniques de RL employées pour optimiser les comportements du DM grâce aux données. Nous avons particulièrement détaillé la modélisation du problème de la gestion de l'interaction par POMDP, qui nous permet de pouvoir modéliser explicitement l'incertitude à laquelle va devoir faire face le système de dialogue.

Nous avons discuté des différentes techniques d'apprentissage et également de leurs limites. Nous avons également avancé les raisons qui nous poussent à vouloir considérer un apprentissage en ligne de la politique de dialogue par l'intermédiaire de KTD. De plus nous avons détaillé le paradigme HIS qui va être employé dans le suite de cette étude.

La partie suivante commencera par présenter des techniques nouvelles visant à améliorer l'apprentissage en ligne de zéro de la politique par l'introduction soit de connaissances expertes, soit d'évaluations subjectives capables de s'adapter au profil de l'utilisateur et qui peuvent être employées pour accélérer l'apprentissage et l'adaptation en ligne d'une politique de dialogue.

Deuxième partie

Contributions et cadres applicatifs

Dans cette partie nous présentons nos différentes propositions pour établir un cadre d'apprentissage permettant d'envisager un apprentissage en ligne efficace et de « zéro » du système de dialogue ainsi qu'une application concrète sur une plateforme robotique.

Tout d'abord, dans le chapitre 4, nous nous attachons au problème d'optimisation de la politique du dialogue en situation d'interaction directe avec de vrais utilisateurs. Ce chapitre est pour nous un moyen de présenter les différentes approches explorées pour rendre l'apprentissage de « zéro » de la politique plus efficace, mais aussi plus réactif face à un changement de dynamique de la part de l'environnement (adaptation à un nouveau profil utilisateur).

Ensuite, dans le chapitre 5, nous présentons un moyen de limiter les efforts de conception du module compréhension par l'exploitation d'un espace sémantique appris sur une grande quantité de données dans le but de généraliser les connaissances initiales sur la tâche de dialogue (domaine précis). Une procédure d'adaptation en ligne de ce modèle sera ensuite proposée, de même qu'un moyen d'effectuer son optimisation en ligne afin d'en améliorer efficacement les performances tout en limitant les efforts de supervision consentis par les utilisateurs. Nous profitons également de ce chapitre pour étendre cette solution à un cadre supervisé plus standard, à savoir celui d'un modèle CRF.

Enfin, le chapitre 6 est consacré au système de dialogue multimodal réalisé dans le cadre du projet *MaRDi* pour répondre à une tâche de robotique domestique. Ainsi, nous présentons les solutions retenues pour intégrer les informations multimodales et contextuelles dans la chaîne du dialogue ainsi que pour rendre possible l'apprentissage de la politique d'interaction de par sa mise en situation face à de vrais utilisateurs. De plus, nous proposons dans ce chapitre une extension au paradigme HIS permettant de prendre en compte, au travers de l'introduction de raisonnements cognitifs haut niveau, de l'aspect situé de l'interaction Homme-Robot au sein du processus d'optimisation de la politique d'interaction, et ce afin d'en améliorer la qualité globale. Il est à noter que nous profitons également de ce chapitre pour évaluer une partie des propositions faites dans les chapitres précédents sur un cas concret de développement de « zéro » d'un système de dialogue.

Chapitre 4

Apprentissage par renforcement en-ligne de « zéro » de la politique de dialogue

Sommaire

Envisager un apprentissage en-ligne de « zéro » de la politique de dialogue est un sujet d'étude assez récent dans la littérature car il était jusqu'à peu considéré comme non envisageable par la communauté (du fait du temps de convergence trop grand des algorithmes de RL employés). En effet, comme nous l'avons déjà précisé dans la section 3.5.2, de récentes études ont montré qu'il était désormais possible d'apprendre une politique de dialogue (quasi-)optimale en quelques centaines d'interactions (Gašić et al., 2010; Sungjin et Eskenazi, 2012; Daubigney et al., 2012) contre plusieurs centaines de milliers dans les configurations antérieures (Young et al., 2010; Thomson et Young,

2010). Cependant envisager un tel apprentissage ne se fait pas sans quelques difficultés. Dans ce chapitre nous nous intéresserons particulièrement à deux problématiques rencontrées lors de sa mise en œuvre :

La première, que nous noterons **P1**, est liée au très bas niveau performance dans lequel se trouve le système à l'état initial (démarrage à froid). Afin de rendre possible l'interaction avec des utilisateurs finaux il faut garantir un état de fonctionnement minimal du système (on parlera aussi de seuil d'acceptabilité). Or, en début d'apprentissage le système ne peut s'appuyer sur ses propres connaissances pour résoudre les tâches utilisateurs et pour en acquérir il doit procéder à une exploration intense de l'espace de recherche liant les états du dialogue et les actions possibles. Dans cette phase initiale, dont nous nous attacherons à essayer d'en réduire la durée dans cette étude, seul un utilisateur expert (ou un panel d'utilisateurs dédié) est à même d'accepter les comportements (souvent incohérents) du système.

La seconde problématique, notée **P2** par la suite, est quant à elle liée aux possibles changements de dynamiques de l'environnement intervenant au cours du cycle d'apprentissage de la politique du DM (voir problème de non-stationnarité dans la section 3.5.2). En effet, le fait de considérer un apprentissage en-ligne efficace sur le long terme permet d'envisager l'idée d'un système capable de raffiner sa politique au fil de ses interactions. Dans ces conditions, il n'est pas exclu que des utilisateurs avec différents profils (patience, qualité d'audition, etc.), pour lesquels la politique optimale n'est pas la même, soient amenés à interagir avec le système tout au long de son cycle d'apprentissage (et ce sur plusieurs dialogues). De même, si un utilisateur particulier utilise fréquemment le système de dialogue, une situation de co-adaptation peut apparaître (Chandramohan et al., 2014). Dans celle-ci l'homme, symétriquement à la machine, va progressivement adapter ses comportements pour tenter d'être plus efficace. Ainsi, l'agent apprenant doit être doté de capacités lui permettant de s'adapter efficacement au cours du temps à ces changements.

Outre l'adoption de l'algorithme RL KTD (Geist et Pietquin, 2010; Daubigney et al., 2012) pour à la fois permettre l'apprentissage face à de vrais utilisateurs et être tolérant aux changements de dynamiques de l'environnement, nous proposons dans cette étude l'emploi d'informations additionnelles pour influer sur le déroulement de l'apprentissage afin de le rendre plus efficace. Pour cela, nous considérerons en premier lieu des éléments issus de l'expertise du domaine afin de répondre à **P1**, puis nous proposerons l'introduction de critères subjectifs (caractérisant le profil utilisateur) pour pouvoir poursuivre plus efficacement la politique optimale et ainsi répondre à **P2**.

4.1 Exploiter les connaissances expertes pour faciliter l'apprentissage

Lors de la phase de conception d'un système, il n'est pas rare d'avoir en tout premier lieu établi un ensemble de spécifications sur les comportements attendus du système. Ces dernières prennent notamment la forme de règles expertes représentant la straté-

gie (politique) du gestionnaire de dialogue. Dans l'industrie, ces règles sont confrontées à de vrais utilisateurs puis manuellement ajustées jusqu'à l'obtention d'un niveau de satisfaction satisfaisant (Pieraccini et al., 2009). De façon analogue, nous comptons exploiter ces connaissances expertes pour guider la phase initiale de l'apprentissage par RL de la politique de dialogue, cependant à la différence du processus décrit juste avant elles n'auront pas vocation à évoluer par intervention manuelle, puisque l'amélioration de la politique jouera ce rôle d'ajustement automatiquement.

L'idée d'exploiter des règles expertes pour assister l'apprentissage RL de la politique du DM n'est pas nouvelle en soi. Par exemple, dans (Williams, 2008b) l'auteur propose l'utilisation d'un contrôleur expert afin de pré-sélectionner un ensemble d'actions possibles à chaque tour de parole, la « meilleure » étant ensuite déterminée par l'agent apprenant. Cette solution a la capacité d'accélérer sensiblement l'apprentissage en restreignant localement l'espace de recherche tout en plaçant des précautions utiles dans l'industrie pour garantir la complétude de la solution proposée (voir la section 2.2.2). Cependant contrairement à notre proposition, cette technique requiert l'établissement préalable de règles de qualité permettant de réduire suffisamment l'espace de recherche localement (sur un état) pour effectivement accélérer l'apprentissage sans pour autant exclure l'action optimale (point qui est difficile à prouver théoriquement). De plus cette méthode ne se pose pas la question de quand retirer ces gardes-fous et dans ce cas quel serait le comportement de la politique apprise ensuite.

Une autre solution envisageable consiste à considérer l'exploitation d'un corpus collecté avec la politique experte (qui là encore serait présupposée complète et capable de mener une interaction de bout en bout) et, à l'instar des travaux présentés dans (Li et al., 2009; Pietquin et al., 2011), d'apprendre une politique par l'intermédiaire d'un algorithme hors ligne et hors politique efficace. Cependant, dans ces conditions, l'obtention d'une politique optimale se ferait uniquement au prix d'une exploration suffisante de l'espace de recherche dans les données, qui dans ce cas précis reposerait essentiellement sur le recours à de l'aléatoire (sous-optimal) et confronterait les utilisateurs précoces à des situations pénibles. Se poserait également à nouveau le problème de quand basculer de ce mode de contrôle « expert », à celui résultant de l'application de la politique apprise par RL. En effet, si ce basculement intervient prématurément les performances du système peuvent chuter de façon drastique.

Ainsi, dans cette thèse, nous avons fait le choix d'étudier deux options alternatives pour introduire dans le système cette expertise a priori du domaine :

— **Option 1** : utiliser ces règles pour guider l'exploration initiale de l'agent apprenant vers des actions localement compatibles. L'objectif sous-jacent étant de répondre au dilemme exploration/exploitation, introduit dans la section 3.5.2, en forçant l'agent apprenant à explorer plus tôt des actions censées conduire à des états favorables tout en lui garantissant un niveau suffisant de liberté pour qu'il apprenne de meilleures solutions.
— **Option 2** : injecter cette connaissance experte dans un signal de renforcement additionnel (sous forme d'une fonction de récompense immédiate supplémentaire). L'objectif étant cette fois de répondre au problème de l'affectation tempo-

raire de la valeur tel qu'évoqué dans la section 3.2, en proposant une fonction de récompense immédiate plus diffuse le long du déroulement du dialogue.

Ces deux options, dont nous allons donner les détails par la suite, sont proposées pour répondre à **P1**. Pour **P2** en revanche, bien qu'il soit également possible de faire évoluer (manuellement) les connaissances expertes ainsi employées, une solution moins rigide est nécessaire. Sachant que l'utilisateur est le principal acteur du changement des dynamiques (nouveau profil, adaptation au système, etc.), tenir compte de son évaluation subjective tout au long de l'interaction pourrait constituer en soit une réponse à cette problématique. Ce sera l'objet de la section 4.2.

4.1.1 Option 1 : orienter l'exploration par l'expertise

La première méthode que nous proposons consiste en la définition d'une stratégie d'exploration intégrant dans sa définition les connaissances expertes (règles). Cette stratégie est dérivée de l'approche de référence *bonus-gloutonne* (Geist et Pietquin, 2011) introduite dans la section 3.5.2. Notre proposition consiste en l'ajout d'un terme additionnel dans l'équation 3.26. Ce terme correspond à une fonction « de conseils experts », noté $v(s_t, a)$ par la suite. Le principe est que cette fonction retourne un bonus β_1 quand l'action a est également choisie par les règles expertes sur l'état du dialogue courant s_t ou la valeur 1 sinon.

L'équation 3.26 devient donc :

$$a_t = \underset{a}{\operatorname{argmax}}\, \mu_Q(s_t, a) + \beta \frac{\sigma_Q^2(s_t, a)\ \boldsymbol{v(s_t, a)}}{\beta_0 + \sigma_Q^2(s_t, a)\ \boldsymbol{v(s_t, a)}} \tag{4.1}$$

où β et β_0 sont deux constantes et $\mu_Q(s_t, a)$ et $\sigma_Q^2(s_t, a)$ sont respectivement la moyenne et la variance de l'estimation courante de la fonction de qualité Q. Plus l'incertitude sur l'estimation courante de la fonction de qualité pour le couple état-action (s_t, a) est grande, plus la variance qui lui est associée est grande. Ainsi, suivant l'équation 4.1 l'attribution de ce bonus β favorise l'exploration des actions où l'incertitude sur l'estimation de la fonction de valeur est la plus forte. En outre, une action qui bénéficie d'un agrément expert (c'est à dire qui serait obtenue en appliquant les règles expertes) est également favorisée grâce au facteur de mise à l'échelle que représente la fonction $v(s_t, a)$ définie ci-dessus.

L'objectif ici est de guider l'exploration initiale en premier lieu vers des zones de hautes variances ayant une approbation donnée par l'expertise du domaine (règles expertes). Quand l'espace de recherche est de plus en plus exploré, l'incertitude décroît et la variance diminue de fait. Le choix de l'action gloutonne (maximisant la moyenne) est ainsi favorisé. Nous ferons référence à cette méthode dans la suite de ce manuscrit en tant que technique d'exploration **expert-gloutonne** (*expert-greedy* en anglais).

Nous avons également envisagé l'utilisation d'une stratégie d'exploration alternative nommée **expert-guidée** (*expert-endguided* en anglais) dont l'équation est donnée

ci-dessous :

$$a_t = \begin{cases} \underset{a}{\text{argmax}}\, \mu_Q(s_t, a) + \beta \dfrac{\sigma_Q^2(s_t, a)}{\beta_0 + \sigma_Q^2(s_t, a)} & (t < t_{\text{HDC}}) \\ \text{appliquer les règles expertes} & (t \geq t_{\text{HDC}}) \end{cases} \tag{4.2}$$

avec $t_{\text{HDC}} \in \mathbb{N}^+$. Dans notre implémentation, nous avons fait augmenter progressivement ce paramètre au fil des succès rencontrés par l'agent apprenant. Cette méthode consiste donc à appliquer l'approche *bonus-gloutonne* pendant les premiers tour du dialogue (jusqu'à $t = t_{\text{HDC}}$) puis de guider l'interaction par les règles expertes après coup et ce jusqu'à la fin de l'interaction. Cette technique permet à la tâche d'apprentissage d'assurer un niveau de fonctionnement minimal en début d'apprentissage (guidage expert important si par exemple t_{HDC} est initialisé suffisamment bas) tout en augmentant le niveau d'exploration progressivement (lorsque t_{HDC} va prendre des valeurs de plus en plus grande). Par rapport à **expert-gloutonne**, **expert-guidée** va permettre d'assurer un niveau de qualité de service minimal durant la phase d'apprentissage initiale. Idéalement, il devrait être possible de proposer une valeur de t_{HDC} qui, reliée à l'équation 4.2, permettrait d'assurer un taux de réussite moyen ne s'éloignant que d'une constante du taux de réussite moyen obtenu avec les règles expertes. Ainsi les effets de l'exploration seraient contenus et mieux repartis dans le temps. Pour cette première étude nous avons établi la fonction temporelle de t_{HDC} par quelques seuils définis manuellement.

4.1.2 Option 2 : guider l'apprentissage par l'ajout de récompenses additionnelles déduites de l'expertise du domaine

La seconde proposition consiste en la définition d'une fonction de récompense additionnelle reposant sur les connaissances expertes du domaine (on parlera également de fonction de récompense experte). L'objectif visé est de guider l'apprentissage de l'agent apprenant plus rapidement vers une bonne politique (idéalement optimale) en détectant en amont des situations favorables/défavorables et en le répercutant directement sur les récompenses émises tout au long de l'interaction. Cette idée d'une fonction récompense immédiate transmettant l'information utile à l'apprentissage de façon plus diffuse, par opposition à l'approche plus généralement employée consistant à attendre la fin d'un épisode (dialogue) pour attribuer la récompense la plus utile (succès de l'interaction), emprunte à la notion de *reward shaping* (notamment évoquée dans les sections 3.2 et 3.5.2).

Lorsque cette option est choisie, nous utilisons la fonction de récompense $R^{'}$ dans l'apprentissage en lieu et place de la fonction de récompense initiale R. $R^{'}$ est la somme de la fonction de récompense immédiate en provenance de l'environnement R_{env} ($R = R_{env}$) et de l'experte R_{expert} nouvellement introduite. Ainsi, $R^{'} : S \times A \times S \rightarrow \Re$ avec :

$$R^{'}(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{expert}(s_t, a_t, s_{t+1}) \tag{4.3}$$

où $R_{expert} : S \times A \times S \rightarrow \Re$ est une fonction à valeur bornée dans l'espace des réels. Le modèle MDP $M^{'}$ qui en résulte est représenté par le quintuplet suivant : $\{S, A, T, R^{'}, \gamma\}$.

Cependant, même si l'apprentissage de la politique se fait sur $M^{'}$, l'objectif est de résoudre le problème d'optimisation du MDP M d'origine défini par le quintuplet $\{S, A, T, R, \gamma\}$.

La question qui se pose donc est la suivante :

« *Quelle forme doit prendre R_{expert} pour garantir que la politique optimale sur $M^{'}$ le soit également sur M ?* »

Dans une situation où l'on ne peut pas s'appuyer sur une connaissance a priori des dynamiques du modèle, une solution efficace consiste à avoir recours à une technique reposant sur la définition d'une fonction potentiel sur les états du MDP, notée ψ_{expert}. La fonction potentiel peut être considérée comme définissant une topographie sur l'espace d'état. Comme cela a été démontré dans (Ng et al., 1999) [1], cette technique a pour propriété essentielle de pouvoir garantir que la politique (quasi-)optimale reste inchangée entre $M^{'}$ et M mais également qu'elle peut en accélérer l'apprentissage.

De façon identique à la formulation adoptée pour la fonction F dans (Ng et al., 1999), R_{expert} est définie par :

$$R_{expert}(s_t, a, s_{t+1}) = \gamma\psi_{expert}(s_{t+1}) - \psi_{expert}(s_t) \tag{4.4}$$

où ψ_{expert} est la fonction potentiel. Dans notre proposition, ψ_{expert} fournit une approximation de la progression actuelle de dialogue grâce à des heuristiques basées sur les règles expertes définies manuellement. Ces dernières sont employées pour quantifier l'effort restant pour atteindre le succès du dialogue sur la base de l'état courant du dialogue. Plus le dialogue semblera s'approcher d'un dénouement favorable, plus un fort bonus (récompense positive) sera renvoyé à l'agent apprenant.

4.2 Utiliser l'évaluation subjective de l'utilisateur au cours de l'interaction

Dans cette section nous nous intéressons à la prise en compte de l'évaluation subjective de l'utilisateur au cours de l'interaction pour tenter de répondre à **P2**. En effet, l'utilisateur est le principal acteur du changement de dynamiques (nouveau profil, adaptation au système, etc.) que doit pouvoir gérer l'agent apprenant. Le postulat fait dans cette thèse est que l'évaluation subjective, s'il elle est capturée et utilisée avec une certaine précaution, peut constituer une information précieuse à même de favoriser et d'accélérer l'apprentissage et l'adaptation en ligne de la stratégie de dialogue.

Dans cette thèse, pour permettre l'évaluation subjective, nous nous intéressons tout particulièrement aux **signaux sociaux**. Ces derniers sont décrits dans les travaux de sciences humaines et sociales (anthropologie, psychologie, etc.), comme étant des signaux dont le but est de transmette des informations socialement pertinentes comme

1. Dans (Ng et al., 1999) la preuve était donnée uniquement pour un MDP, dans (Eck et al., 2015) la preuve a également était faite dans le cadre POMDP.

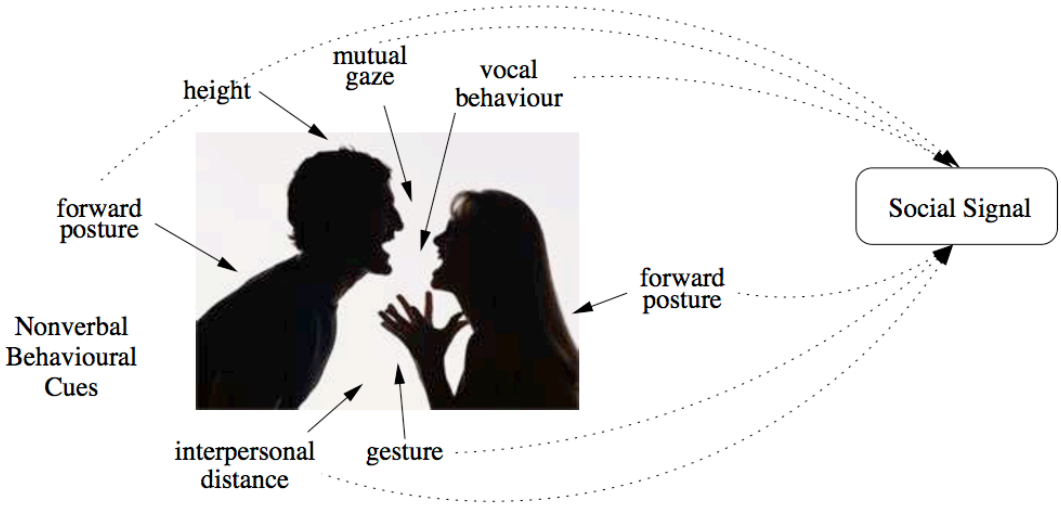


FIGURE 4.1 – *Schéma illustrant les signaux sociaux extrait de (Vinciarelli et al., 2009)*

l'accord (ou le désaccord), l'empathie ou encore l'hostilité et de fait sont capables de façonner nos interactions du quotidien (Richmond et al., 1991; Kunda, 1999; Custers et Henk, 2005). Dans ce qui caractérise l'interaction Homme-Homme, de tels signaux sont exprimés sous la forme d'un ensemble d'indices comportementaux tels que le fait de sourire, de croiser de bras ou encore de rire (voir figure 4.1 à titre d'illustration). Dans (Vinciarelli et al., 2009), les auteurs font un état de l'art complet d'un domaine émergent qui consiste à vouloir doter les ordinateurs de mécanismes permettant à la fois de percevoir ces signaux, d'y donner un sens mais aussi d'être capable d'en émettre en retour. Dans nos travaux, nous nous concentrerons sur l'impact potentiel sur l'apprentissage de considérer dans la définition de la fonction de récompense une sous-classe de ces signaux, à savoir ceux avec lesquels un utilisateur peut transmettre une certaine forme d'évaluation (positive/négative) de la situation courante du dialogue tout au long de l'interaction. Aussi nous emploierons le terme de RL **socialement inspiré** (socially-inspired RL en anglais) pour y faire référence.

Comme nous l'avons déjà mentionné dans la section 3.2, il n'existe pas vraiment de standard pour définir la fonction de récompense immédiate. Or, cette fonction joue un rôle déterminant dans la recherche de la meilleure politique d'interaction du fait qu'elle permet à elle seule de définir la notion d'optimalité pour la tâche qui incombe au système. Dans la plupart des travaux, des critères dits objectifs, tels que la durée et la pleine réalisation du but utilisateur, sont exclusivement employés à cette fin. Cependant, des études récentes ont montré les limites de l'utilisation de ces mesures dans des conditions d'apprentissage en ligne avec de vrais utilisateurs. Par exemple, la difficulté de recueillir avec fiabilité l'information relative au succès de la tâche (Gašić et al., 2010; Sungjin et Eskenazi, 2012). Si dans le paradigme d'évaluation PARADISE (Walker et al., 1997) les critères objectifs ont pu être corrélés aux subjectifs (satisfaction utilisateur) par

régression linéaire, nous arguons que le point de vue de l'utilisateur ne peut être totalement ignoré en pratique sans impacter le naturel de l'ensemble du système. D'autant plus que dans une situation comme celle visée dans **P2**, où la notion d'optimalité est définie comme quelque chose de dynamique (variant au cours du temps), il nous semble pertinent de faire intervenir dans la définition des récompenses attribuées à l'agent apprenant des métriques plus dépendantes du contexte local de l'interaction plutôt que de compter principalement sur une évaluation finale (possiblement erronée).

Malgré quelques tentatives en RL faisant l'usage de signaux subjectifs (comme les émotions) dans la fonction de récompense, par exemple (Broekens et Haazebroek, 2007), peu de travaux ont étudié leur impact dans l'apprentissage de la politique du DM. Deux difficultés techniques peuvent être avancées pour expliquer cela :

— l'algorithme RL employé doit pouvoir garantir un apprentissage en ligne efficace tout en étant conçu pour faire face à la variabilité supplémentaire introduite par les récompenses socialement inspirées ;
— le besoin de techniques robustes permettant de détecter les indices comportementaux (tels que des sourires ou des hochements de tête) et de les interpréter en tant que source de jugements subjectifs sur la conduite de l'interaction.

Bien que la dernière difficulté soulevée n'a pas à proprement parlé fait l'objet de notre étude (puisque nous avons fait le choix de simplifier le problème de l'acquisition de ces récompenses dans nos expériences préliminaires), nous proposons l'utilisation conjointe d'un algorithme RL efficace par échantillon, capable de gérer l'aspect non-stationnaire du processus, à savoir KTD (Geist et Pietquin, 2010), et d'une méthode de *reward shaping* qui nous permettra d'intégrer de façon plus sûre l'information sociale dans le processus d'apprentissage.

4.2.1 Apprentissage par renforcement socialement inspiré

Dans le formalisme RL, l'information portée par les signaux sociaux pourrait être utilisée de multiple façon dans le formalisme RL. Par exemple, elle pourrait faire partie intégrante de l'état du dialogue, ou intervenir en tant que paramètre permettant de réguler le rapport exploration/exploitation dans la problématique du contrôle. De plus, le système pourrait également chercher à émettre de tels signaux (agent émotionnel).

Dans cette thèse, cependant, nous considérons ces derniers exclusivement dans le but de collecter des informations de renforcement additionnelles. Plus exactement nous exploiterons des *récompenses socialement inspirées* (on parlera également de récompenses sociales) traduisant les évaluations positives et négatives transmises par l'utilisateur au cours de l'interaction (et ce possiblement par le biais des signaux sociaux). Ces récompenses sont perçues par l'agent apprenant comme des évaluations intermédiaires, supplémentaires à chaque tour de dialogue. Dans ce scénario, ces évaluations sont considérées comme une estimation du jugement qu'a l'interlocuteur sur l'évolution de l'interaction et donc, implicitement, sur le progrès global de la tâche. Ainsi, la récompense sociale est définie pour traduire l'aspect positif ou négatif de l'évaluation utilisateur au travers d'une valeur réelle signée.

Pour intégrer ces récompenses dans l'apprentissage, le cadre du *reward shaping* que nous avons notamment introduit dans la section 4.1.2, nous a là encore paru adapté. Dans ce cas particulier, la fonction récompense introduite $R^{''}$ est celle faisant la somme de celle donnée par l'environnement R_{env} (critère objectif) et de la fonction de récompense sociale R_{social} (critère subjectif). Le MDP $M^{''}$ qui en résulte est représenté par le quintuplet $(S, A, T, , R^{''}, \gamma)$ où $R^{''}$ est définie par :

$$R^{''}(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{social}(s_t, a_t, s_{t+1}) \tag{4.5}$$

où $R_{social} : S \times A \times S \rightarrow \Re$ est une valeur réelle bornée.

Là encore nous ferons usage de l'approche de *reward shaping* proposée dans (Ng et al., 1999) et reposant sur l'utilisation d'une fonction potentiel. Ce choix est motivé par le fait de vouloir garantir que l'information sociale nouvellement introduite ne puisse pas faire diverger l'agent apprenant de son objectif initial défini au travers du MDP d'origine M (accomplissement de la tâche le plus rapidement possible). Pour cela, R_{social} est définie par :

$$R_{social}(s_t, a, s_{t+1}) = \gamma \psi_{social}(s_{t+1}) - \psi_{social}(s_t) \tag{4.6}$$

où ψ_{social} est une fonction potentiel à valeurs réelles, qui traduit ici la valence de l'évaluation de l'utilisateur observée en chaque état.Sa valeur est réelle bornée, positive (resp. négative) quand l'évaluation subjective l'est aussi.

Dans la suite de cette section nous détaillerons en tout premier lieu comment nous avons choisi de simuler la production et l'acquisition de ces signaux sociaux afin d'en déterminer l'impact sur l'apprentissage. Puis, nous discuterons de quelques pistes d'intérêts concernant l'implémentation d'une solution permettant la capture de ces signaux en pratique. Il est à noter que dans le chapitre 6 nous présenterons des résultats obtenus avec cette méthode dans des conditions d'interactions impliquant de vrais utilisateurs.

4.2.2 Simulation d'évaluations subjectives en cours d'interaction

Bien que nous souhaitons à terme, grâces à nos diverses propositions, proposer un cadre d'apprentissage à même de nous affranchir du recours à la simulation (optimisation de la politique de dialogue avec de véritables interactions), nous comptons dans le cadre d'une étude préliminaire exploiter un simulateur existant afin d'évaluer sur des conditions d'apprentissage et de tests plus paramétrables l'impact de l'ajout de ces récompenses sociales sur la convergence d'une politique apprise. Pour ce faire, nous avons pris pour outil de référence le simulateur utilisateur au niveau intentionnel (actes de dialogue) par agenda tel que décrit dans (Schatzmann et al., 2007b).

Cette approche décompose l'état mental de l'utilisateur S en un agenda A et un objectif à atteindre G :

$$S = (A, G) \text{ avec } G = (C, R) \tag{4.7}$$

G correspond à une représentation du but utilisateur. Son objectif est de garantir que l'utilisateur simulé réagit d'une manière cohérente et orientée vers sa réalisation.

C_0	=	[type=bar drinks=beer area=west]
R_0	=	[name= near= phone=]
Sys_1		Bonjour, comment puis-je vous aider ?
A_1	=	[inform(type=bar) inform(drinks=beer) inform(area=west) request(name) request(near) request(phone) bye()]
Usr_1		Je cherche un bar qui sert de la bière
Sys_2		Ok, un bar à vin. Quelle partie de la ville avez-vous en tête ?
A_2	=	[negate(drinks=beer) inform(area=west) request(name) request(near) request(phone) bye()]
Usr_2		Non, j'ai dit à bière !
Sys_3		Vous cherchez un bar qui sert de la bière, est-ce exact ?
A_3	=	[affirm() inform(area=west) request(name) request(near) request(phone) bye()]
Usr_3		Oui, un bar dans la partie ouest de la ville
Sys_4		Il n'y a pas de bar à bière dans cette zone Mais le Kings Shilling, près du Château, est un bar à bière très réputé.
C_4	=	[type=bar drinks=beer]
R_4	=	[name=Kings Shilling near=Castle phone=]
A_4	=	[request(phone) bye()]
Usr_4		Ok, ça me parait être un bon plan. Quel est son numéro ?
Sys_5		Pouvez-vous répéter s'il vous plaît ?
A_5	=	[request(phone) bye()]
Usr_5		Quel est son numéro de téléphone ?
Sys_6		Son numéro de téléphone est le 728-64-32.
R_6	=	[name=Kings Shilling near=Castle phone=728-64-32]
A_6	=	[bye()]
Usr_6		Merci, au revoir !

TABLE 4.1 – *Exemple de gestion de l'agenda pour la simulation utilisateur sur un dialogue concernant la tâche TownInfo. A_t et G_t ne sont visibles que quand des modifications leur sont apportées.*

Le but utilisateur se décompose pour cela en un ensemble de contraintes C et de requêtes R. Les contraintes correspondent aux arguments de la négociation (valeur des concepts). Ces derniers peuvent notamment être l'objet d'évolutions au cours de l'interaction (changement explicite de but). L'offre effectuée par le système devra cependant les satisfaire pour que l'utilisateur considère effectivement qu'il s'agit là d'une réponse à son but. Les requêtes représentent quant à elles les informations que l'utilisateur cherche à obtenir de la part du système. Par exemple, cela pourra être l'adresse ou le numéro de téléphone d'un restaurant ou encore les horaires d'une ligne de bus. L'agenda A est ici défini comme une pile contenant les actes de dialogue utilisateur en attente. Ces actes sont localement (au tour courant) jugés nécessaires pour atteindre l'objectif de l'utilisateur.

L'exemple donné dans le tableau 4.1 illustre sur un exemple concret d'interaction (tâche de recherche d'informations touristiques sur des établissements d'une ville, *TownInfo*, décrite plus en détail dans la section 4.3.1) la plupart des mécanismes de mise à jour de l'agenda et du but utilisateur intervenant au cours du dialogue pour en permettre l'évolution réaliste.

Au début du dialogue, un nouvel objectif est généré en utilisant la base de données

exploitée également par le système (C_0 et R_0). L'agenda est alors initialisé en y ajoutant toutes les contraintes sous forme d'actes de dialogue de type *inform*, puis toutes les requêtes en actes de type *request* et enfin un acte *bye()* au bas de la pile pour pouvoir mettre fin au dialogue (voir A_1). Les actes de dialogue utilisateur effectivement transmis au système sont construits à partir des n actes de dialogue placés au sommet de A. Dans ce cas de figure, n représente le niveau courant de prise d'initiative de l'utilisateur simulé. Ce niveau est déterminé dynamiquement selon un taux de prise d'initiative établi pour l'utilisateur. Plus celui-ci est haut, plus n a de chance d'être grand. Lors de la réception d'un nouvel acte de dialogue en provenance de la machine, les informations données par le système seront analysées et de nouveaux actes utilisateurs seront insérés dans l'agenda. Tous les actes qui ne sont plus pertinents au regard de A et G sont ensuite supprimés. Selon leur niveau de priorité les nouveaux actes pourront être ajoutés en haut de la pile. C'est par exemple le cas lorsqu'une contrainte utilisateur est violée par le système (comme dans A_2) ou si le système pose une question ou demande une confirmation à l'utilisateur (voir A_3).

Différents profils utilisateurs peuvent être modélisés en agissant sur le mécanisme de prise de décisions de l'utilisateur simulé en réglant sa patience (tolérance aux comportements non appropriés du système) et son taux de prise d'initiative (nombre maximal d'informations transmises au système en un tour de parole). Le lecteur peut se référer à (Schatzmann et al., 2007b) et (Keizer et al., 2010) pour avoir plus de détails sur cette méthode de simulation et les moyens de raffiner les modèles dont elle fait usage avec des données d'apprentissage.

Afin de nous affranchir du besoin de simuler de vrais comportements sociaux, qui nécessiteraient l'utilisation de corpus d'apprentissage spécifiques, nous nous contentons dans cette étude préliminaire de l'identification de situations dans lesquelles un utilisateur (selon son profil) serait susceptible d'émettre un jugement subjectif (qu'il soit positif ou négatif) sur la tournure prise par l'interaction. Par exemple, l'identification d'une situation très négative où l'utilisateur pourrait manifester au système son mécontentement en haussant le ton « non je n'ai pas du tout dit ça ! » ou en grimaçant.

L'hypothèse que nous faisons dans cette simulation est que la nature du jugement que peut émettre l'utilisateur sur le déroulement de l'interaction est en fait fonction des évolutions de A et de G. Prenons pour cela un exemple concret. Dans A_2, la présence d'un acte de négation (*negate*) au sommet de A est du au fait qu'une contrainte de G a été violée par le système (ici *drinks=beer*). Cela peut être perçu comme un indice qui pourrait influencer négativement le jugement que ferait un vrai utilisateur sur le déroulement de l'interaction. De la même manière, l'acte d'affirmation *affirm* au sommet de A_3 pourrait à l'inverse contribuer à l'émission de signaux exprimant un jugement plus favorable. En suivant cette logique, le type de l'acte de dialogue au sommet A peut être employé pour déterminer la valence de l'appréciation que l'utilisateur pourrait vouloir transmettre au système au tour courant.

Ainsi, la simulation des évaluations subjectives (calcul de ψ_{social}) que nous avons choisie, nécessite en tout premier lieu de collecter sur A et G un ensemble d'indices (positifs/négatifs) qui permettront par la suite de déterminer dans quelle situation dia-

Indices positifs	Indices négatifs
1 Acte de dialogue positif au sommet de A (affirm, confirm, etc.)	1 Acte de dialogue négatif au sommet de A (negate, deny, etc.)
2 Nombre d'attributs remplis dans R	2 Taille de l'agenda
3 Résolution partielle de la tâche atteinte	3 Nombre de tours effectués
4 Résolution totale de la tâche atteinte	4 Information déjà transmise au sommet de A

TABLE 4.2 – *Liste des indices positifs et négatifs extraient de l'agenda et du but utilisateur.*

logique se trouve l'utilisateur et ce faisant être à même d'émettre une hypothèse sur son état émotionnel.

Le tableau 4.2 donne la liste complète des indices ainsi employés dans notre étude. Il fait état de quatre indices positifs et négatifs extraits à chaque pas de temps t sur A_t et G_t. Bien sur cette liste pourrait être facilement complétée et améliorée mais nous souhaitons montrer que la simulation peut permettre de montrer l'intérêt des approches proposées même dans un cadre volontairement simplifié (et aussi peu couteux en développement).

Dans un second temps les valeurs des indices collectées sont pondérées avec un facteur d'importance (dépendant notamment du profil de l'utilisateur simulé) afin de déterminer la valeur ψ_{social} pour un état donné. Nous avons fait le choix de nous rapprocher de la façon dont les mesures subjectives sont traitées dans le paradigme d'évaluation PARADISE (Walker et al., 1997). Ainsi, cette valeur sera fonction d'une échelle d'accord en cinq points allant de « très négatif » à « très positif » (échelle de Likert). Pour ce faire, la somme de toutes les valeurs pondérées extraites des indices donne un score global $H_t \in [-1, 1]$. Ce dernier sera replacé sur l'échelle de Likert considérée en utilisant un seuil ξ.

Ainsi, à chaque pas de temps t, ψ_{social} est définie comme suit :

$$\psi_{social}(s_t) = \tau \times \begin{cases} -1 & \text{, si } H_t < -\xi \quad \text{(très négatif)} \\ -0,5 & \text{, si } -\xi \leq H_t < 0 \quad \text{(négatif)} \\ 0 & \text{, si } H_t = 0 \quad \text{(\textit{neutre})} \\ 0,5 & \text{, si } 0 < H_t \leq \xi \quad \text{(positif)} \\ 1 & \text{, si } H_t > \xi \quad \text{(très positif)} \end{cases} \tag{4.8}$$

où τ est un facteur d'échelle (nombre réel) permettant de régler l'importance R_{social} par rapport à R_{env}.

Le calcul de la fonction de récompense sociale peut être décomposé en trois étapes :

1. Collecte d'indices positifs et négatifs sur le nouvel état s_{t+1} (plus exactement sur A_{t+1} et G_{t+1} que nous supposons liés à l'état du dialogue s_{t+1}) ;
2. Calcul de $\psi_{social}(s_{t+1})$;
3. Estimation de la récompense sociale attribuée en utilisant l'équation 4.6 qui fait également intervenir la valeur $\psi_{social}(s_t)$ déterminée avec A_t et G_t.

Un exemple d'un tel processus est résumé dans le tableau 4.3. La première colonne représente l'état du dialogue s_t tel que perçu au travers de l'agenda A_t et du but G_t

s_t	Indices positifs	Indices négatifs	$\psi_{social}(s_t)$	R_{social}
s_3	1(1)	2(-6), 3(-4)	0,5	0,45
s_4	2(2/3) 3(1)	2(-2) 3(-5)	1	

TABLE 4.3 – *Exemple du calcul de la récompense sociale simulée avec* $\gamma = 0.95$ *et* $\tau = 1$.

correspondants (voir tableau 4.1). Les deuxième et troisième colonnes sont respectivement les listes des indices positifs et négatifs détectés avec leur valeur respective entre parenthèses. Par exemple, dans la première rangée et la troisième colonne, l'indice 2 correspond au nombre d'actes de dialogue encore dans l'agenda (voir la deuxième colonne du tableau 4.2), la valeur 6 est déterminée à partir de A_3, le signe $-$ indique que l'indice est négatif. La quatrième colonne correspond à ψ_{social} (c'est à dire au score de Likert) calculé grâce à l'équation 4.8. Pour déterminer cette valeur il est nécessaire d'avoir recours à une pondération sur les indices pour à la fois normaliser leur valeur et en régler l'importance les unes par rapport aux autres (score H_t). Ainsi, différents profils « sociaux » pourront être modélisé en faisant varier cette pondération. Nous avons notamment fait usage de cette capacité dans nos expériences lorsque nous avons modélisé deux profils utilisateur simulé différents : un expert et un novice (voir section 4.3.3). Enfin, la dernière colonne indique la récompense sociale attribuée en appliquant l'équation 4.6 dans le contexte courant. Avec $\gamma = 0,95$ et $\tau = 1$, on obtient le score positif de $0,45$ qui traduit une évolution favorable de la situation entre s_3 et s_4.

4.2.3 Exploiter les signaux sociaux en conditions réelles

Dans des conditions réelles, ces évaluations subjectives doivent être déterminées sur la base de vrais indices comportementaux employés (consciemment ou non) par les utilisateurs du système pour transmettre leur propre jugement de l'évolution du cours de l'interaction. Ceci pourrait se faire au travers de l'emploi d'un jeu de détecteurs opérant sur plusieurs modalités (lecture d'émotions grâce aux traits caractéristiques d'un visage, classification de gestuelles expressives, détecteurs de mots-clés spécifiques, etc.). Ces derniers pourraient ainsi fournir un ensemble de signaux positifs et négatifs détectés durant le tour utilisateur (un sourire, un timbre de voix traduisant l'impatience) avec leur score de confiance respectif dans le but de procéder à une interpolation pondérée, similaire à celle décrite précédemment, pour estimer ψ_{social}. Cette pondération pourrait notamment être estimée en exploitant des données annotées, par exemple en utilisant des techniques de régression comparables à celles employées dans (Rieser et Lemon, 2011) ou (El Asri et al., 2013).

Si de nos jours de nombreux travaux abordent la question de la détection et l'analyse de ces signaux sociaux dans la littérature, on pourra notamment mentionner ce qui se fait dans l'*Interspeech Computational Paralinguistics Challenge* (Schuller et al., 2013). Les performances des méthodes employées demeurent variables si ces signaux sont capturés de façon non contrainte et implicite (Vinciarelli et al., 2009). Outre l'amélioration des composants employés pour la détection de ces signaux, une solution possible (bien que non optimale) à cette problématique consiste à informer au préalable les utilisateurs

du potentiel intérêt qu'il a à émettre ces signaux et d'en forcer les traits (émettre des signaux sociaux plus compréhensibles pour la machine dans le but d'accélérer l'adaptation à leur profil) et ainsi susciter une forme d'adhésion de leur part (*enrolment*). Ceci dit cette adhésion peut être obtenue de façon plus naturelle. Par exemple si l'utilisateur interagissant avec le système en premier lieu est son concepteur lui-même ou si le contexte applicatif s'y prête tout particulièrement (assistant personnel dont on aurait mentionné cette capacité dans le didacticiel).

Dans une situation plus expérimentale, ce processus de captation complexe peut même être simplifié à l'extrême en proposant à l'utilisateur de noter explicitement l'avancement de l'interaction après chaque réponse du système sur une échelle d'accord en cinq points (via par exemple une interface graphique dédiée), note qui sera ensuite directement considérée comme la valeur ψ_{social} pour l'état courant. C'est notamment sous cette forme que nous avons évalué l'impact des signaux de renforcement additionnels sur l'apprentissage en conditions réelles avec de vrais utilisateurs dans le chapitre 6.

4.3 Expériences et résultats

Dans cette section nous allons présenter les conditions expérimentales déployées pour faire l'étude de nos différentes propositions. Nous commencerons ensuite par étudier l'exploitation des connaissances expertes dans le processus d'apprentissage. Enfin nous étudierons en détail l'utilisation de l'évaluation subjective de l'utilisateur.

4.3.1 Conditions expérimentales

Dans un premier temps, nous décrivons la tâche de dialogue sur laquelle nous avons réalisé toutes les expériences de ce chapitre. Puis nous détaillons les conditions d'apprentissage en ligne RL employées spécifiquement dans cette étude. Enfin nous détaillerons les métriques utilisées ainsi que les méthodes de collecte des résultats.

Description de la tâche *TownInfo*

La tâche que nous avons choisie pour évaluer et valider nos différentes propositions est dénommée *TownInfo*. Cette tâche, bien connue dans la littérature, a notamment été employée pour réaliser l'évaluation des systèmes de dialogue développés par les universités de Cambridge et d'Edimbourg (Lemon et al., 2006; Young et al., 2010; Thomson et Young, 2010). Elle vise à fournir aux utilisateurs un service vocal donnant accès à des informations touristiques concernant une ville virtuelle et portant essentiellement sur la recherche de données pratiques (téléphone, adresse, etc.) sur des établissements (hôtels, restaurants, bars et infrastructures). On considèrera donc comme réussi un dialogue où le système aura su retrouver l'intégralité des informations demandées par l'utilisateur. Un exemple d'un tel dialogue est donné dans le tableau 4.4.

Sys_1 >	Bonjour, Comment puis-je vous aider ? hello() [Greet]
Usr_1>	Je suis à la recherche d'un restaurant inform(task=find, type=restaurant)
Sys_2>	Quel type de nourriture aimeriez-vous ? request(food) [BoldRQ]
Usr_2>	Un snack si possible à proximité du cinéma inform(food=snack, near=cinema)
Sys_3>	Ok un restaurant snack près du cinéma. Dans quelle gamme de prix ? confreq(type=restaurant, food=snack, near=cinema, pricerange) [TentRQ]
Usr_3>	Un truc pas trop cher, j'ai un petit budget inform(pricerange=cheap)
Sys_4>	The Express est un restaurant snack bon marché à deux pas du cinéma inform(name='The Express', type=restaurant, food=snack, pricerange=cheap, near=cinema) [Offer]
Usr_4>	Quel est son numéro de téléphone ? request(phone)
Sys_5>	726 4391 inform(phone='726 4391') [Inform]
Usr_5>	Merci et au revoir bye()
Sys_6>	Au revoir et merci d'avoir utilisé notre service bye() [Bye]

TABLE 4.4 – *Exemple d'un dialogue sur la tâche TownInfo.*

```
task -> find (entity){1.0};
entity -> venue(+type,+area, near, -name, -addr, -phone, -comment){1.0};
type -> restaurant(+food, +pricerange, -price, music, drinks, stars){0.33};
type -> hotel(+pricerange, stars, -price, -drinks ){0.33};
type -> bar(+drinks, music, pricerange){0.33};
type -> amenity(){0.01};
area = ("central" | "east" | "north" | "riverside" | "south" | "west" );
near = ("Castle" | "Cinema" | "Fountain" | "Main Square" | "Museum" | ...);
name = ("Alexander Hotel" | "Art House Hotel" | .... );
food = ("Chinese" | "English" | "French" | "Indian" | "Italian" | "Russian" | ...);
pricerange = ("cheap" | "expensive" | "moderate");
music = ("Classical" | "Ethnic" | "Folk" | "Jazz" | "Pop" | "Rock");
drinks = ("beer" | "cocktails" | "soft drinks" | "wine");
stars = ("1" | "2" | "3" | "4" | "5");
addr = ();
phone = ();
price = ();
comment = ();
```

TABLE 4.5 – *Ontologie de la tâche TownInfo.*

Cette tâche, dans sa version cambridgienne (Young et al., 2010) que nous utiliserons par la suite, est définie par l'ontologie arborescente donnée dans le tableau 4.5. Elle fait état de 15 concepts distincts (plus de détails sont disponibles dans l'annexe C.1). A

chaque tour, de nombreuses actions sont à la disposition du DM. En effet, on décompte un total de 11 type d'actes de dialogue différents exploitables dans les réponses du système (plus de détails sont disponibles dans l'annexe A.1).

Pour gérer cette tâche dans nos travaux nous faisons l'usage d'une implémentation du système de dialogue telle que proposée par l'université de Cambridge et qui repose sur le formalisme POMDP HIS (Young et al., 2010), détaillé dans la section 3.4.4. Ce dernier offre un cadre formel capable de ternir compte des erreurs introduites par la chaîne de compréhension en considérant l'état du dialogue comme partiellement observable, mais aussi dans lequel il est possible d'envisager l'optimisation de la politique par RL malgré les grandes dimensions des espaces d'état et d'action considérés.

En effet, comme mentionné dans la section 3.4.4, ce formalisme exploite plusieurs techniques pour rendre le problème soluble. Ainsi, on notera l'utilisation conjointe d'une décomposition de l'état du dialogue qui va permettre de simplifier l'équation de mise à jour de l'état de croyance, d'approximations des modèles ainsi mis en jeu dans cette équation, du regroupement des buts utilisateurs en partitions et de la projection du MDP continu maître dans une version résumée, et enfin de l'emploi d'heuristiques pour appliquer dans l'espace maître les décisions prises dans l'espace résumé (voir section 3.4.4 pour une description plus détaillée de ces différentes techniques).

Dans cette configuration, la fonction de récompense immédiate non enrichie (R_{env}) qui est utilisée pour l'apprentissage et pour l'évaluation des différentes proposition repose exclusivement sur deux critères objectifs, à savoir la réussite (ou l'échec) de la tâche et le nombre de tours. Elle est définie pour pénaliser chaque tour de dialogue par une récompense négative de -1. À l'issue de l'interaction, si l'objectif a été atteint (satisfaction du but utilisateur), une récompense de $+20$ est attribuée contre une de 0 le cas échéant. De par son utilisation dans un mécanisme d'optimisation, on espère ainsi favoriser une politique de dialogue conduisant le plus rapidement au succès de la tâche (efficacité et rapidité). Cependant, on peut d'ores et déjà faire le constat que le critère de réussite de la tâche, qui a le plus d'intérêt pour l'utilisateur (et le concepteur du système) est perçu relativement tardivement (uniquement révélé en fin d'interaction). Il sera donc intéressant d'étudier l'impact de l'utilisation d'une fonction de récompense plus diffuse au travers de nos propositions de *reward shaping* expert et social.

Conditions d'apprentissage par renforcement

Pour des besoins de cohérence et reproductibilité de nos expériences, tous les apprentissages RL considérés par la suite ont été réalisés en ligne en faisant interagir le système avec le simulateur d'utilisateurs présenté dans (Schatzmann et al., 2007b) (ou quand cela est mentionné de son extension proposée dans la section 4.2.2 produisant également des récompenses socialement inspirées). Dans la plupart des configurations que nous considèrerons par la suite, le taux d'erreur de compréhension simulé sera fixé à 10% lors des expériences pour reproduire des situations où l'incertitude doit être gérée.

Pour réaliser l'apprentissage RL nous avons fait l'usage exclusif de l'algorithme

hors-politique KTD-Q avec $\gamma = 0.95$. Ce choix est justifié en raison de ses bonnes propriétés qui lui permettent de réaliser un apprentissage en ligne très efficace mais aussi d'être plus tolérant aux phénomènes de non-stationnarités (voir 3.5.3).

Une paramétrisation linéaire de la fonction de qualité similaire à celle employée dans (Daubigney et al., 2012) a été choisie. Cette technique a été préférée à une paramétrisation non-linéaire (par exemple un réseau de neurones) dans ce même formalisme du fait sa meilleure garantie de convergence (Daubigney et al., 2012). Dans la configuration *TownInfo* l'espace d'état résumé peut être défini par :

$$S =< b(hyp1), b(hyp2), \text{last-uact}, \text{p-status}, \text{h-status} > \tag{4.9}$$

où $(b(hyp1), b(hyp2)) \in [0,1] \times [0,1]$, last-uact $\in [\![1,20]\!]$, p-status $\in [\![1,5]\!]$ et h-status $\in [\![1,6]\!]$. L'espace d'action est quant à lui défini par :

$$A = \{a | a \in [\![1,11]\!]\} \tag{4.10}$$

Il est à noter cependant que nous n'exploitons pas le h-status dans nos configurations apprises par KTD afin de mieux coller à la proposition initiale faite dans (Daubigney et al., 2012). Cette paramétrisation considère l'utilisation d'un RBF défini pour chaque $s \in S$ et pour chaque $a \in A$ par l'équation :

$$\phi^T(s,a) = \delta(a,a_1)\phi^T(s), \ldots, \delta(a,a_{11})\phi^T(s) \tag{4.11}$$

avec

$$\begin{aligned}\phi^T(s) = [1, \varphi_1^1(b(hyp1), b(hyp2)), \ldots, \varphi_3^3(hyp1, hyp2),\\ \delta(\text{last-uact}, 1), \ldots, \delta(\text{last-uact}, 20),\\ \delta(\text{p-status}, 1), \ldots, \delta(\text{p-status}, 5)]\end{aligned} \tag{4.12}$$

et où δ est une fonction de Kronecker définie par $\delta(x,y) = 1$ si $x = y$, 0 sinon.

Ainsi, trois gaussiennes équi-réparties φ dans $[0,1] \times [0,1]$ et d'écart-type $\sigma = \sqrt{0.2}$ sont utilisées afin de quantifier l'espace à deux dimensions couvert par les deux variables continues de l'espace d'état résumé, $b(hyp1)$ et $b(hyp2)$.

$$\varphi_j^i(hyp1, hyp2) = exp(\frac{-(\|b(hyp1) - x_i\|^2 + \|b(hyp2) - x_j\|^2)}{2\sigma^2}) \tag{4.13}$$

(x_i, x_j) étant les centres des gaussiennes considérées.

En ce qui concerne les variables catégorielles last-uact et p-status (ici vu comme des variables discrètes), chacune d'entre elles est associée à un vecteur composé de la concaténation des fonctions Kronecker sur chacune de ses valeurs (dont une et une seule est alors à 1).

Sachant qu'en configuration linéaire Q est approximée par :

$$\hat{Q}_\theta(s,a) = \theta^T\phi(s,a) \tag{4.14}$$

Ainsi, cela porte à $(1 + 9 + 20 + 5) * 11 = 385$ le nombre de paramètres du vecteur θ dont la valeur doit être estimée par l'intermédiaire de KTD.

Lorsqu'il n'est pas précisé explicitement le contraire, la stratégie d'exploration utilisée lors de l'apprentissage est l'approche *bonus-gloutonne* décrite dans la section 3.5.2 (équation 3.26), avec $\beta = 1000$ et $\beta_0 = 100$. Cette stratégie est employée pour effectuer une recherche dans l'espace des solutions plus raffinée en exploitant la variance des estimations de la fonction de qualité. Lorsque l'agent apprenant fera l'usage exclusif de R_{env} dans les conditions listées ci-dessus, nous considèrerons qu'il s'agit là de la configuration d'apprentissage état de l'art de référence. Elle sera notée **BASELINE** par la suite.

Métriques d'évaluation

Dans cette étude, nous avons fait le choix d'utiliser deux métriques pour juger du niveau de performance des différentes méthodes considérées, à savoir la **moyenne des R_{env} cumulées** sur chaque dialogue et le **taux de réussite**. Le rapport entre ces deux métriques informant directement sur la durée moyenne des échanges (voir définition de R_{env}).

Nous avons également distingué deux conditions de collecte de ces métriques : celles d'apprentissage et celles de test.

Dans le premier mode, la politique de contrôle est autorisée à effectuer de l'exploration dans l'espace de recherche. Dans ces conditions, les résultats seront donnés en fonction du nombre de dialogues réalisés. Ils seront lissés sur les courbes présentées grâce à l'utilisation d'une fenêtre glissante de 100 dialogues (en raison d'un point tous les 30 dialogues) afin de tenir compte de l'évolution dynamique du niveau de performance des politiques apprises.

En conditions de test, la politique de contrôle détermine l'action suivante en exploitant de façon gloutonne la fonction de qualité précédemment apprise en ligne ($a_t = \underset{a}{\text{argmax}}\, \mu_Q(s_t, a)$). Les résultats seront cette fois acquis selon différents niveaux de CER et seront des moyennes obtenue sur 1000 dialogues utilisant le même niveau de CER.

Du fait de l'aspect stochastique du problème de la conduite de l'interaction, nous avons fait le choix de réaliser 50 processus d'apprentissage (resp. de test) indépendants et ce pour toutes les configurations considérées dans cette étude pour pouvoir faire reposer nos analyses sur des résultats statistiquement significatifs. Les résultats qui vont suivre seront donc des moyennes obtenues sur ces 50 processus distincts, auxquelles nous ajouterons les écarts-types sur les courbes et les tableaux qui y font référence.

4.3.2 Utilisation de l'expertise dans l'apprentissage

La première série d'expériences que nous présentons ici a pour but d'évaluer l'impact sur **P1** de l'utilisation des connaissances expertes, au travers de la mise œuvre

Préconditions	**Actions résumées**
h-status = Rejected ou last-uact = reqalts	OfferAlt
h-status = Supported et (p-status $\in$ [Unique, Group, Unknown])	Offer
b(hyp1) $>$ 0.7	Request
0.5 $<$ b(hyp2) $\leq$ 0.7	ConfReq
h-status = Completed	Bye
cas par défaut	Confirm

TABLE 4.6 – *Extrait des règles expertes employées.*

des deux options présentées dans les sections 4.1.1 et 4.1.2. Comme il en a déjà été fait mention plus haut, le concepteur d'un système de dialogue doit généralement définir les comportements de ce dernier (sa politique) au travers de l'établissement de règles expertes pour la gestion du dialogue. Il s'agit notamment là d'une approche courante dans l'industrie (Pieraccini et al., 2009) pour initier un cycle d'amélioration de la politique d'interaction en alternant phases de test face à de vrais utilisateurs et ajustement manuel de la politique par des experts.

Selon le même principe, nous avons donc défini une politique experte pour la tâche *TownInfo*, notée **HDC** dans la suite de ce manuscrit. Cette politique repose exclusivement sur l'application de règles exploitant les informations de l'état résumé afin de déterminer la prochaine action. Les règles ainsi considérées ont toutes été définies manuellement par un expert du domaine. Un extrait de ses règles est disponible dans le tableau 4.6. Comme nous pouvons le voir, elles sont assez intuitives et prennent en compte l'incertitude de façon relativement simple par la définition de seuils sur les scores des deux meilleures hypothèses présentes dans l'état résumé *b(hyp1)* et *b(hyp2)*.

L'application de ces règles permet par exemple de choisir l'action résumée *OfferAlt*, qui permet au système de proposer un autre établissement (alternative), lorsque la meilleure hypothèse fait état de la présence de valeurs niées par l'utilisateur ou lorsque ce dernier a fait une demande explicite au système de lui proposer une alternative (voir ligne 2 du tableau 4.6). Il est à noter qu'en l'état ces règles sont largement perfectibles. Cependant, il n'était pas ici question d'établir une politique experte optimisée sur la tâche mais de reproduire une situation initiale, où l'expert, sur la base de quelques intuitions, définit une première politique d'interaction. Il est à noter aussi que c'est sur la base de ces mêmes connaissances (mêmes règles que celles utilisées pour **HDC**) que nous avons implémenté les options décrites dans la section 4.1.

En effet, pour l'Option 1 (voir section 4.1.1), elles ont permis pour la technique *expert-gloutonne* de définir la fonction υ qui donne un bonus β_1 quand l'action évaluée par l'agent apprenant est la même que celle qui serait prise par les règles, et 1 sinon. Selon l'équation 4.1, plus β_1 est grand, plus les conseils experts sont favorisés. Pour ce qui est de la stratégie *expert-guidée* elles sont employées pour conduire l'interaction dès le tour t_{HDC} atteint.

Pour l'Option 2, la fonction ψ_{expert} prendra après chaque transition une valeur traduisant l'effort restant en ce point pour résoudre le but utilisateur (estimation). Pour ce faire, il conviendra au préalable de déterminer si l'action sélectionnée est bien celle

qui correspond à celle choisie par l'application des règles (couverture des jugements). Dans nos expérience, ψ_{expert} pourra prendre une valeur entière entre 0 et 5, la valeur par défaut étant 0 (cas notamment où l'on est en dehors de la couverture des jugements, situations non prévues par les règles). Plus la valeur est grande, plus cela veut dire que l'expert estime que l'on s'approche d'une fin positive du dialogue. Par exemple si l'état résumé s_t a pour h-status *Completed* et que la décision prise est *Bye* cela veut dire qu'on se dirige vers une fin positive du dialogue, ψ_{expert} prendra alors la valeur 5 pour s_{t+1}.

Autre exemple, si le p-status *Unique*, que le h-status est *Supported* et que la décision est *Offer*, il s'agit là encore d'une situation plutôt positive, car le système a proposé sa première solution. Cependant, on constate que la situation est moins favorable que dans l'exemple précédent car l'utilisateur n'a pas encore validé l'offre du système. Ainsi, ψ_{expert} prendra une valeur plus faible (2).

Dans la suite, nous allons comparer à nos deux références, à savoir **BASELINE** et **HDC**, à cinq autres configurations en liens avec nos premières propositions. Ainsi on distinguera :

— **EXPERT-GREEDY** et **EXPERT-GREEDY-FULL**, font toutes deux usage du schéma d'exploration *expert-glouton* en phase d'apprentissage (Option 1, voir section 4.1.1) mais avec un β_1 différent (respectivement $\beta_1 = 12$ et $\beta_1 = 1000$) ;
— **EXPERT-ENDGUIDED** qui adopte la stratégie d'exploration *expert-guidée* (Option 1, voir section 4.1.1). Nous avons choisi d'initialiser $t_{\text{HDC}} = 1$ et d'augmenter de 1 sa valeur tous les 10 dialogues réussis ;
— **EXPERT-SHAPING** qui utilise la technique de *reward shaping* expert (Option 2, voir section 4.1.2) ;
— **EXPERT-GREEDY-SHAPING** qui exploite les deux options simultanément selon les configurations combinées de **EXPERT-GREEDY** et **EXPERT-SHAPING**.

Dans un premier temps nous évaluerons ces différentes configurations en condition d'apprentissage. Au maximum ce sont 500 dialogues qui seront considérés pour nous concentrer sur la phase d'amorce qui est la plus critique pour l'apprentissage en ligne d'un système de dialogue. Enfin, nous nous attacherons à l'étude en condition de test de l'influence du bruit (différents niveaux de CER simulés).

Étude en condition d'apprentissage

Dans cette expérience, les résultats sont présentés sur les figures 4.2 et 4.3 pour en améliorer la lisibilité.

Malgré ses caractéristiques sous-optimales (règles déterministes intuitives), **HDC** obtient globalement de bons résultats que ce soit en termes de récompenses cumulées ou de taux de réussite (respectivement 11,1 et 86,1% en moyenne). Cependant, le niveau de performance atteint par **BASELINE** en fin d'apprentissage montre l'avantage qu'il y a à considérer un apprentissage par RL plutôt qu'une approche déterministe fixe. En effet, on constate qu'en seulement 210 dialogues les performances de **BASELINE** surpassent celles de **HDC** sur les deux métriques considérées (+0,6 en termes de récompenses cumulées et +7,8% sur le critère de réussite en moyenne). Pour informa-

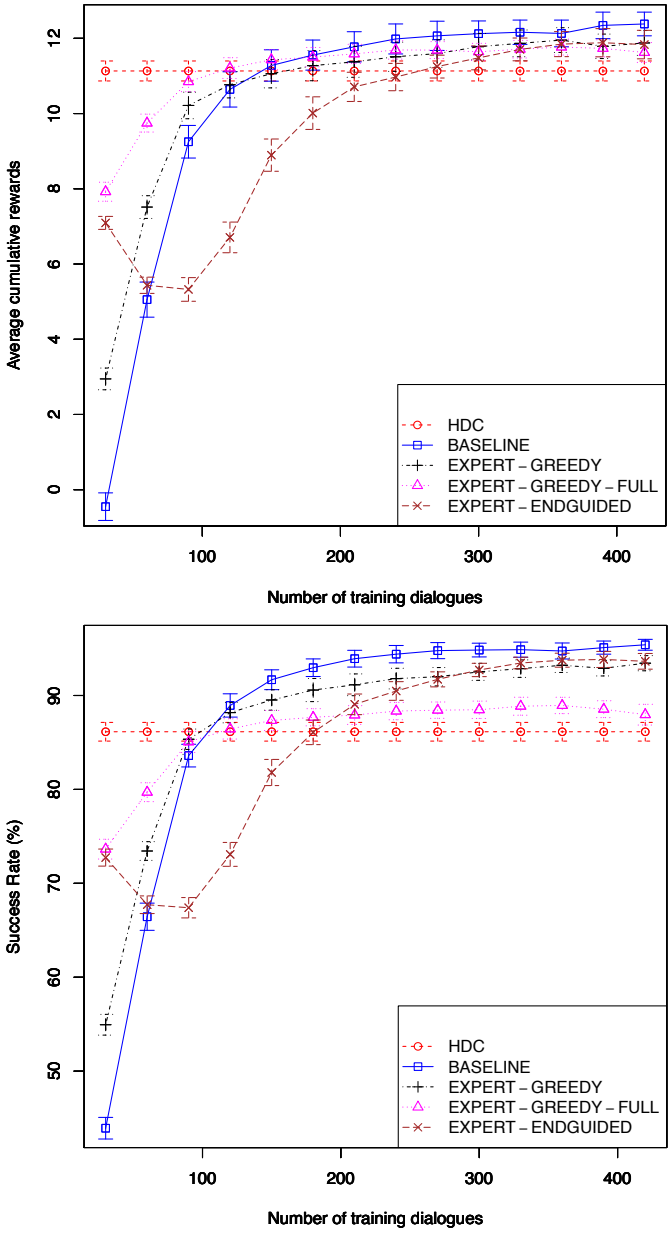


FIGURE 4.2 – *Résultats de HDC et KTD-Q avec et sans l'utilisation du schéma d'exploration expert-glouton et celui expert-guidé (contexte d'apprentissage).*

tion, ces gains ont été déterminés comme étant statistiquement significatifs au regard d'un test non paramétrique de Mann-Whitney, $p < 0,05$. On constate également que cet écart se creuse encore après plusieurs centaines de dialogues. Néanmoins, ces bonnes performances sont obtenues au prix de mauvais résultats en début de l'apprentissage.

Lorsque que l'on considère les courbes de performances obtenues par **BASELINE**, il

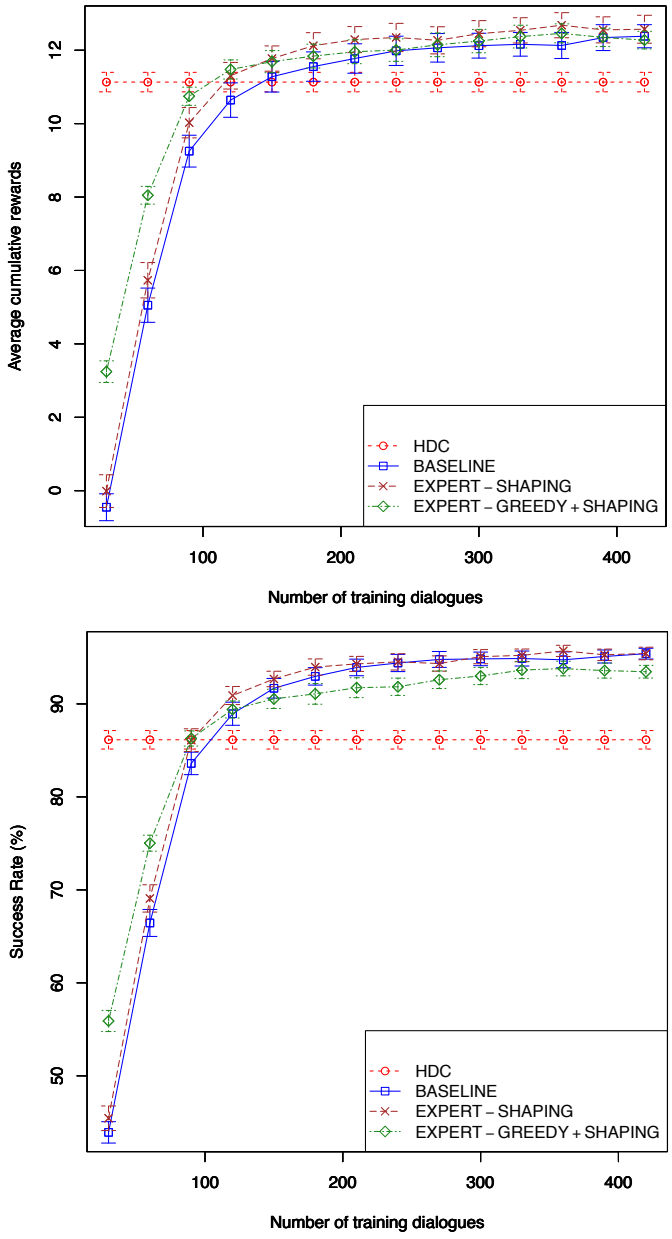


FIGURE 4.3 – *Résultats de HDC et KTD-Q avec et sans l'utilisation de la fonction de récompense experte.*

est possible d'identifier trois phases dans le processus d'apprentissage [2] que nous avons également choisi de représenter sur la figure 4.4 pour plus de clarté :

— **Phase de démarrage à froid** (A) de 0 à 100 dialogues. L'agent explore principa-

2. Il est à noter que des phases semblables sont également observables dans des conditions d'apprentissage faisant intervenir de vrais utilisateurs. On pourra donner comme exemple les expériences réalisées dans (Gašić et Young, 2011). Les bornes données sont pas contre différentes car elles dépendent de l'efficacité de l'algorithme de RL choisi, mais aussi de la difficulté de la tâche de dialogue visée.

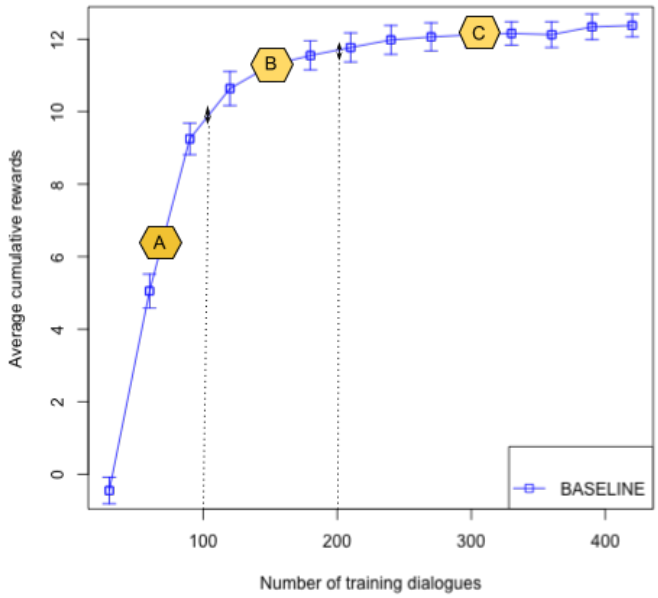


FIGURE 4.4 – *Illustration des différentes phases de l'apprentissage sur les performances obtenues en ligne avec BASELINE (récompenses cumulées).* ***A*** *= phase de démarrage à froid,* ***B*** *= phase d'amélioration et* ***C*** *= phase de convergence.*

lement l'espace d'action sans connaissance a priori sur les dynamiques en utilisant la variance de ses estimations (schéma d'exploration *bonus-glouton*). À ce stade de l'apprentissage les performances du système sont telles qu'il est difficile d'en envisager son utilisation avec des utilisateurs finaux (clients). En pratique, seul un utilisateur préalablement formé peut agir face à un tel niveau de performance.

— **Phase d'amélioration** (B) de 100 à 200 dialogues. Ici, l'agent exploite principalement son estimation courante de la fonction de qualité tout en s'autorisant quelques explorations. Ses décisions sont désormais comparables à celles prises par des règles expertes, mais l'efficacité globale de sa politique continue de s'améliorer.

— **Phase de convergence** (C) au-dessus de 200 dialogue, l'agent affine son estimation par de l'exploration occasionnelle et converge vers un optimum stable (et ce aussi longtemps que les dynamiques de l'environnement restent inchangées).

Ce découpage en phases permet d'expliciter l'objectif qu'il y a derrière le recours aux connaissances expertes, à savoir résoudre **P1**. Cet objectif est donc double. En effet, il consiste à améliorer le niveau de performance atteint au cours de la phase de démarrage à froid, mais aussi à réduire la durée de la phase d'amélioration sans pour autant retarder la phase de convergence optimale.

La figure 4.2 montre que le premier point peut être résolu en mettant davantage l'accent sur les conseils de l'expert lors de la phase d'exploration initiale. En effet, à la fois **EXPERT-GREEDY** et **EXPERT-GREEDY-FULL** obtiennent en début d'apprentissage de meilleurs résultats que **BASELINE** (respectivement $+3,5$ $+12,1\%$ et $+8,2$ $+29,7\%$ à 30

dialogues). On constate toutefois que si l'on met trop de poids sur l'expertise en début d'apprentissage cela a un impact négatif sur les performances obtenues en phase de convergence. Ainsi, **EXPERT-GREEDY** et **EXPERT-GREEDY-FULL** obtiennent respectivement $-0,57$ $-1,92\%$ et $-0,61$ $-6,56\%$ à 390 dialogues. Affirmations là encore validées par un test statistique de Mann-Whitney, $p < 0,05$.

Ces conclusions sont à mettre en relation avec le dilemme entre exploration et exploitation que nous avons mentionné dans la section 3.5.2. Nous savons en regardant **HDC** que les connaissances expertes ne sont pas suffisantes pour déterminer la politique optimale, il conviendra donc de définir de manière appropriée la fonction v. Il s'agit là de faire un compromis entre un niveau de performance initial, permettant d'améliorer l'interaction avec de vrais utilisateurs, et le fait de retarder l'atteinte de la phase convergence vers le véritable optimum (du fait d'une exploration trop faible de l'espace de recherche).

Les résultats obtenus par **EXPERT-ENDGUIDED** démontrent là encore le besoin de l'exploration pour atteindre des performances plus proches de **BASELINE**. Cette méthode propose en effet de retarder l'exploration afin de garantir un niveau de performance minimal durant l'apprentissage. Elle présente l'avantage sur **EXPERT-GREEDY-FULL** d'avoir un meilleure comportement en phase de convergence. Cependant ce gain est obtenu au prix d'une perte de performance observable en début d'apprentissage et qui est liée à l'introduction progressive de l'exploration dans le processus. Une telle méthode peut s'avérer appropriée dans le cadre industriel afin de garantir un niveau de qualité de service minimal (utilisateurs $\approx$ clients) sans sacrifier pour autant l'atteinte (plus tardive) de la politique (quasi-)optimale. Nonobstant, dans le contexte applicatif qui est le nôtre, il nous a paru plus avantageux de favoriser l'utilisation de techniques nous permettant d'atteindre plus rapidement de bonnes performances et donc nous avons préféré laisser de côté cette proposition et de garder en perspective une exploration plus complète de cette piste de recherche.

La figure 4.3, quant à elle, permet de constater que la méthode **EXPERT-SHAPING** obtient des résultats légèrement supérieurs à ceux de **BASELINE** dans la phase d'amélioration (respectivement $+0,66$ $+1,96\%$ en moyenne à 120 dialogues). De plus, cet avantage est conservé lors de la phase de convergence puisqu'une politique plus efficace (moins de tours pour atteindre le succès de la tâche) a été apprise à la fin de la phase d'amélioration (500 dialogues). Résultats qui sont là encore validés par un test statistique de Mann-Whitney, $p < 0,05$. Ainsi cela constitue en soit une réponse au second point mentionné plus haut. On peut également constater que les performances sur les deux métriques augmentent légèrement plus vite que celles de **BASELINE**. Néanmoins, si l'on compare cette méthode à celles utilisant le schéma d'exploration *expert-glouton*, le niveau de performance initial est encore très bas.

Afin de répondre aux deux points soulevés de façon simultanée, nous proposons pour cela de combiner les deux options proposées (**EXPERT-GREEDY-SHAPING**). On constate que cette proposition semble bénéficier des mêmes avantages identifiés sur les deux options utilisées de façon isolée. Une légère perte en termes de taux de réussite est tout de même observable en phase d'amélioration et de convergence par rapport aux

résultats de **BASELINE** et de **EXPERT-SHAPING** seule.

Étude en condition de test

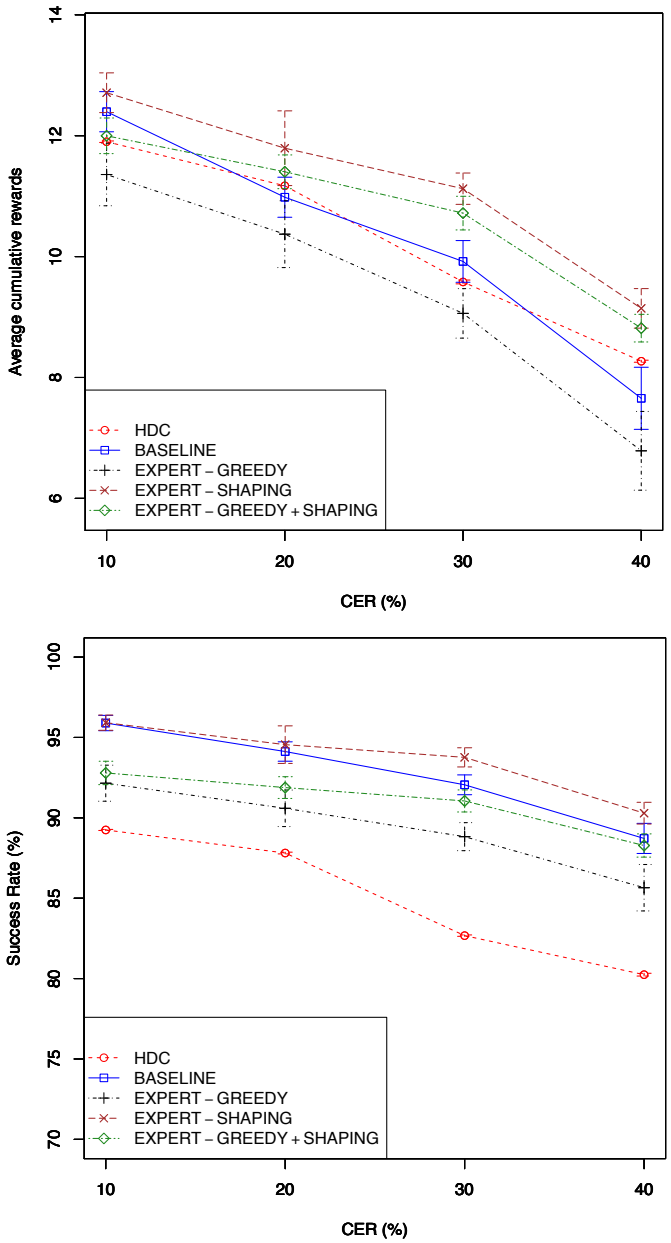


FIGURE 4.5 – *Résultats de **HDC**, KTD-Q avec et sans l'utilisation de connaissances expertes (options) dans différentes conditions bruités (contexte de tests).*

Cette expérience se concentre sur l'impact du bruit intervenant sur l'observation (erreurs de compréhension) dans la procédure d'optimisation de la stratégie du dia-

logue. Les résultats sont présentés dans la figure 4.5 en termes de moyenne sur les récompenses cumulées par rapport à différents niveaux de CER. Sur ces courbes, chaque point est une moyenne obtenue sur les résultats du test de 50 politiques apprises en ligne sur 400 dialogues. Tests effectués sur 1000 dialogues par niveau de CER.

Sur toutes les courbes représentées, on constate qu'une augmentation du CER implique automatiquement une diminution des performances. En présence d'un niveau de bruit élevé sur les observations, **HDC** et **BASELINE** obtiennent des résultats comparables en termes de récompenses cumulées (respectivement $8,3$ et $7,6$ à 40% de CER). Cependant, lorsqu'on considère uniquement le taux de réussite les résultats sont cette fois en faveur de la méthode d'apprentissage ($88,7$ contre $80,2$ à 40% de CER).

Cette baisse en termes de récompenses cumulées observée pour **BASELINE** malgré un taux de réussite plutôt élevé traduit le fait que des politiques moins efficaces (au regard du nombre de tours moyen pour résoudre la tâche de dialogue) sont apprises lorsque l'agent apprenant est confronté à des conditions d'interaction moins favorables (corrections des erreurs). Ce problème semble cependant atténué quand une technique de *reward shaping* expert est adoptée. En effet, l'approche **EXPERT-SHAPING** surclasse l'ensemble des méthodes considérées dans toutes les conditions de bruits, et ce malgré sa définition plutôt approximative (règles expertes sous-optimale). L'une des raisons avancées pour expliquer ce phénomène est la forme plus diffuse prise par la fonction de récompense qui est désormais capable d'attribuer un bonus (ou une pénalité) à un comportement local du système. Ainsi, en présence d'un niveau de CER élevé, la récompense experte est à même de (dé)favoriser une transition système malgré l'échec (ou la réussite) de la tâche globale.

En ce qui concerne **EXPERT-GREEDY**, nous n'observons pas d'amélioration sur les résultats. Ainsi, guider l'exploration par l'expertise ne semble pas suffisant pour faire face aux conditions bruitées. Enfin, **EXPERT-GREEDY+SHAPING** fait état de meilleures récompenses que **BASELINE** mais d'un taux de réussite global plus faible. Ceci confirme que l'utilisation de l'exploration guidée par l'expertise se limite à des cas où les conditions d'interaction ne sont pas trop dégradées. En effet, la prise en compte de l'incertitude dans les règles est quelque chose de difficile car elle requiert plus que de l'intuition. En effet, les seuils employés dans la prise de décision doivent correspondre aux conditions réelles d'interaction, et ceci passe généralement par une analyse plus fine des dialogues réalisés qui n'est plus compatible avec un apprentissage complètement en ligne.

Bilan intermédiaire

Pour conclure cette première étude, nous pouvons dire, au regard des résultats précédents, que les connaissances expertes (règles) ont la capacité d'outiller l'amorçage de l'apprentissage système de zéro par un algorithme RL efficace par échantillon (ici KTD). Si l'Option 1 via ses deux variantes a montré qu'elle pouvait améliorer les performances à l'état initial de l'agent apprenant ou du moins en assurer un niveau minimal, l'Option 2 a quant à elle permis d'obtenir des gains significatifs sur l'ensemble de l'apprentissage

et su accroître la tolérance de l'agent apprenant aux conditions bruitées. De plus, cette dernière, par l'utilisation d'une technique de *reward shaping* introduit une représentation plus diffuse de la fonction de récompense. Ainsi, contrairement à une approche classique reposant principalement sur un jugement final de la réussite du dialogue, certains comportements locaux pourront également être pénalisés/renforcés. De plus, de par l'utilisation d'une technique reposant sur la définition d'une fonction potentiel, l'Option 2 dispose également de propriétés théoriques qui lui garantissent la convergence vers une solution (quasi-)optimale malgré la qualité effective des règles expertes (sous-optimales, incomplètes, etc.).

Si dans ce travail nous nous sommes limité à des conseils experts, il est néanmoins tout à fait envisageable que de telles règles ou conseils soient obtenues par le biais d'une autre politique (traitant par exemple une tâche connexe). Dans ce cas, l'emploi de techniques de transfert pourraient être alors être envisagé. Un panorama complet de ces techniques est notamment disponible dans (Taylor et Stone, 2009). Cependant, l'idée de s'appuyer ici sur des heuristiques expertes nous permet de pouvoir apporter une réponse à **P1** sans besoin particulier en données d'apprentissage sur la tâche de dialogue visée. Il est à noter que quand de telles données sont disponibles il est possible d'utiliser des techniques telles que celle proposée dans (Su et al., 2015), où un RNN est employé pour estimer une fonction potentiel similaire à celle employée dans l'Option 2 sur des données (ici des dialogues simulées) afin d'accélérer l'apprentissage en ligne de la politique de dialogue face à de vrais utilisateurs mais au prix d'une collecte préalable de données.

4.3.3 Utiliser l'évaluation subjective de l'utilisateur dans l'apprentissage

Avant de nous concentrer sur **P2**, nous allons dans un premier temps évaluer l'impact sur l'apprentissage de l'utilisation d'évaluation subjective, au travers de la mise œuvre de la technique de simulation décrite dans la section 4.2.2. Il conviendra dans un premier temps d'étudier plusieurs configurations pour identifier quels types de signaux sociaux sont utiles. Puis comme précédemment, nous évaluerons l'impact du bruit sur l'apprentissage (erreurs de compréhension mais aussi des signaux de renforcement). Enfin nous étudierons l'effet de notre proposition sur les capacités d'adaptation du DM à de nouvelles dynamiques.

Étude en condition d'apprentissage

Dans cette première étude, nous voulons évaluer l'impact sur l'apprentissage en ligne de l'introduction des récompenses socialement inspirées. Pour cela, les conditions décrites dans la section 4.3.1 sont reprises. Le mécanisme d'apprentissage de référence **BASELINE** est ainsi comparé à quatre configurations du simulateur capable d'émettre des récompenses sociales tel qu'introduit dans la section 4.2.2.

L'approche standard, notée **SOCIAL**, considère à la fois les indices positifs et négatifs présentés dans le tableau 4.2. Les configurations **SOCIAL-NEG** et **SOCIAL-POS**

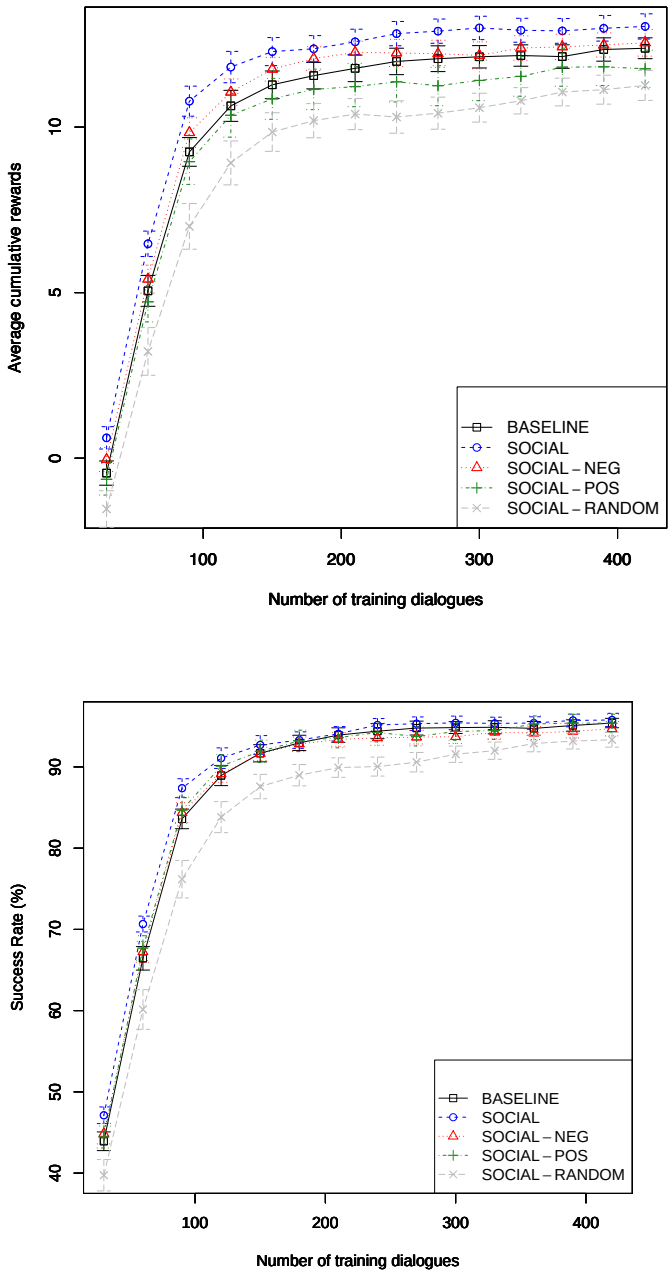


FIGURE 4.6 – *Résultats de 4 configurations différentes d'apprentissage socialement inspiré comparées à la méthode de KTD-Q de référence (contexte d'apprentissage)*

font respectivement l'usage exclusif des indices négatifs et positifs. Lorsqu'il est fait usage des indices, les pondérations utilisées pour calculer ψ_{social} ont toutes été déterminées par intuitions expertes afin correspondre le plus possible au jugement subjectif qu'émettrait un utilisateur « moyen » du système. Dans toutes les configuration τ sera égal à 4. La dernière configuration, notée **SOCIAL-RANDOM**, effectue elle une géné-

ration complètement aléatoire de ses récompenses sociales et nous permettra ainsi de vérifier l'impact limité d'une mauvaise utilisation de cette fonctionnalité.

Les résultats obtenus dans les conditions d'apprentissage en ligne sont présentés dans la figure 4.6 selon les métriques proposées dans la section 4.3.1.

On peut observer qu'en termes de performances, **SOCIAL** surpasse légèrement **BASELINE** en faisant mieux d'environ 0,5 point en récompenses pour un taux de réussite équivalent. De même, le temps d'apprentissage pour atteindre un même niveau de performance est réduit. Par exemple, le niveau de performance obtenu après avoir effectué 200 dialogues avec **BASELINE** est atteint en environ 100 dialogues en utilisant **SOCIAL**.

Comme attendu, **SOCIAL-RANDOM** obtient les moins bonnes performances, suivi par **SOCIAL-POS**, **BASELINE** et **SOCIAL-NEG**. **SOCIAL** qui combine les deux indices positifs et négatifs obtient les meilleurs résultats. Toutes les configurations (sauf **SOCIAL-RANDOM**) sont assez proches si l'on considère la déviation standard des résultats. Cependant, un point important confirmé par **SOCIAL-RANDOM** est que, même dans le cas où l'attribution des récompenses sociales repose entièrement sur l'aléatoire (non-informative), la technique de *reward shaping* employée assure que la convergence vers la politique (quasi-)optimale est encore préservée. D'après cette expérience, même si **SOCIAL-POS** et **SOCIAL-NEG** obtiennent des résultats proches sur la métrique du taux de réussite, il semble que l'apprentissage est conduit plus efficacement à l'aide de l'information négative si l'on considère les récompenses cumulées.

Étude en condition de test

Bien que l'expérience précédente ait montré des résultats encourageants lorsqu'un apprentissage par renforcement socialement inspiré est employé, il convient de garder à l'esprit que dans les conditions précédentes les récompenses sociales étaient issues d'observations parfaites des « signaux sociaux » émis par l'utilisateur simulé. Dans une configuration plus réaliste, de tels signaux, en raison de leur complexité inhérente (dimension multimodale, interprétation dépendante du contexte, etc.), ne pourront en toute logique être interprétés parfaitement par la machine. Du fait de l'utilisation dans cette étude préliminaire d'un outil de simulation, ce niveau de complexité additionnel a été introduit artificiellement en simulant un certain taux d'erreur des récompenses sociales (*Social Reward Error Rate* - $R_{soc}ER$). Ainsi selon une fréquence contrôlée, la valeur de ψ_{social} est modifiée de façon aléatoire pour l'état courant.

Comme mentionné dans la section 3.5.2, lorsqu'un apprentissage en ligne est adopté, cela requiert une évaluation du système de dialogue de la part de l'utilisateur à la fin de chaque interaction. Cette évaluation porte généralement sur la réussite de la tâche (métrique objective). En conditions réelles, bien que cette évaluation soit donnée par de vrais utilisateurs, les travaux présentés dans (Gašić et Young, 2011) montrent que ces retours peuvent s'avérer erronés. Pour expliquer ce phénomène, on peut avancer la nature subjective du processus d'évaluation chez l'homme. Par exemple, bien que l'objectif soit atteint à la fin de l'échange, toute utilisation d'actions incohérentes de la part

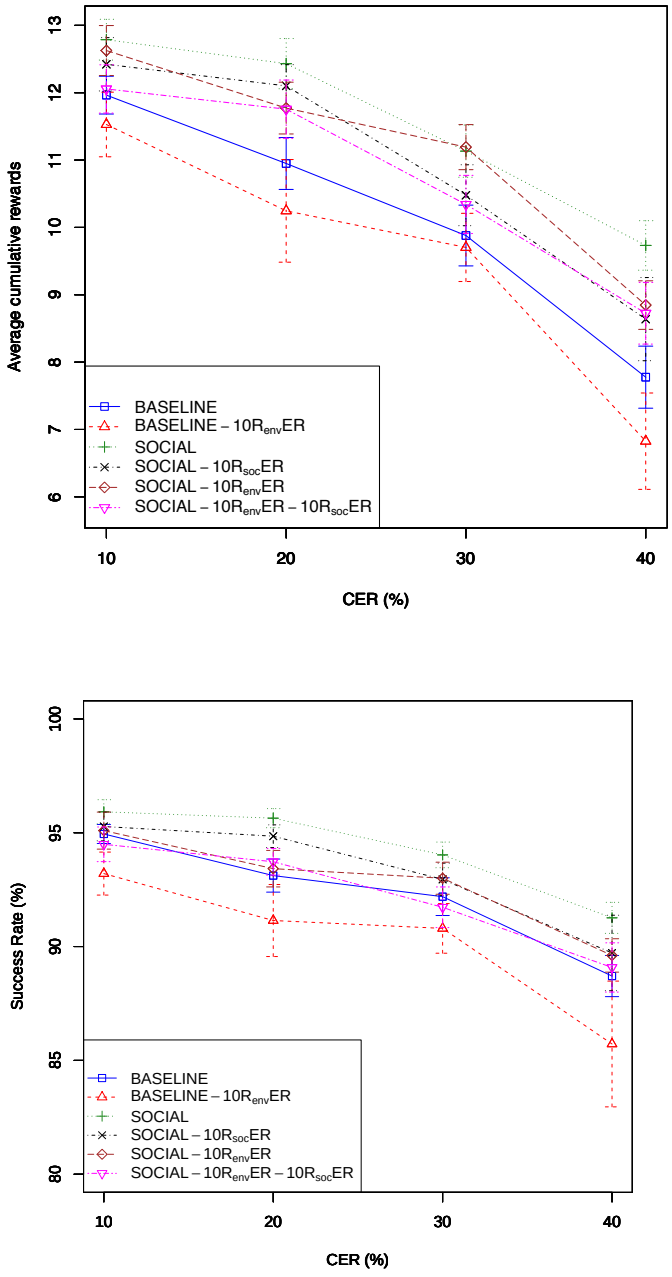


FIGURE 4.7 – *Résultats de la méthode KTD-Q de référence et des méthodes socialement inspirées dans différentes conditions de bruits (tests)*

du système au cours du dialogue peut conduire l'utilisateur à pénaliser le système dans le questionnaire final. Une autre explication possible est que les utilisateurs participant à ce type d'expériences ne sont pas réellement engagés dans la tâche qu'ils poursuivent. En effet, le plus souvent il s'agit là de tâches artificielles, créées uniquement pour les besoins d'une expérience. Dans ces conditions, si le système ne parvient pas à remplir

l'objectif visé, les conséquences pour ces utilisateurs sont nulles. De même, si le système propose de lever une contrainte et que cela n'est pas prévue dans le scénario, les utilisateurs n'ont pas de motivations personnelles leur permettant de guider efficacement la négociation.

Dans notre cadre expérimental nous avons fait le choix de simuler un niveau de taux d'erreur des récompenses de l'environnement (*Environment Reward Error Rate* - $R_{env}ER$) pour refléter ce phénomène. Ainsi, selon une certaine fréquence le critère binaire sur la réussite ou l'échec du dialogue verra sa valeur inversée. Il est important de noter que les $R_{env}ER$ et $R_{soc}ER$ sont ici simulés sans aucune hypothèse préalable spécifique. En effet, une approche générant des erreurs aléatoires est employée. Dans une étude plus poussée des modèles d'erreur pourraient être appris par le biais de données. Nous avons donc comparé sept configurations d'apprentissage : **BASELINE** and **BASELINE-10$R_{env}ER$**, **SOCIAL**, **SOCIAL-10$R_{env}ER$**, **SOCIAL-10$R_{soc}ER$** et **SOCIAL-10$R_{env}ER$-10$R_{soc}ER$**. Ici 10XER signifie que le taux d'erreur sur la reward X est fixé à un niveau de 10%.

Les résultats sont présentés dans la figure 4.7 en termes de récompenses cumulées et taux de réussite par rapport à différents niveaux de CER. Pour ces courbes, chaque point est une moyenne faite sur les résultats obtenus en utilisant 50 politiques apprises avec 400 dialogues, puis testés avec 1000 dialogues. Dans la configuration de test, la prochaine action est choisie de façon gloutonne par rapport à la Q-fonction apprise (exploitation uniquement).

Si ne sont considérés que les résultats de **BASELINE** et **BASELINE10$R_{env}ER$**, l'influence du CER et du $R_{env}ER$ est facilement identifiable. En effet, plus le $R_{env}ER$ et le CER augmentent, plus le niveau de performance global diminue. Les performances de **BASELINE** peuvent être comparées à celles obtenues avec **SOCIAL** et **SOCIAL-10$R_{soc}ER$**. En effet, à l'exception de l'utilisation des récompenses sociales (bruitées ou non) pour les deux dernières, ces trois configurations ont été apprises dans des conditions similaires. Globalement, nous observons que les deux méthodes exploitant les récompenses sociales atteignent un meilleur niveau performance que **BASELINE** quelque soit le niveau de CER. Cependant, si on tient compte des déviations standards, nous ne pouvons affirmer cela que pour **SOCIAL**, à l'exception des résultats à 20% CER où cela est vrai pour les deux méthodes sociales. Nous pouvons également remarquer que la baisse en termes de performance entre 10 et 40% CER est moins importante pour la méthode **SOCIAL** que pour **BASELINE** (respectivement -3 points de récompense contre -4). De façon similaire, la configuration **BASELINE-10$R_{env}ER$** peut être comparée à **SOCIAL-10R_{env} ER** et **SOCIAL-10$R_{env}ER$-10$R_{soc}ER$**. Nous observons également que les deux méthodes sociales considérées atteignent une meilleure performance que celle de référence pour tous les niveaux de CER. Cette fois, nous pouvons affirmer cela à 20 et 40% CER pour les deux méthodes sociales.

Ainsi, dans toutes les configurations étudiées, l'utilisation des signaux sociaux dans le mécanisme d'apprentissage a un impact positif sur la performance de l'algorithme KTD-Q. Le guidage introduit par les évaluations subjectives émises par l'utilisateur aux travers des récompenses sociales améliore la robustesse aux bruits, CER et $R_{env}ER$,

Social ?	$R_{soc}ER$	Récompenses cumulées	Taux de réussite
non	-	10.24 ($\pm$0.76)	91.14 ($\pm$1.58)
oui	0	11.77 ($\pm$0.38)	93.42 ($\pm$0.80)
oui	10	11.75 ($\pm$0.43)	93.73 ($\pm$0.58)
oui	20	11.28 ($\pm$0.45)	92.53 ($\pm$0.88)
oui	30	10.80 ($\pm$0.42)	91.68 ($\pm$1.10)
oui	40	10.67 ($\pm$0.43)	91.33 ($\pm$1.01)
oui	50	10.06 ($\pm$0.71)	89.34 ($\pm$3.70)

TABLE 4.7 – *Résultats en tests de le l'algorithme KTD-Q à* 20% *de CER et* 10% *de* $R_{env}ER$ *avec différents niveaux de* $R_{soc}ER$ *sur TownInfo.*

et même dans le cas où la récompense sociale est elle-même sujette au bruit $R_{soc}ER$. Là encore, l'aspect diffus introduit par les récompenses sociales semble bénéfique à l'apprentissage. Par exemple, dans le cas où l'utilisateur donne une récompense finale erronée, les récompenses sociales positives et négatives recueillies jusqu'à lors peuvent contrebalancer cette erreur (comme un indice de la satisfaction globale de l'utilisateur). En outre, en cas de fort CER, les récompenses sociales peuvent favoriser ou pénaliser le comportement local du système et ce malgré l'échec ou la réussite de la tâche globale. Toutefois, l'apport de l'approche par renforcement socialement inspiré diminue lorsque le niveau de $R_{soc}ER$ augmente.

Afin d'étudier l'impact de $R_{soc}ER$ seul, le tableau 4.7 présente les résultats obtenus avec différents niveaux $R_{soc}ER$. Nous avons choisi de faire ce comparatif à 20% CER et 10% $R_{env}ER$. Au-dessus de 30% $R_{soc}ER$, tenir compte des récompenses sociales semble être inutile, voire désavantageux. En fait, même si les résultats obtenus avec 40% $R_{soc}ER$ sont légèrement meilleurs que ceux obtenus avec **BASELINE**, ils ne convergent pas aussi rapidement (par exemple, à 200 dialogues **BASELINE** surpasse la version sociale).

Capacité d'adaptation aux profils utilisateurs

Dans cette dernière section, les avantages de l'utilisation des signaux de renforcement socialement inspirés sont évalués pour l'adaptation de la politique aux profils des utilisateurs.

Dans cette étude deux types d'utilisateurs sont simulés en jouant sur la configuration du simulateur (voir section 4.2.2). Ici, nous employons différents taux de prise d'initiative (nombre de concepts renseignés dans les actes de dialogue utilisateur) et de patience (tolérance aux erreurs du système). La première configuration correspond à celle d'un utilisateur « novice » qu'on peut qualifier de patient (tolérant aux erreurs systèmes) et ayant une initiative limitée (fournit une à une les informations nécessaire au système et attend ses instructions). Le second profil est celui d'un utilisateur « avancé » ou « expert », défini cette fois comme impatient et disposant d'un haut niveau d'initiative (son taux d'initiative étant fixé à 80% au lieu de 30% pour le novice). Nous avons également adapté le calcul des récompenses socialement inspirées selon chaque profil ainsi défini pour refléter le caractère subjectif de cette métrique. Par exemple, le nombre de tours sera un critère négatif plus important dans la détermination du score ψ_{social}

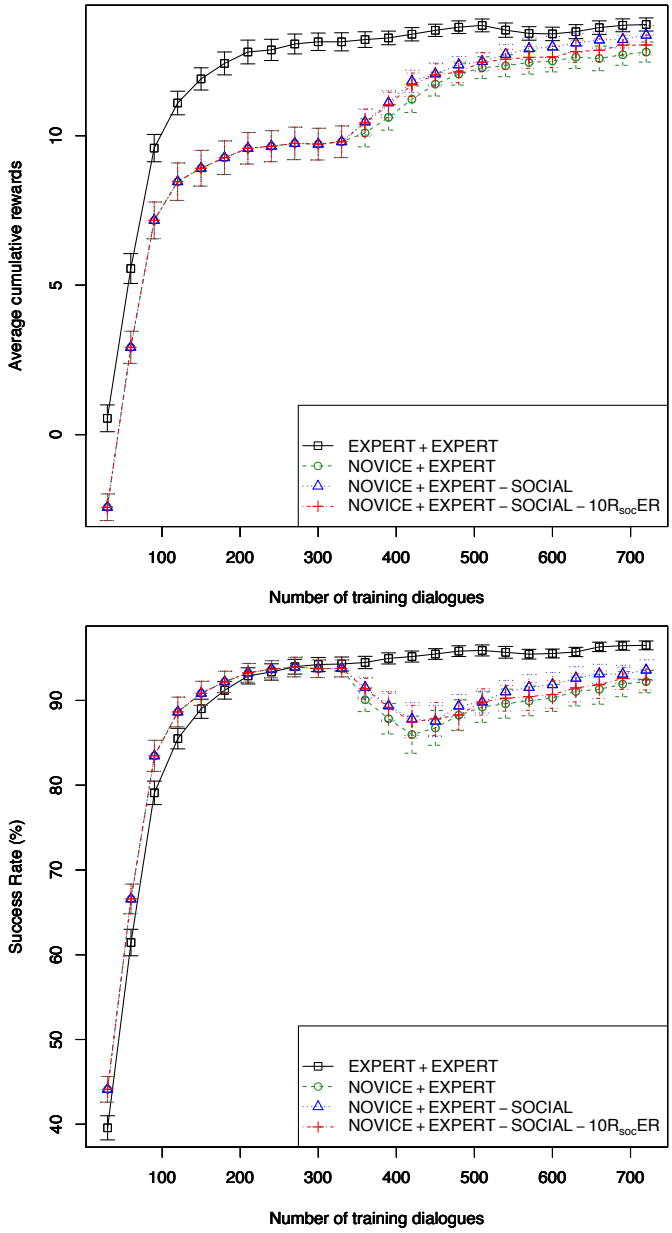


FIGURE 4.8 – *Résultats de la méthode KTD-Q avec et sans renforcement social selon différents profils d'utilisateurs simulés pour la tâche TownInfo (condition d'apprentissage).*

pour l'utilisateur expert que pour le novice. En effet un expert veut accomplir sa tâche de façon efficiente.

Les expériences peuvent être décomposées en deux étapes principales. Dans la première étape, l'algorithme KTD-Q est employé pour apprendre deux types distincts de politiques : celles apprises avec l'utilisateur expert simulé et celles apprises avec le novice. En raison de la convergence rapide de KTD-Q (efficace par échantillon) cet appren-

tissage se fera sur 400 dialogues. Dans la deuxième étape, l'apprentissage des politiques se poursuivra en interagissant exclusivement avec l'utilisateur expert pendant 400 dialogues dans toutes les configurations. Dans ces conditions, quatre scénarios sont ainsi identifiés :

— **EXPERT+EXPERT** : pour les politiques apprises précédemment en interagissant avec l'utilisateur simulé expert, ces nouvelles interactions poursuivent simplement le processus apprentissage engagé précédemment (pas de changement de comportement).
— **NOVICE+EXPERT** : le deuxième scénario diffère du précédent par le fait qu'un changement dans les comportements de l'environnement intervient. En effet, dans ce scénario, comme dans les scénarios suivants, les politiques précédemment apprises en interagissant avec l'utilisateur simulé novice doivent s'adapter. Ce scénario particulier constitue notre méthode d'adaptation de référence.
— **NOVICE+EXPERT-SOCIAL** : le troisième scénario vise à introduire des récompenses sociales avec l'espoir qu'elles aiderons l'agent d'apprentissage à s'adapter au nouvelles dynamiques de l'environnement plus rapidement.
— **NOVICE+EXPERT-SOCIAL-10$R_{soc}ER$** : la dernière expérience est dérivée du troisième scénario. Cette fois cependant les récompenses sociales sont bruitées artificiellement pour simuler un taux d'erreur de 10% pour simuler des configurations réalistes où les indices des récompenses sociales ne pourront être capturés et interprétés parfaitement.

Les comparaisons sont présentées en termes de moyenne mobile sur les récompenses de l'environnement cumulées et de taux de réussite sur la figure 4.8. Chaque point sur ces figures est le résultat d'une moyenne faîte sur une fenêtre glissante de largeur 100 points. Ces résultats sont analysés comme suit. Les 400 premiers dialogues sont utilisés pour expliquer l'impact du profil utilisateur dans le processus d'apprentissage (soit la différence entre le fait d'apprendre avec un utilisateur expert et un utilisateur novice). Les 400 autres dialogues (de 400 à 800) servent à comparer les capacités d'adaptation avec et sans récompenses sociales, à la fois dans une configuration sans bruit et avec 10% de $R_{soc}ER$.

Lorsque seuls les 400 premiers dialogues d'apprentissage sont considérés, les résultats présentés dans la figure 4.8 montrent que de meilleures récompenses sont obtenues en interagissant avec l'utilisateur avancé (+3 par rapport à un apprentissage réalisé avec un novice en moyenne). Ceci s'explique par le fait que la stratégie d'interaction de l'expert simulé est plus efficace puisqu'il est configuré pour transmettre plus d'informations dans un même tour de dialogue que l'utilisateur novice. Les résultats sont par contre assez semblables lorsqu'on considère le taux de réussite. Ainsi les deux profils atteignent leur objectif dans la même proportion mais à un rythme différent (efficacité). Néanmoins une convergence légèrement plus rapide est observable lorsqu'il s'agit de l'utilisateur novice (+2% en moyenne autour de 100 dialogues). Ce constat peut être expliqué par le fait que l'utilisateur avancé est moins patient que celui novice, de sorte que le système doit faire face à plus d'échecs en début d'apprentissage. Il semble donc plus facile d'apprendre une politique intermédiaire face à un utilisateur plus flexible que face à un utilisateur exigeant.

Pour les dialogues restants (400-800), une baisse globale de la performance est observable en termes de taux de réussite dans la figure 4.8. En effet, lorsque des changements surviennent dans les dynamiques de l'environnement (ici simulé par la modification de la configuration de l'utilisateur simulé) la politique apprise ne cadre plus avec les nouveaux comportements de l'utilisateur. Cependant, cette baisse est suivie par d'une phase d'adaptation qui ramène la performance globale proche de la courbe de référence, **EXPERT+EXPERT**.

En ce qui concerne les performances en termes de récompenses cumulées moyenne, la figure 4.8 montre que toutes les courbes considérées sont proches de la référence. Aucune baisse comme celle observée en terme de taux de réussite n'est visible sur ces courbes car les utilisateurs experts effectuent les tâches plus efficacement que les novices. En effet, ces derniers prennent moins de tours pour effectuer leurs tâches de dialogue et donc ils permettent à l'agent apprenant de collecter une plus grande récompense cumulée de la part environnement à la fin de chaque tâche réussite (moins de pénalités du au nombre de tour) mais également une pénalité plus faible dans des situations d'échec (l'utilisateur étant moins patient il mettra fin à la conversation avant). Par conséquent, la perte en termes de taux de réussite est en quelque sorte « compensée ».

Comme on peut le constater, lorsque le renforcement socialement inspiré est utilisé, le processus d'adaptation est légèrement plus efficace que dans le scénario de référence. En effet, la perte en termes de taux de réussite n'est pas aussi importante que lorsque ces récompenses ne sont pas impliquées ($+1,5\%$ en moyenne). De plus, autant en termes de taux de réussite que de récompenses cumulées moyenne, le niveau de performance converge un peu plus vite en phase d'adaptation que celui de **BASELINE**. Ces résultats peuvent être expliqués par le fait que les récompenses sociales sont collectées tout au long du dialogue et offrent une fonction de récompense avec une bonne granularité. Ces récompenses peuvent ainsi favoriser ou pénaliser de façon plus locale un comportement spécifique du système (de réaction état-action) et ce malgré l'échec ou le succès de la tâche. Ainsi, ces récompenses permettent donc de favoriser l'adaptation et de compenser rapidement la perte de performance due à l'apparition d'un nouveau comportement.

Les résultats obtenus par **NOVICE+EXPERT-SOCIAL-10$R_{soc}ER$** montrent qu'en présence de récompenses sociales légèrement bruitées les performances sont malgré tout encore améliorées par rapport à la référence tant en termes de récompenses cumulées moyenne que de taux de réussite moyen (respectivement $+0,4$ et $+0,8\%$). Au début de la phase d'adaptation, la perte en termes de taux de réussite n'est pas aussi forte que dans le scénario de référence. En effet elle présente un niveau de performance similaire à celui de la version sociale exempte de bruit. Cependant, pour le reste de la procédure d'adaptation, la méthode bruitée est tout de même moins performante que cette dernière. L'avantage apporté par le renforcement socialement inspiré semble donc diminuer lorsque le $R_{soc}ER$ grandit. Cependant, la convergence vers la politique quasi-optimale est préservée grâce à la technique de *reward shaping* en cas de récompenses sociales très bruitées ($R_{soc}ER$ élevé).

Bilan intermédiaire

Pour conclure cette seconde étude, nous pouvons dire qu'une preuve de concept a été apportée montrant l'intérêt de considérer l'évaluation subjective dans l'apprentissage. Cependant, dans les conditions expérimentales qui ont été les nôtres les signaux sociaux ont été simulés, ce qui nécessite des études complémentaires pour être validé en pratique. Pour cela, nous présenterons dans le chapitre 6 un premier usage de cette proposition avec des évaluations subjectives émises par de vrais utilisateurs durant le cours de l'interaction grâce à l'emploi d'une méthode permettant d'en simplifier la capture.

En pratique, pour envisager un passage à l'échelle de cette technique, des mécanismes pour détecter des indices multimodaux et procéder à leur analyse devront être envisagés pour passer de la preuve de concept à l'application concrète de la technique. Si dans cette thèse nous nous sommes limités à l'étude de ces mécanismes, nous pouvons tout de même orienter le lecteur vers les travaux présentés dans l'*Interspeech Computational Paralinguistics Challenge* (Schuller et al., 2013). Même si la robustesse de ces approches est discutable dans le cas où les indices sociaux sont détectés de façon non contrainte et implicite (Vinciarelli et al., 2009), les expériences conduites précédemment montrent cependant qu'ils pourront être détectés avec un certain niveau d'imprécision sans remettre en cause le bien-fondé de la méthode proposée.

En outre, en cas de conflit entre les récompenses socialement inspirées et environnementales (du notamment à la nature subjective des évaluations de l'utilisateur), la technique de *reward shaping* basée sur les potentiels prévient d'une dégradation drastique des performances en cas de signaux de récompenses additionnels inappropriés. Ce problème peut cependant être simplifié si l'on considère une interaction avec un utilisateur « expert/tuteur » agissant de façon coopérative et rationnelle avec le système (par exemple, le concepteur de système) et à qui il sera offert au fil de l'interaction le moyen d'évaluer le système grâce à un jeu de signaux limités (gestes spécifiques). C'est d'ailleurs sur cette logique que nous intégrerons cette technique pour la tâche *MaRDi* dans le chapitre 6.

Comme signalé dans la section 4.2.3 on peut envisager une approche faisant usage de données d'apprentissage pour remplacer les paramètres artisanaux représentés dans la section 4.2.2 avec des indices plus sophistiqués. Par exemple, les méthodes de régression telles que celle employée dans (Rieser et Lemon, 2011) pour déterminer la fonction de récompense immédiate optimale sur un corpus de dialogues annotés obtenu par WoZ. Dans le cas présent, l'effort de collecte pourrait être effectué sur plusieurs tâches, puisque les récompenses sociales sont conçues pour exprimer le jugement de l'utilisateur sur l'état d'avancement de dialogue, ce qui constitue un critère relativement indépendant de la tâche. De plus, des techniques telles que celles employées dans (Schmitt et al., 2011; Ultes et al., 2015) pour estimer tout au long du dialogue une métrique traduisant de la qualité du déroulement de l'interaction pourraient être employées de façon transparente dans notre proposition.

Un autre point important est que l'information sociale permet une vision plus gra-

nulaire de la fonction de récompense plutôt qu'un jugement unique, à la fin de l'épisode. Il peut aider à éviter ou à renforcer certains comportements locaux du système et peut être utilisé pour mieux gérer le problème de l'adaptation de l'utilisateur du fait que ce signal de renforcement est non seulement basé sur des mesures objectives mais aussi sur les indices subjectifs provenant de l'utilisateur. Ainsi, lorsque des algorithmes efficaces par échantillons sont considérés une telle approche peut être employée comme un moyen d'éviter la nécessité d'un simulateur d'utilisateurs, comme nous illustrerons pour répondre à la tâche *MaRDi*. En ce sens, une telle configuration peut notamment être liée à de l'apprentissage actif comme ce qui est fait dans (Doshi et Roy, 2008) ou à des techniques d'imitation comme dans (Price et Boutilier, 2003).

4.4 Bilan

Dans ce chapitre nous avons vu un ensemble de méthodes permettant d'outiller un apprentissage RL pour mieux faire face aux problématique d'apprentissage de zéro et d'adaptation (**P1** et **P2**). Ceci a été réalisé au travers de l'utilisation des connaissances expertes a priori sur le domaine et d'informations subjectives capturées au fil de l'interaction. Ces travaux ont également permis de souligner l'intérêt de considérer une récompense plus diffuse grâce au *reward shaping* pour mieux faire face aux conditions bruitées.

Il est à noter que le choix de l'outil KTD dans cette étude n'est pas limitant, il sera ainsi possible d'appliquer nos diverses propositions (*reward shaping* expert/social et schéma d'exploration expert-glouton) à d'autres algorithmes RL efficaces en ligne, comme par exemple GPTD (Engel et al., 2003; Gašić et al., 2010). En ce qui concerne le recours au schéma d'exploration expert-glouton, ce dernier nécessite (en plus des règles) l'utilisation d'un algorithme capable de fournir la variance sur ses estimations des valeurs de la fonction de qualité (Q-valeurs).

Au travers de nos deux études, nous avons montré à quel point la qualité de la chaîne de compréhension pouvait avoir un impact sur l'apprentissage. Or, lorsqu'un système est développé de zéro, comme ce sera le cas pour *MaRDi*, les données nécessaires au développement d'un modèle SLU très robuste sont souvent manquantes. Réduire le coût de développement d'un tel système (le moins de données possible) tout en conservant une qualité suffisante pour permettre un apprentissage de qualité semble donc primordial et nous avons décider d'y consacrer une étude complète.

Chapitre 5

Compréhension de la parole sans données de références

Sommaire

Actuellement, les systèmes de l'état de l'art pour la compréhension de la parole sont basés sur des approches probabilistes et sont appris grâce à différentes méthodes d'apprentissage automatique afin de pouvoir attribuer des étiquettes sémantiques aux entrées des utilisateurs. Si elles présentent de bonnes propriétés générales qui justifient leur large déploiement, il reste que les techniques d'apprentissage supervisé requièrent cependant un grand nombre de données annotées dans la forme de représentation sémantique retenue pour la tâche visée. Ces corpus annotés sont à la fois coûteux en expertise humaine et en temps de construction. De plus ils sont dépendants du domaine applicatif (souvent restreint) ainsi que de la langue employée.

Plusieurs études ont comparé les différentes approches probabilistes pour la compréhension de la parole, par exemple (Lefèvre, 2007; Hahn et al., 2010; Deoras et Sarikaya, 2013). Les approches état de l'art utilisent le plus souvent des modèles statistiques

discriminants, tels que les CRF (Wang et Acero, 2006), les réseaux de neurones profonds DBN (Deoras et Sarikaya, 2013) ou récurrents (Yao et al., 2013; Mesnil et al., 2013) ou encore R-CRF (Yao et al., 2013, 2014). Malgré leurs bonnes performances, ces approches ont en commun d'être très dépendantes de la quantité et de la nature des données qu'elles utilisent pour l'apprentissage et sont donc difficilement généralisables sur de nouvelles tâches.

Dans ce chapitre nous présentons une méthode visant à limiter le besoin en données annotées qu'ont en commun les méthodes état de l'art de compréhension automatique de la parole et qui représente un obstacle lors du développement d'un système pour une nouvelle tâche ou une nouvelle langue.

Nous commencerons par présenter les méthodes proposées dans la littérature pour limiter les coûts de développement d'un module de compréhension dans la section 5.1. Puis dans la section 5.2 nous présenterons le mécanisme d'apprentissage sans données de référence (en anglais *zero-shot learning*) retenu dans notre étude pour répondre à cette problématique. Cette méthode combine une description ontologique minimale de la tâche visée avec l'utilisation d'un espace sémantique continu appris par des approches à base de réseaux de neurones à partir de données génériques non-annotées. Nous proposons ensuite une stratégie permettant d'adapter le modèle en ligne. L'idée étant d'améliorer les performances de notre approche à l'aide d'une légère supervision, ajustable par l'utilisateur. Nous proposons pour cela de guider le processus d'adaptation en ligne par l'intermédiaire d'une politique apprise en utilisant l'algorithme du bandit contre un adversaire sur information partielle. Enfin, nous établirons une extension du modèle initial dans un cadre supervisé plus standard (ici les CRF).

Dans la section 5.3, nous présenterons les résultats obtenus sur une tâche état de l'art de compréhension de la parole. Nous montrerons notamment que le modèle simple et peu coûteux proposé peut atteindre, dès le démarrage, des performances comparables à celles de systèmes état de l'art reposant sur des règles expertes ou sur des approches probabilistes. Nous montrons également l'intérêt de considérer une stratégie adaptation du modèle et de son optimisation en ligne visant à équilibrer le coût de la supervision réalisée par les utilisateurs et la performance générale du modèle. Enfin nous présenterons les résultats de notre extension du modèle au cadre supervisé et ce au travers des différentes configurations.

5.1 Limiter les coûts de développement d'un nouveau module de compréhension

Comme mentionné précédemment, les différentes techniques d'apprentissage statistique état de l'art ont en commun d'être très dépendantes de la quantité et de la nature des données qu'elles utilisent pour l'apprentissage. Pour faire face à cette limite, plusieurs études ont proposé un processus d'annotation non-supervisé. Par exemple en se basant sur la détection automatique de concepts par le biais des espaces de thèmes issus d'une allocation latente de Dirichlet (*Latent Dirichlet Allocation* - LDA) comme

cela est proposé dans (Camelin et al., 2011). D'autres travaux ont employé des algorithmes d'apprentissage non-supervisé (Tur et al., 2011; Lorenzo et al., 2013) ou semi-supervisé (Celikyilmaz et al., 2011; Hakkani-Tur et al., 2011) pour palier à l'absence de ressources annotées en exploitant notamment le web sémantique pour permettre une recherche de données d'apprentissage supplémentaires afin d'améliorer les performances des classifieurs employés.

Un autre groupe d'études s'est intéressé à proposer des techniques visant à réduire le temps de collecte, de transcription et d'annotation de nouveaux corpus. Par exemple, dans (Gao et al., 2005) ou encore dans (Sarikaya, 2008), il a été proposé de construire au préalable un petit corpus pour initialiser un système pilote visant l'acquisition de nouvelles données pour raffiner le modèle initial. D'autres travaux, tels que ceux présentés dans (Tur et al., 2003, 2005), ont employé des techniques issues de l'apprentissage actif (*Active Learning* - AL) pour réduire le temps nécessaire à l'annotation et à la vérification d'un corpus. Plusieurs recherches ont été conduites pour diminuer le coût et l'effort de collecte de données par l'étude de portabilité de systèmes à travers les langues et les domaines (Minker, 1998; Lefèvre et al., 2010; Huet et Lefèvre, 2011; Misu et al., 2012; Lefèvre et al., 2012; Jabaian et al., 2013; Chowdhury et al., 2014).

Ayant toujours le même objectif de minimiser le besoin en données d'apprentissage, coûteuses en temps et en expertises humaines, différentes approches ont déjà été employées dans la littérature pour exploiter les connaissances structurés (sous forme de graphes) du web sémantique pour des tâches de classification d'énoncés. Par exemple, les auteurs de (Heck et Hakkani-Tur, 2012) ont proposé une approche non-supervisée pour la compréhension de la parole s'appuyant sur une procédure d'extraction automatique d'exemples d'apprentissage sur le web à partir des triplets entité-relation-entité présents dans le graphe de connaissances manipulé, par exemple *Avatar-Directed By*-James Cameron.

Dans (Anastasakos et Deoras, 2014) les auteurs ont proposé d'exploiter un espace continu pour modéliser les mots appris de façon non-supervisée sur des données du domaine visé. L'idée étant d'obtenir des représentations vectorielles spécifiques au domaine applicatif pour apprendre un système de compréhension. Ils ont également proposé de transférer ces représentations d'une langue à une autre pour permettre l'apprentissage d'un système de compréhension multilingue.

Une autre piste de recherche consiste en l'étude d'une solution permettant d'adapter les modèles en ligne. Par exemple, dans (Bayer et Riccardi, 2013) une approche récupérant les exemples d'apprentissage les plus proches de l'énoncé utilisateur en termes de distance d'édition et de correspondance n-gram. Ces exemples sont employés pour rescorer localement la liste de n-meilleures hypothèses en sortie du module de compréhension (voir également celle en sortie de l'ASR). Une autre solution présentée dans (Gotab et al., 2010) propose de faire confirmer directement auprès d'utilisateurs les prédictions du système sur une tâche d'identification de motifs d'appel. Les utilisateurs ayant juste pour rôle de spécifier à chaque prédiction du système si celle-ci lui paraît juste ou fausse (supervision limitée). Ces retours sont ensuite utilisés pour mettre à jour dynamiquement un ensemble de classifieurs binaires (un par motif d'appel considéré).

Une autre solution à cette problématique consiste à considérer le cadre formel de l'apprentissage sans données de référence (*zero-shot learning*). Ce paradigme a été introduit pour la première fois dans (Larochelle et al., 2008) et correspond à un cas particulier de l'apprentissage supervisé où certaines valeurs de l'ensemble des sorties possibles ne sont pas toutes couvertes dans les exemples du corpus d'apprentissage (petit corpus d'apprentissage, trop grand nombre de classes ou classes inconnues a priori). Pour pouvoir compenser ce manque de données étiquetées, le modèle s'appuie alors sur des descriptions minimales de ces valeurs manquantes pour pouvoir généraliser ses connaissances. Dans (Larochelle et al., 2008) les auteurs appliquent ce paradigme à une tâche de reconnaissance optique de caractères. Pour ce faire, une représentation standardisée de tous les caractères alphanumériques est employée. Elle permet de généraliser le modèle sur des images en provenance de plaques d'immatriculation ou d'écritures manuscrites où figurent des caractères absents des données d'apprentissage.

Les auteurs de (Palatucci et al., 2009) ont proposé une approche similaire pour identifier des images décrivant l'activité neuronale des mots auxquels les sujets expérimentaux pensent mais qui n'ont pas tous des occurrences dans les données d'apprentissage du classifieur employé. Pour ce faire, les auteurs s'appuient sur l'établissement d'une base de connaissances décrivant les propriétés sémantiques des mots cibles. Ainsi, chaque mot est représenté soit par un vecteur des co-occurences avec les mots les plus fréquents d'un corpus de grande taille (ici le *Google Trillion-Word-Corpus2*), soit sur la base de réponses utilisateurs à des questions sélectionnées pour refléter les propriétés de l'encodage neuronal chez l'Homme, comme par exemple « Pouvez-vous attraper cet objet ? », « Est-ce un objet créé par l'Homme ? ».

Concernant son application dans le cadre du NLP, dans (Dauphin et al., 2014) un tel mécanisme d'apprentissage est proposé pour réaliser une tâche de classification sémantique globale de l'énoncé utilisateur (par exemple en thèmes comme « films », « évènements », ou « restaurants »). Cette méthode tente de trouver un lien entre les thèmes cibles et les énoncés utilisateurs dans un espace sémantique dédié. Ce dernier est appris par un réseau de neurones profond sur une grande quantité de données non-annotées et non-structurées en lien avec la tâche visée (ici des *logs* de clics d'un moteur de recherche). Dans cette configuration, le réseau considéré prend en entrée une requête utilisateur (sous une forme de sac de mots) et doit la faire correspondre avec un site web pertinent (déterminé grâce aux *logs* de clics). Dans cette configuration, l'espace sémantique est en fait la couche cachée la plus profonde du réseau et il sera exploitable en mettant en entrée l'énoncé de l'utilisateur (également sous une forme de sac de mots). L'idée derrière la mise en place d'un tel réseau est qu'il puisse modéliser le fait que des requêtes soient sémantiquement « proches » lorsqu'elles ont suscité un clic utilisateur sur un même site. Par exemple, on peut raisonnablement supposer que des requêtes qui ont conduit l'utilisateur à se rendre sur le site web *imdb.com* partagent une relation sémantique tournant autour du cinéma. L'idée supplémentaire introduite dans (Dauphin et al., 2014) consiste à ajouter dans la fonction objectif employée dans l'apprentissage du réseau de neurones profond un critère d'optimisation additionnel plus en lien avec le but de la tâche visée (ici minimiser l'entropie relative aux classes considérées). Cette technique permet alors d'obtenir un espace plus discriminant sur les différentes sorties

possibles.

La proposition que nous faisons s'inscrit dans une ligne similaire à ces travaux mais avec pour objectif premier l'annotation sémantique complète des énoncés utilisateurs. De plus, nous nous distinguons des travaux précédents par la nature et la manière dont nous définissons notre représentation sémantique. En effet, dans notre cas nous n'avons pas l'intention d'exploiter des données particulièrement reliées au domaine applicatif, puisque nous partons du principe que ces données peuvent nous faire défaut dans une situation d'un développement de zéro selon le système de dialogue considéré. Ainsi, nous nous reposerons uniquement sur une représentation apprise sur des données généralistes (plus nombreuses et plus facilement accessibles), pour exploiter au mieux les quelques connaissances initiales (limitées voire incomplètes) du domaine.

5.2 Solution d'apprentissage sans données de référence pour la compréhension

Dans ce manuscrit nous présentons une méthode visant à limiter la dépendance aux données annotées par l'utilisation d'un mécanisme similaire à celui proposé dans (Dauphin et al., 2014). En effet, notre méthode repose sur une description ontologique minimale de la tâche visée et sur l'utilisation d'un espace sémantique continu appris par des approches à base de réseaux de neurones sur des données génériques non-annotées (facilement disponible sur web). Nous consacrons la section 5.2.1 à la description du modèle initial pour répondre à la problématique du démarrage de zéro du SLU.

Toutefois l'approche proposée reste dépendante de la qualité de la description ontologique utilisée et de l'espace sémantique continu considéré (sa capacité à modéliser la richesse sémantique du domaine cible). Pour faire face à ces deux limites, nous proposons l'ajout d'une stratégie d'adaptation « en ligne ». Cette approche complémentaire a pour objectif d'introduire une faible supervision dans l'optique de raffiner de façon incrémentale la définition de notre connaissance ontologique et de mieux exploiter l'espace sémantique considéré. Afin de régler le rapport entre coût de la supervision et apport sur le modèle nous proposons l'emploi d'une stratégie d'adaptation en ligne s'appuyant sur la modélisation du problème d'AL sous la forme d'un bandit contre un adversaire.

De même, nous étudierons le liens qui peut exister entre notre méthode et une approche supervisée performante plus classique, comme le modèle CRF (Lafferty et al., 2001). Pour cela, nous proposons dans la section 5.2.3 une extension de notre approche dans un formalisme reposant sur des transducteurs à états finis pour faciliter l'intégration de tels modèles.

5.2.1 Description de l'approche initiale

Dans cette étude, nous examinons le problème de prédire la séquence d'actes de dialogue d'un énoncé utilisateur sans avoir vu au préalable le moindre exemple obtenu

lors d'une véritable interaction. Pour ce faire, une source de connaissance sémantique doit être exploitée pour extrapoler ces sorties à partir de leur définition. Notre méthode se base donc sur trois composants principaux :

— un espace sémantique continu noté F qui peut être défini comme un espace de dimension d à même de coder les différentes propriétés sémantiques nécessaires à la tâche ;
— une base de connaissances K qui peut être vue comme un dictionnaire d'exemples dans F. Elle est utilisée pour relier l'espace sémantique à l'espace de sortie du système ;
— un analyseur sémantique qui extrait une liste ordonnée des meilleures hypothèses de séquence d'étiquettes sémantiques à partir d'un transducteur à états finis représentant l'ensemble des hypothèses reliées à un énoncé utilisateur (ces dernières étant scorées par des informations issues de F et de K).

Dans la suite de cette section nous décrivons plus en détail ces différents composants. Cependant, les choix faits quant à leur implémentation concrète pour la tâche visée seront donnés dans la partie expérimentale.

Espace sémantique continu

De avancées récentes sur les réseaux de neurones, ont permis d'envisager l'apprentissage de diverses représentations vectorielles compactes de mots (*word embedding*) présentant des régularités notables avec les propriétés syntaxiques et sémantiques des mots qu'elles modélisent (Mikolov et al., 2013a; Bian et al., 2014). Des travaux ont déjà pu montrer l'intérêt de considérer ce type de représentation sur différentes tâches de traitement automatique des langues naturelles (Bengio et Heigold, 2014; Clinchant et Perronnin, 2013).

L'objectif du module de compréhension étant d'extraire des informations sémantiques à partir d'entrées utilisateur en langage naturel, l'utilisation d'une telle représentation pour définir l'espace sémantique continu offre des possibilités de généralisation d'un grand intérêt.

De plus, ce type de représentation ne repose pas explicitement sur l'exploitation de données liées à la tâche, mais au contraire sur un apprentissage réalisé sur une très grande quantité de données (de large couverture) souvent plus facilement accessible (le dump Wikipédia étant un exemple possible). De plus, différentes techniques, comme celle présentée dans (Zou et al., 2013), permettent d'adapter/de transférer le modèle ainsi appris pour l'appliquer à une tâche spécifique ou encore dans une autre langue.

En l'état, notre proposition n'est pas dépendante d'un type particulier de représentation mais seulement de ses capacités de généralisation et de sa facilité d'usage. Dans nos travaux nous avons fait le choix d'exploiter le modèle *word2vec*.

Word2vec ce modèle, proposé initialement dans (Mikolov et al., 2013a), repose sur l'utilisation d'un réseau de neurones artificiels pour apprendre une représentation vectorielle continue des mots qui sera capable de capturer des régularités sémantiques et

syntaxiques (Mikolov et al., 2013c). Cette approche repose sur l'adoption d'une architecture neuronale simple mais également sur des simplifications calculatoires permettant d'exploiter efficacement une très grande quantité de données textuelles pour son apprentissage. Cette solution permet d'obtenir des résultats comparables (si ce n'est meilleurs) à des représentations obtenues par des modèles bien plus complexes, par exemple les réseaux récurrents ou de convolution (Mikolov et al., 2013b).

Deux architectures neuronales alternatives peuvent être employées dans cette modélisation : une dénommée *CBOW* et l'autre *Skip-gram* dans la littérature. Elles sont toutes deux illustrées dans la figure 5.1. Comme on peut le voir sur cette figure, ces deux architectures sont simples puisqu'elles utilisent seulement trois couches, à savoir une couche d'entrée, une couche cachée et une couche de sortie.

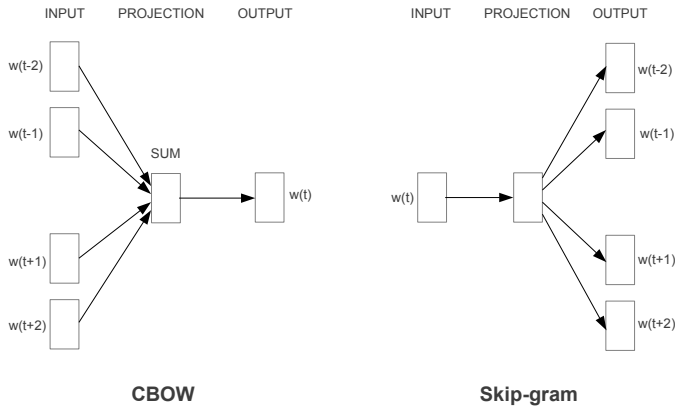


FIGURE 5.1 – *Architectures CBOW et Skip-gram du modèle word2vec, image extraite de (Mikolov et al., 2013a).*

L'objectif de l'approche *CBOW* est de prédire un mot à partir de son contexte d'apparition. La couche d'entrée de ce réseau représente la présence ou l'absence des mots dans le contexte de manière binaire (1 pour la présence, 0 pour l'absence). La couche cachée correspond à la projection des mots en entrée dans une matrice de poids. Cette matrice est partagée par tous les mots et constitue le modèle *word2vec*. Chaque mot du contexte est donc projeté dans cette matrice. La somme de ces représentations est ensuite considérée pour réaliser la prédiction. Pour ce faire, la couche de sortie emploie un modèle de classification log-linéaire (*softmax*) afin de déterminer le mot courant. Cette architecture présente l'avantage d'être plus efficiente du point de vue algorithmique que l'approche *Skip-Gram*. De plus, elle semble plus efficace dans la modélisation des mots fréquents mais aussi pour capturer les relations syntaxiques entre les mots (Mikolov et al., 2013c).

L'objectif de *Skip-gram* est de prédire, pour un mot donné, le contexte dont il est issu (mots environnants). La couche d'entrée de ce réseau est cette fois un vecteur ne contenant que le mot. La couche cachée est donc similaire à *CBOW*, sauf que la somme n'est plus nécessaire pour réaliser la prédiction. La couche de sortie correspond donc à la concaténation d'un modèle *softmax* par mot à prédire. Comparativement à l'ap-

proche *CBOW*, cette architecture permet de mieux modéliser les mots peu fréquents et de capturer plus efficacement les relations sémantiques (Mikolov et al., 2013c). C'est d'ailleurs essentiellement pour ces deux raisons que nous avons adopté cette architecture dans nos travaux. En effet, selon la tâche de dialogue considérée, l'utilisateur peut faire usage de mots très spécifiques et donc peu fréquents dans les données généralistes employées pour apprendre la représentation. De plus, ce sont essentiellement des relations sémantiques que nous cherchons à exploiter.

Pour faire face à la complexité algorithmique engendrée par la taille du contexte et du vocabulaire considérés, il est proposé dans la littérature deux alternatives pour simplifier le calcul : le *softmax hiérarchique* (Morin et Bengio, 2005; Mikolov et al., 2013a) et l'échantillonnage négatif (Mikolov et al., 2013b). Dans notre étude nous avons fait l'usage exclusif de la première stratégie.

Concernant les régularités sémantiques et syntaxiques qui nous intéressent particulièrement dans notre étude, il a été montré dans (Mikolov et al., 2013c) que les angles observés entre les projections des mots (similarité cosinus) sont corrélés aux relations complexes qui les relient, telles que « féminin-masculin », « singulier-pluriel » (analogies syntaxiques) ou encore « pays-capitale » et « film-réalisateur » (analogies sémantiques). De même, ces travaux ont montré qu'il est possible d'exploiter ces relations avec des opérations arithmétiques simples sur ces vecteurs (addition et soustraction), comme illustré dans l'ensemble suivant :

$$vector(king) - vector(man) + vector(woman) \approx vector(queen)$$

Base de connaissance sémantique

La base de connaissance sémantique K est définie comme la matrice d'affectation représentant les informations ontologiques du domaine visé. Dans notre étude, ces dernières se limitent à la liste des étiquettes sémantiques possibles et aux exemples de formes de surface qui leurs sont associées. Dans cette matrice (illustrée dans la figure 5.2), chaque ligne correspond à un vecteur d'exemple de dimension d dans F et chaque colonne à une étiquette sémantique. Ainsi la valeur de chaque cellule de la matrice (notée $c_{i,j}$ et appelée valeur d'affectation par la suite) indique s'il existe une éventuelle affectation entre le $i^{ème}$ vecteur dans l'espace sémantique F et la $j^{ème}$ étiquette sémantique.

Les exemples (entrées de la matrice) sont obtenus en projetant dans F un certain nombre de formes de surface associées à la description ontologique du domaine. Ces formes de surface peuvent être composées d'un ou plusieurs mots. Par exemple, pour la tâche TownInfo, « what food is served ? » pour `request(food)`, « yes » pour `affirm()` ou encore « french food » pour `inform(food=french)`.

Elles peuvent être obtenues automatiquement en se basant sur l'ontologie du domaine (guide d'annotation), la base de données associée à la tâche (extraction des valeurs possibles pour chaque concept) ainsi que sur un certain nombre d'exemples illustrant les différents types d'actes de dialogue génériques (donnés par un expert).

Il est à noter que la méthode employée ne nécessite en aucun cas d'être exhaustive lors de la définition de ces exemples (conjugaison des verbes dans tous les temps, utilisation de synonymes ou d'expressions équivalentes, formes singulières et plurielles, etc.). En effet, le recours à l'espace sémantique F permettra leur généralisation après coup. Ce comportement n'est bien sur pas accessible avec des approches à base de règles expertes/grammaire (dictionnaire de synonymes, etc.) traditionnelles, ce qui va impliquer un effort de conception sans commune mesure pour le concepteur selon la complexité de la tâche et de la richesse expressive de la langue visée.

Sur la figure 5.2, les lignes et colonnes de K sont respectivement identifiées par les formes de surface et les étiquettes sémantiques pour en faciliter la lecture. Les valeurs d'affection sont d'abord initialisées par des valeurs binaires, 1 si affectation et 0 sinon. Il est important de noter que nous ne contraignons pas la représentation actuelle de K par une correspondance unique entre une forme de surface et une étiquette sémantique. Ainsi, plusieurs valeurs d'affectation peuvent être mise à 1 sur une même ligne. Par exemple la forme de surface « Paris » pourrait très bien être à la fois affectée à l'étiquette sémantique `inform(location=Paris)` et à `inform(name=Paris)` si un établissement présent dans la base de données porte ce nom.

Analyseur sémantique

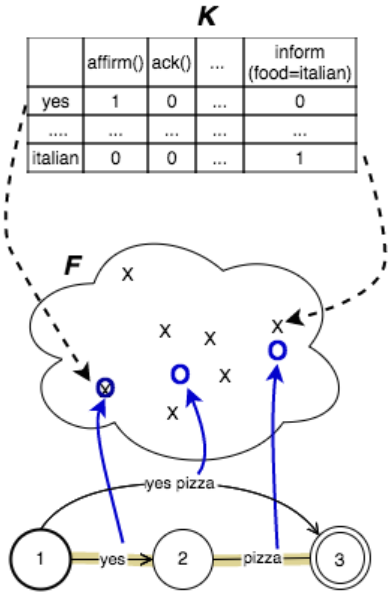


FIGURE 5.2 – *Illustration d'un décodage sémantique basé sur une technique d'apprentissage sans données de référence.*

En phase de décodage, pour chaque nouvelle phrase utilisateur que l'on cherche à étiqueter, toutes les séquences de mots contigus (formes de surface) sont considérées [1].

1. Ce choix s'explique notamment car pour les tâches traitées ici les énoncés utilisateur sont générale-

Par exemple pour la phrase « yes pizza », trois formes de surface différentes sont extraites : « yes », « pizza » et « yes pizza ». Ces formes de surface sont ensuite projetées dans l'espace sémantique F (cercles bleus dans la Fig. 5.2) pour être comparées aux vecteurs associés aux exemples de la base de connaissance K (croix noires dans la Fig. 5.2).

Pour ce faire un critère de similarité entre ces vecteurs est employé pour déterminer pour chaque forme de surface extraite de la phrase utilisateur les k plus proches exemples dans K. Selon la représentation choisie, un choix pertinent peut être celui de la similarité cosinus, c'est notamment le cas des représentations apprises avec la méthode *word2vec*. Une fois ces exemples sélectionnés, pour chaque forme de surface est extrait une liste ordonnée d'hypothèses sémantiques dont le score sera basé sur le produit scalaire effectué entre les distances du *word2vec* (estimée à partir des similarités) et les coefficients d'affectation avec chaque classe sémantique, correspondant aux k exemples retenus dans K. Ces hypothèses sont ensuite employées pour construire un transducteur à états finis dans lequel les formes de surface, leurs hypothèses sémantiques et leur score associé sont respectivement les entrées, les sorties et les poids des arcs considérés.

Un processus de repondération (pénalité appliquée pour chaque mot présent sur un arc) permet de régler l'influence de la longueur des formes de surface considérées. L'algorithme du plus court chemin est ensuite appliqué sur l'automate à états finis obtenu pour générer les hypothèses ordonnées de séquences d'étiquettes sémantiques (le plus court chemin est mis en couleur sur la figure 5.2).

5.2.2 Adaptation du modèle en ligne

Les performances du système décrit précédemment dépendront essentiellement de la bonne définition de K et aussi de la qualité de la représentation sémantique vectorielle employée F. Sachant qu'on ne peut garantir leur optimalité à l'état initial du système, nous proposons une méthode permettant d'adapter le modèle en fonction de retours utilisateurs sur les sorties du module de compréhension ainsi obtenu. L'idée générale est d'enrichir dynamiquement K avec de nouvelles formes de surface (amélioration de la couverture) mais aussi de pouvoir mettre à jour les valeurs d'affection des différentes entrées (liens en F et les étiquettes sémantiques) pour corriger d'éventuel faux positifs introduits (formes de surface ambigües).

Nous envisagerons également l'ajout de nouveaux concepts et valeurs (extension de domaine). Pour ce faire, nous proposons une stratégie d'adaptation en ligne, possiblement optimisée par l'intermédiaire d'un algorithme de bandit, afin d'être en mesure d'étendre le modèle avec de nouvelles connaissances en permanence. En effet, le transducteur obtenu par l'intermédiaire de l'analyseur syntaxique permet de retrouver une association directe entre les mots (ou séquence de mots) de l'utilisateur et l'étiquette sémantique. On peut ainsi exploiter cette information pour faciliter l'adaptation dynamique du modèle.

ment assez court (quelques mots en moyenne). Il est cependant possible de définir une taille de segment maximale à ne pas dépasser.

Selon la technique d'AL considérée, l'effort de supervision peut être de nature très différente selon si l'on considère que l'utilisateur doit procéder à une annotation d'énoncé complète ou seulement partielle, s'il doit faire une sélection parmi plusieurs hypothèses de sortie du système ou si simplement un retour positif/négatif sur la meilleure hypothèse est attendu.

Dans cette étude nous nous sommes premièrement intéressé à un scénario dans lequel la supervision est limitée à un ensemble de retours binaires sur les étiquettes sémantiques produites par le système (validation/rejet). Compte tenu du fait que cette technique ne nécessite pas une correction explicite des étiquettes de la part des utilisateurs, elle peut être facilement intégrée au sein d'une plate-forme de dialogue existante en utilisant des demandes de confirmation simples à l'utilisateur (question fermée à réponse oui/non). Dans l'étude actuelle, nous considérons un utilisateur (qui dans notre étude préliminaire sera simulé) donnant à chaque tour un retour sur chaque étiquette sémantique produite par le système. Ces retours sont ensuite utilisés pour mettre à jour K en K^*.

En tant que telle, la procédure décrite ci-dessus permet uniquement de corriger les erreurs de classification du modèle, mais n'autorise pas l'extension du domaine (ajout de nouveaux concepts et/ou valeurs dans les sorties du modèle). Ainsi, dans un second temps, nous avons étendu cette stratégie d'adaptation en ligne par l'introduction de nouvelles actions afin de répondre également à cette problématique. Pour définir cette nouvelle stratégie nous proposons de considérer le problème d'adaptation en ligne du module de compréhension comme celui d'un bandit contre un adversaire (*adversarial bandit* en anglais) avec pour objectif de déterminer une politique de demande de retours aux utilisateurs. Ce choix vise à minimiser les coûts de supervision relatifs à l'intervention humaine tout en posant aux utilisateurs les questions avec le plus d'impact possible sur la qualité future du modèle et par là même tenter d'optimiser un ratio coût/amélioration.

Les algorithmes de bandit ont été largement étudiés dans la communauté de l'apprentissage automatique (Auer et al., 2002; Bubeck et Cesa-Bianchi, 2012). Leur objectif vise à déterminer le meilleur compromis entre l'exploration des options qui ont donné le meilleur rendement (gains) dans les itérations précédentes et l'exploration de nouvelles options qui pourraient donner une meilleure performance à l'avenir. Peu de travaux ont déjà employé ce genre de techniques pour optimiser un module de NLP. Parmi ceux-ci, on pourra donner comme exemple (Ralaivola et al., 2011) où les auteurs appliquent un algorithmes de bandit contextuel pour raffiner un classificateur multi-classes avec des retours utilisateurs de type oui/non sur un tâche visant à identifier le motif d'un appel téléphonique (*call routing*).

Nous décrirons dans un premier le mécanisme d'adaptation du modèle sur le cas des retours binaires utilisateurs systématiques. Puis nous étendrons cette description au cas de l'optimisation d'une politique d'adaptation par un algorithme de bandit contre un adversaire.

Adaptation du modèle par retours binaires sur les hypothèses SLU

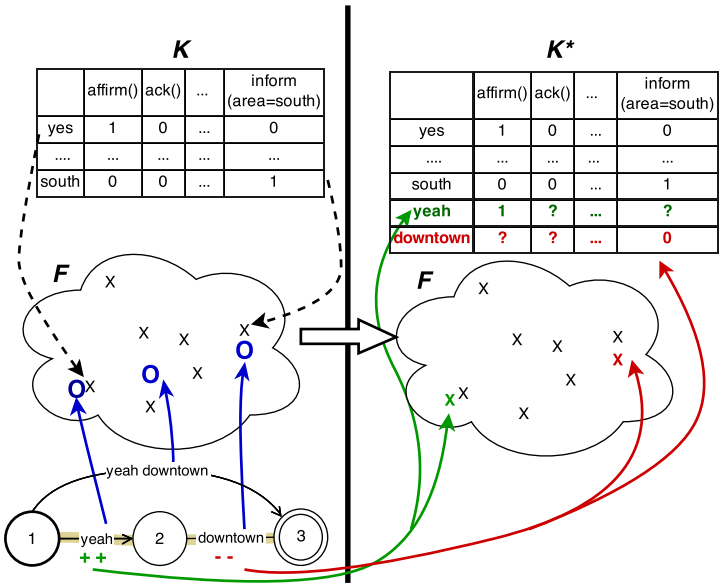


FIGURE 5.3 – *Illustration du mécanisme d'adaptation employé par la technique d'apprentissage sans données de référence.*

Un exemple du processus d'adaptation exploitant les retours binaires utilisateurs sur les hypothèses SLU est donné dans la figure 5.3. Ce dernier illustre un cas où les véritables étiquettes sémantiques de la phrase utilisateur sont mal reconnues par l'analyseur sémantique : ici la phrase « yeah downtown » est étiquetée comme `affirm() | inform(area=south)` au lieu de `affirm()|inform(area=centre)`.

Nous considérons dans un premier temps un approche systématique de collecte de retours binaires. Ainsi pour une phrase utilisateur dont la meilleure hypothèse sémantique contient m segments, les m retours utilisateurs constituent un jeu de m tuples $U := ((c_k, T_k, f_k))_{1 \leq k \leq m}$, où (c_k, T_k) est le couple forme-de-surface/étiquette-sémantique proposé à l'utilisateur et f_k est son retour (1 si positif, 0 si négatif). L'algorithme 1 (partiellement illustré sur la figure 5.2) est utilisé pour mettre à jour K en K^* en fonction de K et U.

Chaque cellule (i, j) dans K est constituée de 4 valeurs distinctes : $p_{i,j}$ et $n_{i,j}$ représentant respectivement le nombre de retours positifs et négatifs observés jusqu'alors, $knn_{i,j}$ la valeur obtenue par une addition par élément des k plus proches lignes voisines (repondérée par un produit scalaire des similarités normalisées, voir Algorithme 1.16) et $c_{i,j}$ la valeur d'affectation présentée ci-dessus qui est aussi la valeur utilisée par notre analyseur sémantique. Lors d'une mise à jour, l'ensemble des ces valeurs peut être impacté. Ainsi, l'algorithme 1 montre les conditions et la nature des mises à jour opérées

mais aussi comment K peut être étendue (ajout d'une nouvelle ligne en présence d'une séquence de mots inconnus, par exemple).

Dans un premier temps, K^* est initialisée avec une copie de K. Puis toutes les nouvelles formes de surface c de U qui ne figurent pas parmi les exemples connus dans K sont ajoutées dans K^* (cf. Algorithme 1.4-6). Ensuite tous les comptes sur les retours sont mis à jour en se basant sur les informations contenues dans U (cf. Algorithme 1.8-9). Pour ce faire deux facteurs d'échelles α_p et α_n on été introduit afin de permettre d'ajuster l'importance d'une nouvelle observation par rapport aux connaissances courantes au regard de sa valence (on pourrait par exemple choisir de faire plus confiance au retours positifs). Pour les couples forme-de-surface/étiquette-sémantique initiaux (issus de notre définition ontologique initiale du domaine), les valeurs $p_{i,j}$ sont initialisées avec une valeur a priori p_0.

Dans le cas général, la valeur d'affectation à une étiquette sémantique est obtenue par un simple ratio entre les retours positifs et négatifs associés à la cellule concernée (voir Algorithme 1.12).

Pour chaque modification de ligne, un marqueur m_i est employé afin de détecter si une connaissance a priori (affectation positive) a été remise en question par de nouvelles observations (détecté par une baisse de la valeur d'affection $c_{i,j}$, cf. Algorithme 1.13). Dans ce cas particulier, les affections pour lesquelles il n'y a eu aucune observation pour cette forme de surface (autres cellules sur la même ligne) et devrait prendre la valeur 0 prennent en fait celle de la fonction knn. Cette technique permet après adaptation de pouvoir tester de nouvelles propositions si la forme de surface venait à se représenter (processus d'exploration de l'espace d'affectation).

Extension et optimisation en ligne de la stratégie d'adaptation du modèle

Cette section vise à étendre la stratégie d'adaptation en ligne décrite précédemment pour permettre également la création de concepts/valeurs (extension de domaine) tout en tentant d'optimiser un ratio coût/amélioration du modèle. Dans cette étude préliminaire, nous adoptons une stratégie simple basée sur un algorithme de bandit contre un adversaire pour résoudre le problème d'optimisation de la stratégie d'adaptation du modèle. Avant d'aller plus loin dans la formulation, nous proposons en premier lieu d'expliciter la problématique sur un cas statique (i.e. sans adaptation).

Cas statique nous postulons que le système a le choix entre plusieurs actions vis à vis de l'utilisateur afin d'améliorer la détection automatique des étiquettes sémantiques associées aux énoncés de celui-ci. Nous devons d'abord définir l'espace de l'action considérée (qui correspondent donc aux bras dans la littérature relative aux algorithmes de bandits). Toutefois, nous pouvons déjà prévoir que chaque action implique une collaboration plus ou moins importante de la part de l'utilisateur. Par conséquent, nous introduisons une mesure de l'effort utilisateur relatif à l'action effectivement choisie par le système sous la forme d'une fonction de coût. Nous définissons également une mesure de l'inefficacité du modèle que nous tenterons de réduire dans le temps. Cette

Algorithm 1 Mise à jour de la base de connaissance K

Sachant : K et U Sortie : K^*
$K^* \leftarrow K$
for all $(c, T, f) \in U$ **do**
    if $c \notin K^*$ **then**
        ajouter une nouvelle ligne pour c dans K^* avec les valeurs des cellules initialisées par défaut
        $m_{last} = 1$
    $i \leftarrow$ identifiant ligne c, $j \leftarrow$ identifiant colonne T
    $p_{i,j} \leftarrow p_{i,j} + f \times \alpha_p$
    $n_{i,j} \leftarrow n_{i,j} + (1 - f) \times \alpha_n$
    if $p_{i,j} + n_{i,j} > 0$ **then**
        $old_c \leftarrow c_{i,j}$
        $c_{i,j} \leftarrow \dfrac{p_{i,j}}{p_{i,j} + n_{i,j}}$
        if $c_{i,j} - old_c < 0$ **then** $m_i \leftarrow 1$
    else $c_{i,j} \leftarrow 0$
for all $c_{i,j} \in K^*$ **do**
    calculer $knn_{i,j}$
for all $c_{i,j} \in K^*$ **do**
    if $p_{i,j} + n_{i,j} = 0$ et $m_i = 1$ **then** $c_{i,j} \leftarrow knn_{i,j}$

mesure nous permet de quantifier l'amélioration de modèle résultant d'une action spécifique. Enfin le problème de l'adaptation du modèle est formulé comme un problème d'optimisation linéaire où le système a la pleine connaissance de la fonction objectif .

Espace d'actions et fonction de coût traduisant l'effort utilisateur lorsque l'utilisateur fournit une phrase, le système peut choisir une action (à partir d'une distribution de probabilité) parmi un ensemble $\mathcal{I}$ de M actions. Dans cette configuration préliminaire, nous considérons le cas où $M = 3$ et où $\mathcal{I}$ peut être défini comme :

$$\mathcal{I} := \{\text{Skip, YesNoQuestions, AskAnnotation}\}. \tag{5.1}$$

Soit $i \in \mathcal{I}$ l'indice de l'action. Nous supposons que l'effort de l'utilisateur (coût de l'action), $\phi(i) \in \mathbb{N}$, peut être mesuré par le nombre d' échanges réalisés entre le système et l'utilisateur pour mener à bien l'action i.

Une description des différentes actions et de leurs coûts associés est donnée ci-dessous :

— Skip : n'appliquer aucune mise à jour au modèle. Le coût de cette action est toujours considéré comme étant nul ($\phi(\text{Skip}) = 0$) puisque l'utilisateur n'est pas sollicité.
— YesNoQuestions : mettre à jour K avec les réponses oui/non données par l'utilisateur aux questions de confirmation sur les étiquettes sémantiques détectées

dans la meilleure hypothèse sémantique. Cette action correspond exactement à celle employée de façon exclusive par la stratégie d'adaptation décrite dans la section précédente. Si cette action est prise, le système tentera en premier lieu une confirmation générale sur la meilleure hypothèse SLU pour un coût de 1. En cas de négation, un confirmation (+1 sur le coût) sera demandée pour chaque étiquette sémantique détectée [2].

— AskAnnotation : demander à l'utilisateur d'annoter son tour de parole complètement pour mettre à jour K avec de nouveaux exemples positifs. Si cette action est prise, le système tentera en premier lieu une confirmation générale sur la meilleure hypothèse SLU pour un coût de 1. Dans le cas d'un rejet, l'utilisateur devra procéder à l'annotation étiquette par étiquette de l'énoncé. Pour ce faire, nous supposons que pour chacune d'entre elle l'utilisateur informera le système de la localisation dans sa phrase de l'acte de dialogue qu'il s'apprête à annoter (+1 sur le coût) puis identifiera consécutivement l'*acttype*, le *concept* et la *valeur* si besoin est [3] (+1 sur le coût par question intermédiaire). Par cette action de nouveaux concepts et valeurs pourront donc être ajoutés par l'utilisateur au module SLU (extension du domaine).

Mesure d'inefficacité afin d'estimer la performance du modèle sur l'énoncé de l'utilisateur courant sans le recours à la transcription sémantique de référence, nous choisissons d'introduire une mesure extraite directement des sorties du modèle d'apprentissage sans utiliser de données de référence. Du fait que nous cherchons à optimiser un problème de minimisation globale dont une des composantes est la fonction de coût précédemment introduite, l'inefficacité est mesurée en lieu et en place de l'efficacité. Ainsi, soit $d \in [0,1]$ le poids moyen des arcs [4] constituant le meilleur chemin de la machine à états finis utilisée par l'analyseur sémantique. Comme expliqué dans la section 5.2.1, les poids des arcs correspondent aux produits scalaires des distances entre chaque séquence contiguë de mots pris en compte dans le meilleur chemin et les coefficients d'affectation des k exemples que leur sont les plus similaires dans K. Ainsi, plus le poids moyen est élevé, moins le modèle semble correspondre à l'énoncé. En effet, un poids moyen élevé traduit le fait qu'il n'y a pas suffisamment de similitudes entre l'énoncé courant et les exemples présent dans K.

Selon l'action de raffinement choisie i, d est mis à jour en $d'(i) \in [0,1]$ en raison de la modification du modèle qui en résulte. Nous décrivons ci-dessous les principaux mécanismes mis en jeu :

— Skip : sachant que cette action n'implique pas de changements modèle, la mesure de l'inefficacité reste constante. Ainsi, $d'(Skip) = d$.

— YesNoQuestions : en utilisant cette action, chacun des m DAs dans la meilleure hypothèse sémantique sera confirmé ou nié par l'utilisateur. Selon ce qui a été dit précédemment, ces évaluations utilisateurs sont converties en un ensemble

2. Uniquement dans le cas où il y en a plus d'une, car dans le cas contraire la réponse à la première question est suffisante.

3. Certains *acttype*s sont vides, comme $hello()$, ou ne contienne qu'un concept sans valeur, comme$request(food)$.

4. En supprimant la pénalité appliquée à chaque mot dans le décodage sémantique.

U de m tuples $U := ((c_l, T_l, f_l))_{1 \leq l \leq m}$, où (c_l, T_l) est un couple forme-de-surface / étiquette-sémantique proposé à l'utilisateur et f_l est son retour (1 positif, 0 négatif). Compte tenu de K et U après chaque interaction, l'algorithme 1 est utilisé pour mettre à jour K en K^* (mise à jour des comptes sur les observations positives et négatives des séquences de mots concernées et utilisation du voisinage dans l'espace sémantique F pour remplir les valeurs d'assignation inconnues). Ainsi, $d'(YesNoQuestions) = \delta$ où δ est le nouveau poids moyen de l'énoncé récemment actualisée dans K^* .

— AskAnnotation : si l'hypothèse sémantique de l'énoncé est validée dans son intégrité par l'utilisateur nous considérons que tous les couples forme-de-surface / étiquette-sémantique extraits de la meilleure hypothèse sémantique (plus court chemin) ont été évalués de façon positive par l'utilisateur. Sinon, les m' couples forme-de-surface/étiquette-sémantique annotés par l'utilisateur[5] sont considérées comme un ensemble de tuples $U := ((c_l, T_l, 1)_{1 \leq l \leq m'})$. Dans ce cas précis, de nouveaux concepts et valeurs peuvent être ajoutés dans les sorties possibles du modèle (ajout d'une colonne dans K^*). En raison du fait que des parties de l'énoncé sont désormais dans K^* en tant qu'exemples positifs, $d'(AskAnnotation) \approx 0$.

Fonction de perte nous devons finalement définir une fonction de perte (*loss function* en anglais) telle que le système, par le fait de chercher à l'optimiser, réduira dans le même temps la mesure de l'inefficacité actualisée $d'(i)$ et la mesure de l'effort de l'utilisateur $\phi(i)$. Ainsi, nous proposons de définir la fonction de perte $l(i) \in [0, 1]$ comme étant la combinaison convexe des deux mesures précédemment introduites :

$$l(i) := \underbrace{\gamma d'(i)}_{\substack{\textbf{amélioration} \\ \textbf{du modèle}}} + \underbrace{(1-\gamma)\frac{\phi(i)}{\phi_{max}}}_{\textbf{effort utilisateur}} \tag{5.2}$$

où $\gamma \in [0, 1]$ permet de régler l'importance de l'amélioration du modèle sur l'effort utilisateur dans le processus d'optimisation. $\phi_{max} \in \mathbb{N}_+$ correspond au nombre maximal d' échanges possibles entre le système et l'utilisateur (dans un même tour).

Soit $\mathbf{p} \in \Delta(3) := \{\mathbf{q} \in \mathbb{R}^3_+ \mid \sum_{i \in \mathcal{I}} q(i) = 1\}$ la distribution de probabilité sur les différentes actions. L'objectif d'adaptation du modèle est donc défini comme :

$$\min_{\mathbf{p} \in \Delta(3)} E[l] = \sum_i p(i) l(i). \tag{5.3}$$

Si nous avions une pleine connaissance de $l(i)$ pour chaque action i, le problème d'adaptation du modèle serait équivalent à celui consistant à résoudre $\min_i \{l(i)\}$. Cependant, dans le scénario considéré, ce cadre ne peut pas être appliqué car la fonction de perte $l(i)$ n'est pas connue explicitement (pas observable pour toutes les actions à

5. Il est à noter que m' peut être différent de m car l'utilisateur est alors en mesure de spécifier les frontières des étiquettes sémantiques lors du processus d'annotation de la phrase.

tous les instants). Par exemple, lorsque le système utilise $i = YesNoQuestions$, les valeurs $d'(YesNoQuestions)$ et $\phi(YesNoQuestions)$ ne pourront être déterminées qu'après l'exécution de l'action car elles dépendent à la fois des réponses de l'utilisateur et de l'état courant de K. De plus, le système reçoit en entrée des énoncés très distincts et provenant pas toujours du même utilisateur. De ce fait, il nous a paru intéressant de voir le problème d'adaptation comme celui d'un bandit contre un adversaire.

Cas du bandit contre un adversaire nous considérons donc le scénario AL suivant :

Le problème d'adaptation du modèle par une méthode de bandit contre un adversaire
Paramètres connus : L'espace d'actions $\mathcal{I}$ et le coefficient $\gamma \in [0,1]$.
À chaque tour $t = 1, 2, \ldots$

1. Le système reçoit un énoncé utilisateur, en extrait la meilleure hypothèse sémantique et obtient d_t ;
2. Le système choisit une action $i_t \in \mathcal{I}$, éventuellement en ayant recours à de l'aléatoire (exploration) ;
3. Une fois l'action i_t exécutée, le système calcule :
   - la nouvelle mesure d'inefficacité $d'_t(i_t)$;
   - l'effort utilisateur à t, $\phi_t(i_t)$, qui correspond au nombre d'échanges effectivement réalisés entre le système et l'utilisateur lors de la réalisation de i_t ;
   - la fonction de perte :

$$l_t(i_t) = \gamma d'_t(i_t) + (1-\gamma)\phi_t(i_t).$$

But : Trouver $i_1, i_2, \ldots$, tel que pour chaque T, le système minimise la perte cumulée :

$$\sum_{t=1}^{T} l_t(i_t) = \gamma \sum_{t=1}^{T} d'_t(i_t) + (1-\gamma) \sum_{t=1}^{T} \phi_t(i_t).$$

Aucune hypothèse n'est formulée à propos de $d'_t(i_t) \in [0,1]$ et $\phi_t(i_t) \in [0,1]$. Ainsi, nous ne présupposons pas de l'effet qu'a une action i_{t-l}, avec $l \in \{1, \ldots, t-1\}$, sur la fonction de perte pour le tour t. Ce choix est justifié par le fait qu'une phrase utilisateur ne peut pas être prédite avec précision par le système sans connaissances a priori robuste. Ce scénario est donc étroitement lié à celui d'un démarrage à froid du système. Ce sont ces considérations qui nous ont amené à retenir le cadre du bandit contre un adversaire.

En effet, dans ce formalisme, les récompenses ne sont pas supposées être indépendantes et identiquement distribuées (comme dans le cas stochastique) mais choisies arbitrairement par un adversaire. Parmi les algorithmes envisageables, nous avons retenu l'algorithme Exp3 (*Exploration-Exploitation using Exponential weights*) (Auer et al., 2002) (voir algorithme 2). Il s'agit là d'un algorithme efficace lorsqu'un petit nombre de bras est en jeu. Une preuve mathématique des performances relativement élevées de

cet algorithme est notamment donnée dans (Bubeck et Cesa-Bianchi, 2012).

Algorithm 2 Exp3

Sachant : $\gamma' \in [0,1]$
Initialiser les poids $w_i(1) = 1$ pour $i = 1, \ldots, M$.
for chaque tour t **do** :
    - calculer $p_i(t) = (1-\gamma')\frac{w_i(t)}{\sum_{j=1}^{M} w_j(t)} + \frac{\gamma'}{M}$ pour chaque i.
    - déterminer la prochaine action i_t aléatoirement selon la distribution $p_i(t)$.
    - observer la récompense $x_{i_t}(t)$.
    - calculer la récompense estimée $\hat{x}_{i_t}(t) = x_{i_t}(t)/p_{i_t}(t)$.
    - mettre à jour les poids :
    $w_{i_t}(t+1) = w_{i_t}(t)e^{\gamma' \hat{x}_{i_t}(t)/M}$ et $w_j(t+1) = w_j(t)$ pour tout autre action j.

5.2.3 Intégration dans un mécanisme d'apprentissage supervisé

Une approche complémentaire à celle proposée dans ce chapitre consiste à avoir recours à l'utilisation d'un cadre d'apprentissage supervisé plus standard pour exploiter des données annotées fournies progressivement au système (approche incrémentale). Pour ce faire, nous considérons ici le modèle CRF (Lafferty et al., 2001), et tout particulièrement sa formalisation log-linéaire, en raison de sa bonne performance sur la problématique de la compréhension de la parole (Raymond et Riccardi, 2007). Bien que des alternatives plus récentes existent dans la littérature, comme par exemple l'utilisation de modèles DBN (Deoras et Sarikaya, 2013), RNN (Yao et al., 2013; Mesnil et al., 2013) ou R-CRF (Yao et al., 2013, 2014), les résultats expérimentaux obtenus montrent que leurs performances sont tout de même proches de celles du CRF en pratique voire parfois moins bonnes des tâches plus complexes (Vukotic et al., 2015). De ce fait, ce dernier peut être encore considéré comme un modèle état de l'art pour la tâche de SLU.

Un modèle CRF est un modèle discriminant qui représente la distribution des probabilités conditionnelles d'une séquence de N concepts (c_1^N) sachant une séquence de N mots (w_1^N) comme suit :

$$P(c_1^N|w_1^N) = \frac{1}{Z}\prod_{n=1}^{N} exp(H(c_{n-1}, c_n, \phi(w_1^N, n))) \tag{5.4}$$

avec

$$H(c_{n-1}, c_n, \phi(w_1^N, n)) = \sum_{m=1}^{M} \lambda_m h_m(c_{n-1}, c_n, \phi(w_1^N, n)) \tag{5.5}$$

Les h_m sont des fonctions caractéristiques extraites du corpus d'apprentissage (généralement il s'agit de fonctions booléennes indiquant la présence ou l'absence d'une observation) et les λ_i correspondent aux poids du modèle qui sont estimés lors de l'apprentissage. $\phi(w_1^N, n)$ représente une fonction de motif qui sera utilisée pour déterminer les descripteurs employées pour chaque mot de la séquence (fenêtre d'observation)

dans l'apprentissage. Un choix usuel consiste à choisir w_{n-2}^{n+2} qui représente une fenêtre d'observation de 2 mots autour du mot courant. Z est un terme de normalisation au niveau de la phrase complète qui est défini par :

$$Z = \sum_{\widetilde{c}_1^N} \prod_{n=1}^{N} exp(H(c_{n-1}, c_n, \phi(w_1^N, n))) \tag{5.6}$$

où $\widetilde{c}_1^N$ correspond à l'ensemble des séquences de concepts possibles pour cette phrase.

Dans (Lavergne et al., 2011), un modèle basé sur l'utilisation de transducteurs à états finis a été proposé afin d'obtenir un système de traduction efficace à base de CRF. Ces transducteurs permettent notamment d'offrir un formalisme dans lequel différents modèles de probabilités peuvent être représentés et composés. Cette proposition originale a été revisitée dans (Jabaian et al., 2014) dans le contexte du SLU pour permettre un étiquetage sémantique d'une phrase utilisation par le biais d'une composition de transducteurs de la forme suivante :

$$\lambda_{understanding} = \lambda_S \circ \lambda_T \circ \lambda_F \tag{5.7}$$

où

— λ_S est l'accepteur de la phrase source s ;
— λ_T est un dictionnaire de tuples, combinant des séquences des mots avec leurs possibles étiquettes sémantiques sur la base d'un inventaire de tuples connus ;
— λ_F est une fonction d'extraction de motifs (*feature matcher* en anglais) qui attribue un score aux tuples dans le contexte courant, généralement une probabilité, en utilisant un modèle préalablement appris.

Une architecture similaire est adoptée dans notre proposition. La base de connaissance K est ici utilisée pour initialiser λ_T. En effet, les tuples valides sont extraits par le moyen d'une approche k-NN qui associe à chaque segment de la phrase d'entrée, une liste de tuples candidats (ex_i, y_i) issus de K. Cette liste est obtenue en exploitant une métrique sur F (par exemple la distance cosinus). Ces tuples sont considérés comme les arcs dans le transducteur λ_T. Deux configurations de l'analyseur peuvent être définis en fonction de la nature des candidats extraits de K. En effet, ces dernières peuvent être soit des mots soit des segments de mots (resp . notée **word-parser** et **chunk-parser**) .

Un modèle statistique (CRF dans notre étude) peut donc être entraîné soit sur les mots soit sur les segments pour constituer λ_F. La meilleure hypothèse sémantique au niveau de l' expression est obtenue par décodage de type « meilleur chemin » sur la machine à états finis composite (ici mis en évidence en chemin dans la figure Fig. 5.2).

5.3 Expériences et résultats

5.3.1 Description des données DSTC2 et DSTC3

Notre étude expérimentale a été menée sur une tâche de compréhension de la parole en utilisant les données de la seconde et de la troisième campagne d'évaluation

Dialog State Tracking Challenge [6] (DSTC2 and DSTC3) (Henderson et al., 2014a,b). Ces corpus ont été construits pour un défi de recherche dédié à la détection du but de l'utilisateur tout au long d'un dialogue oral (et non pas uniquement l'étiquetage sémantique des énoncés de l'utilisateur au fur et à mesure). Cependant, dans notre étude expérimentale, nous exploitons ces données (transcriptions, annotations sémantiques, etc.) uniquement pour évaluer notre approche d'apprentissage sans données de référence pour l'étiquetage sémantique sur deux configurations de dialogues réalistes. Ce qui nous permettra de montrer que l'approche proposée et ses variantes offrent des performances comparables à celles obtenues par des systèmes à base de règles expertes ou appris sur des données annotées.

Le défi DSTC2 couvre le domaine de la recherche d'informations sur des restaurants alors que DSTC3 étend le domaine et couvre également la recherche d'informations touristiques plus générale en incluant notamment des nouveaux types d'établissement (pubs, coffee shops) mais aussi de nouveaux concepts et valeurs. Dans notre expérience, seules les données de test de ces deux corpus sont utilisées (9890 énoncés d'utilisateurs pour DSTC2 et 18715 pour DSTC3). Chaque ensemble est évalué en deux modes différents : transcriptions manuelles et N-meilleures transcriptions automatiques des entrées de l'utilisateur.

Si classiquement chaque étiquette sémantique c_i est définie par un couple concept-valeur, comme par exemple *food=Italian* ou encore *destination=Boston*, le standard d'annotation sémantique employé dans les corpus DSTC2 et DSTC3 diffère légèrement (voir annexe A.2). Dans ce contexte, les étiquettes sémantiques correspondent à des actes de dialogue de la forme `acttype(concept=valeur)` où acttype représente le nature de l'acte de dialogue considéré, à savoir son intention dialogique (la confirmation ou la réfutation). Par exemple, la phrase utilisateur « hello i am looking for a french restaurant in the south part of town » sera associée à la séquence d'actes de dialogue suivante « `hello() | inform(food=french) | inform(area=south)` ».

Les combinaisons possibles de `acttype(concept=valeur)` sont déterminées sur la base d'une ontologie du domaine recensant les différents types d'actes de dialogue, de concepts ainsi que leurs valeurs respectives.

Les différents types d'actes de dialogue sont en grande partie indépendants de la tâche visée. Ils peuvent se diviser en quatre grands groupes : ceux ayant pour but de transmettre de l'information (`inform`), ceux représentant différents types de requête (`request, reqalts, reqmore`), ceux relatifs aux confirmations (`confirm, affirm, negate, deny`) et les formules de politesse (`hello, thankyou, bye`). L'ensemble des couples concepts/valeurs est quant à lui très lié à la tâche de dialogue, chaque couple correspond généralement à une entrée spécifique dans la base de données utilisée pour répondre aux requêtes des utilisateurs (contraintes de recherche).

6. http ://camdial.org/ mh521/dstc/

5.3.2 Métriques pour l'évaluation

Tout au long de cet étude nous emploierons la **F-mesure** en tant qu'indicateur principal de la qualité de la sortie d'un décodeur sémantique sur la base de corpus de référence annotés. Cette mesure représente la moyenne harmonique de la **Précision** et du **Rappel** de la meilleure hypothèse sémantique.

$$\text{F-mesure} = \frac{2 \cdot (\text{Précision} \cdot \text{Rappel})}{\text{Précision} + \text{Rappel}} \tag{5.8}$$

avec

$$\text{Précision} = \frac{\text{nombre de concepts corrects trouvés}}{\text{nombre de concepts trouvés}} \tag{5.9}$$

et

$$\text{Rappel} = \frac{\text{nombre de concepts corrects trouvés}}{\text{nombre de concepts à trouver}} \tag{5.10}$$

Il est à noter que ces indicateurs, contrairement au CER (dont la définition est donnée dans l'annexe B), ne tiennent pas compte de l'aspect séquentiel du processus de décodage sémantique. Ainsi, les références employées n'ont pas besoin d'être alignées aux mots pour le calcul de cette métrique.

5.3.3 Évaluation de l'approche standard

Démarrage de zéro du module de compréhension

Afin de constituer notre espace sémantique, un modèle *word2vec* (Mikolov et al., 2013a) a été utilisé pour apprendre une représentation vectorielle des mots sur 300 dimensions. Ce modèle a été appris avec l'algorithme *Skip-gram* (avec une fenêtre de 10 mots) avec un softmax hiérarchique grâce à l'outil Gensim [7] (Řehuuřek et Sojka, 2010) sur une grande quantité de données disponibles librement et présentant une grande couverture thématique. Plus exactement, ont été employé les corpus anglais *enwik9*, *One Billion Word Language Modelling Benchmark*, *Brown corpus*, *English GigaWord* de 1 à 5. Ce qui représente au total plus de 4 milliards de mots en contexte (phrases).

Ce type de représentation présente certaines régularités avec les propriétés syntaxiques et sémantiques des mots comme celles montrées dans (Mikolov et al., 2013c) ainsi qu'une structure linéaire permettant la combinaison des représentations des mots par une simple addition vectorielle élément par élément. Cette technique est donc utilisée pour projeter nos formes de surface vers leur représentation sémantique vectorielle de type *word2vec* vue comme une somme des représentations individuelles de chaque mot les constituant.

Plusieurs travaux état-de-l'art ont montré que la similarité cosinus (équation 5.11) est une métrique pertinente pour comparer les vecteurs de mots *word2vec* entre eux

7. https://radimrehurek.com/gensim/

(Mikolov et al., 2013a,c). Cette métrique considère le cosinus de l'angle θ entre deux vecteurs A et B de n dimension comme une mesure de similarité. Une valeur de -1 indiquera des vecteurs opposés, 0 que l'on est en présence de vecteurs indépendants et 1 que les vecteurs sont similaires. Cette mesure, très commune en NLP, est obtenue en appliquant la formule :

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{5.11}$$

Nous avons également utilisé cette métrique dans l'algorithme de type k plus proche voisins pour la prédiction sur les formes de surface et l'adaptation de la base de connaissance. Ainsi, dans les expériences considérées, $k = 1$ pour l'analyse sémantique et 20 pour les valeurs *knn* dans la matrice d'affectation.

Le graphe sémantique (transducteur) est obtenu à l'aide de l'outil OpenFst [8] et l'algorithme du plus court chemin est employé pour déterminer la ou les meilleures hypothèse [9] (voir la section 5.2.1.).

Les bases de connaissances liées aux deux tâches utilisées pour nos expériences sont extraites des descriptions ontologiques fournies dans le cadre de ces défis scientifiques (e.g. listes des concepts/valeurs) ainsi que d'un ensemble d'information générique en suivant la procédure automatique décrite dans la section 5.2.1. La sémantique du domaine DSTC2 est constituée de 8 concepts et 215 valeurs et celle du DSTC3 de 13 concepts et 279 valeurs (voir détails dans l'annexe C.2). Pour les deux tâches 16 *acttype* sont considérés, il en résultent donc 663 étiquettes sémantiques possibles pour DSTC2 et 855 pour DSTC3.

Nous avons définis manuellement 53 formes de surface associées aux différents *acttype*s. Par exemple « say again » est utilisé pour représenter l'acte *repeat()*. Cet effort est commun aux deux tâches cibles. Dans les deux descriptions ontologiques considérées, les concepts et les valeurs ont des noms significatifs (lexicalisés) qui peuvent directement être utilisés dans les formes de surface comme « address », « french », « has tv »). Au total, 4160 formes de surface ont été ainsi générées automatiquement et sont utilisées pour DSTC2, 6555 pour DSTC3.

Pour évaluer nos propositions, les résultats sont comparés avec deux systèmes état de l'art : le premier est un système à base de règles expertes utilisé dans le défi DSTC et le second est un système présenté dans (Williams, 2014), appris sur les données d'apprentissage du DSTC2 (nommé SLU1 dans l'article de Williams). Ces deux systèmes sont respectivement référencés par « S-règles » et « S-appris » dans la suite.

Les résultats de nos expériences (présentés dans le tableau 5.1) montrent que l'approche proposée, nommé **ZSSP** (pour *Zero-Shot Semantic Parser*) par la suite, atteint un niveau de performance (en termes de F-mesure) légèrement meilleur que celui de l'approche à base de règles (0,794 contre 0,782 sur DSTC2 et 0,826 contre 0,824 sur DSTC3)

8. www.openfst.org
9. La distance cosinus interviendra dans le calcul des poids des arcs. Cette dernière est définie comme $d_{\cos\theta}(A,B) = 1 - \frac{(1+\cos\theta)}{2}$ pour être à valeur dans $[0,1]$

Tâche	Modèle	Entrée	F-mesure	P	R
DSTC2	**S-règles**	n-meilleures	0,782	0,900	0,691
	S-appris	n-meilleures	0,802	0,846	0,762
	ZSSP	manuelle	0,919	0,898	0,942
		n-meilleures	0,794	0,796	0,792
DSTC3	**S-règles**	n-meilleures	0,824	0,852	0,797
	ZSSP	manuelle	0,899	0,873	0,928
		n-meilleures	0,826	0,806	0,849

TABLE 5.1 – *Evaluation des performances de l'analyseur sémantique, ZSSP, basé sur l'apprentissage sans données de référence en termes de F-mesure, **P**récision et **R**appel.*

et comparable à celui d'un modèle appris (0,794 contre 0,802 sur DSTC2). Ainsi le modèle proposé atteint dans son état initial des performances état-de-l'art sans utilisation de nombreuses règles spécifiques manuellement établies (coût d'experts humains) ni de données d'apprentissage (coût de collecte et d'annotation).

Cependant, afin de mesurer l'impact de la représentation sémantique choisie sur la performance globale de l'approche, un système qui n'utilise pas ce type de représentation a été construit. Un F-mesure de 0,839 (contre 0,919 en configuration normale) est obtenu sur les transcriptions manuelles du DSTC2 par une simple stratégie de détection de patrons de mots à partir des exemples de la même base de connaissances K. Cette dernière observation confirme l'avantage d'avoir recours à une représentation sémantique riche apprise sur une grande quantité de données non annotées. En effet, cette dernière permet une meilleure généralisation des connaissances lexicales initiales (qui elles peuvent être assez limitées).

Généralisation

L'avantage majeur de l'utilisation d'un modèle *word2vec* par rapport à un simple modèle de détection par mots clés est l'intégration d'une représentation continue des mots dans le processus de décodage. Cette caractéristique confère au système une capacité de généralisation intrinsèque permettant de couvrir des mots inconnus correspondant à des valeurs non présentes dans l'ontologie définie du domaine ou de la tâche. Par exemple, dans le contexte d'un domaine de recherche de restaurant, il est intéressant pour un système de dialogue de détecter certaines situations où un utilisateur parle d'un type d'aliment inconnu jusqu'alors par le système (si ce dernier n'est pas dans la base de données d'origine) ou au moins être en mesure de proposer une alternative en conséquence (en exploitant par exemple la proximité dans l'espace sémantique).

Afin d'évaluer la capacité de généralisation de notre système, nous avons volontairement supprimé de la base de connaissances de DSTC2 des formes de surface correspondant à différents pourcentages des valeurs possibles de certains concepts spécifiques. Dans cette étude préliminaire, nous avons choisi d'étudier l'impact sur les concepts *food*, *area* et *pricerange*. Les performances du modèle sur les transcriptions manuelles ont été évaluées en termes de F-mesure sur le DA sans valeur (*acttype(concept)*

uniquement au lieu de *acttype(concept=valeur)*) afin d'évaluer le niveau de détection des concepts de haut niveau uniquement.

Ainsi, nous comparons la performance de **ZSSP** avec une autre configuration de l'analyseur, notée **ZSSP-wordspot**. Cette dernière étiquette uniquement les segments qui atteignent un degré de similarité très élevé (une correspondance quasi parfaite de 0, 94). Vu que ce modèle est capable d'exploiter l'espace sémantique, cette configuration peut être assimilée à une stratégie robuste de détection de mots clés.

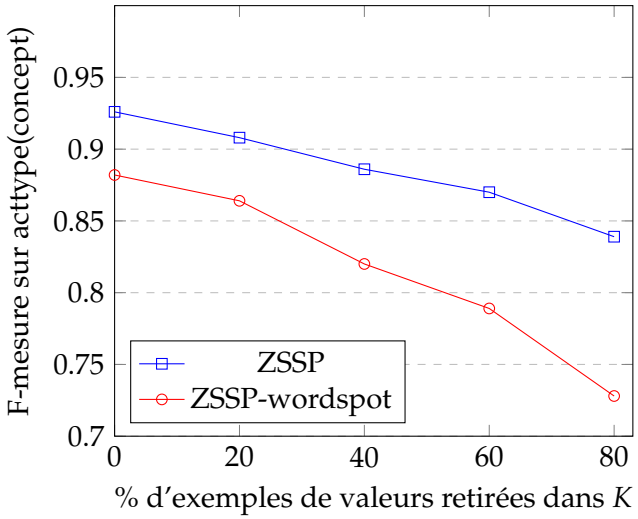


FIGURE 5.4 – *Capacité de généralisation de ZSSP sur le corpus de test DSTC2 en terme de F-mesure sur la détection d 'actes de dialogue génériques (i.e. acttype(concept)), fonction du pourcentage d'exemples de valeurs retirées dans K.*

Les résultats présentés dans la figure 5.4 montrent clairement une légère baisse de performance lorsque le pourcentage de valeurs retirées est grand. La différence entre les deux configurations est de 0, 044 à 0% et de 0, 111 à 80%. Cela confirme que l'approche proposée est tolérante à une faible densité de données dans K. Cette caractéristique peut être utile pour développer un système de dialogue générique permettant une évolution transparente de la base de connaissances contenant une base de données croissante.

5.3.4 Capacité d'adaptation en ligne

Cette section vise a présenter les résultats obtenus avec les deux stratégies d'adaptation en ligne du ZSSP proposées dans la section 5.2.2. Pour ce faire, les transcriptions des énoncés utilisateur du corpus d'apprentissage du DSTC2 sont employées pour simuler des retours utilisateur sur les sorties du système pour initier le processus d'adaptation de la base de connaissance K décrit dans l'algorithme 1. Nous exploitons ici les transcriptions manuelles pour s'abstraire de l'impact du bruit dû aux erreurs de transcription automatique qui compliquerait grandement la procédure de simulation employée (annotations sémantiques de référence pour les sorties ASR indisponibles dans les corpus considérés).

Dans un premier temps nous étudierons l'impact de retours binaires sur les hypothèses du module de compréhension avec diverses configurations du modèle sans données de référence. Puis dans un second temps nous étudierons l'intérêt de considérer l'optimisation en ligne de la stratégie d'adaptation employée.

Adaptation du modèle par retours binaires sur les hypothèses SLU

Pour rendre possible la phase d'adaptation, les retours des utilisateurs sont simulés en comparant la meilleure hypothèse du modèle avec l'étiquette sémantique de référence des phrases utilisateurs dans le corpus d'apprentissage DSTC2. Toutes les formes de surface de notre meilleure hypothèse ayant une étiquette sémantique présente dans l'annotation de référence sont considérées comme positives et toutes les autres comme négatives. K est mise à jour à la fin de chaque tour en suivant l'algorithme 1 présenté dans la section 5.2.2 (avec $\alpha_p = \alpha_n = 1$).

Dans le but de quantifier l'influence de l'espace sémantique considéré F et de la base de connaissance initiale K sur l'approche sans données de références proposée, nous avons fait le choix d'étudier trois configurations différentes de cette dernière. Nous distinguerons donc de l'approche **ZSSP** classique (base de connaissance K de qualité et un espace sémantique reposant sur une représentation *word2vec* apprise sur une grande quantité de données) deux variantes : la première, notée **ZSSP.F̃**, utilise une représentation sémantique « dégradée » et réduite à 50 dimensions, à savoir une représentation *word2vec* apprise avec l'algorithme *Skip-gram* (avec une fenêtre de 5 mots) sur des données non annotées issues du corpus d'apprentissage du DSTC2 (190366 mots en contexte) ; la seconde, notée **ZSSP.K̃** utilise une version « dégradée » de K où 10% des formes de surface (exemples de types d'actes de dialogues) ont été retirées. Il est à noter qu'en l'état aucune étiquette sémantique n'a été enlevée du modèle.

Dans le but de positionner notre approche par rapport à l'état de l'art, les mêmes systèmes de référence que précédemment sont utilisés.

Les résultats présentés dans la figure 5.5 montrent l'évolution de la F-mesure en fonction du nombre de dialogues utilisés pour l'adaptation. Même avant l'adaptation **ZSSP** (0,794) et **ZSSP.K̃** (0,775) atteignent des performances proches d'un système à base de règle (0,782). Mais un espace sémantique appris sur une petite quantité de données peut avoir un impact significatif sur cette performance (comme le montre **ZSSP.F̃**, 0,684) dû à la fois à des mots hors vocabulaire et des mauvaises propriétés de généralisation de cet espace sémantique.

Néanmoins, dans toutes les configurations de **ZSSP**, la performance augmente conjointement avec le nombre de dialogues d'adaptation. En effet, à la fois **ZSSP** et **ZSSP.K̃** obtiennent, après seulement 100 dialogues, des performances nettement meilleures que les modèles de références (0.811 contre 0,782 pour **S-règles** et 0,803 pour **S-appris** [10]).

10. Les performances de S-appris n'ont pas été reportées sur la figure. 5.5 dans le but d'éviter une possible confusion (au regard de l'axe des abscisses) sachant qu'il utilise beaucoup plus de données d'apprentissage.

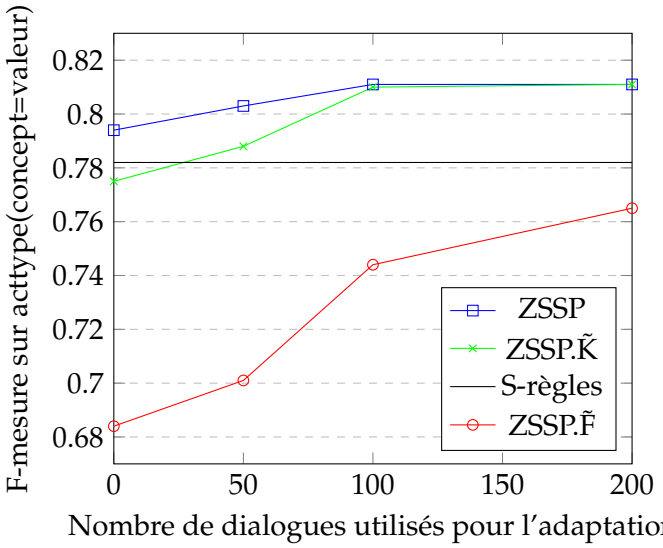


FIGURE 5.5 – *Performances de 3 configurations de la méthode ZSSP en terme de F-mesure, fonction du nombre de dialogues utilisés pour l'adaptation.*

En outre, même l'écart entre **ZSSP.F̃** et le modèle à base de règles est nettement réduit tout au long du processus d'adaptation en ligne (de 0,098 à 0,017 après 200 dialogues). Cette observation montre que la méthode proposée peut aussi fonctionner avec un espace sémantique de mauvaise qualité. Ces résultats confirment donc l'avantage de la méthode d'adaptation en ligne proposée pour faire face aux limites de la couverture initiale de K et à la robustesse de l'espace sémantique F.

Optimisation en ligne de la stratégie d'adaptation du modèle

Afin de tester l'algorithme d'apprentissage de la politique d'adaptation du modèle, nous avons choisi dans ces travaux de simuler les réponses de l'utilisateur. Pour ce faire, nous avons mis en place un indicateur à même de déterminer la qualité de la meilleure proposition du SLU en fonction d'une référence. En raison du fait que les étiquettes sémantiques $acttype(concept = valeur)$ n'étaient pas alignées aux mots dans le corpus considéré (ici DSTC2) et sachant que ce dernier est une condition préalable pour pouvoir simuler l'annotation en séquence de couples forme-de-surface/étiquette-sémantique nous avons donc au préalable du procéder à un alignement automatique similaire à celui proposé dans (Huet et Lefèvre, 2011). Ainsi, à chaque tour, nous avons suffisamment d'informations pour être en mesure de répondre avec précision à l'action de la machine (séquences d'actes de dialogue de référence et leurs alignements aux mots). Ici, un sous-ensemble de transcriptions de l'ensemble d'apprentissage de DSTC2 (750 transcriptions d'énoncés utilisateur) est exploité pour évaluer le modèle d'adaptation en ligne.

Dans notre configuration expérimentale, un utilisateur simulé est employé pour répondre aux actions d'adaptation du modèle pour chaque tour de parole dans le

dialogue d'origine. Cet utilisateur peut faire usage de trois actions distinctes : *Affirm*, *Negate* et *Inform*. Les actions *Affirm* et *Negate* sont employées pour répondre aux demandes de confirmation liées à l'application des actions d'adaptation du modèle (AskAnnotation et YesNoQuestions). L'action *Inform* est utilisée exclusivement dans les échanges supplémentaires ayant lieu dans le cadre de l'action système AskAnnotation (par exemple *Inform(acttype=request)*, *Inform(boundaries="austrian food")*). Ici, nous supposons que les sous-dialogues d'annotation peuvent être gérés par un système réel avec un niveau de précision élevé (par exemple en utilisant une grammaire bien calibrée et une logique d'interaction finement réglée). Bien sur cette hypothèse devra être confirmée en pratique.

Dans ce travail, nous avons délibérément dégradé K en enlevant quelques concepts importants tels que *name* et *signature* et des valeurs (par exemple en gardant seulement 11 valeurs pour le concept *food*). Ainsi, nous commençons avec une F-mesure plus faible de $0,70$ sur les transcriptions du corpus de test DSTC2. Au total, 404 exemples sont considérés et assignés à 78 actes de dialogue différents (sur les 663 possibles d'après l'ontologie d'origine). Du fait que la technique Exp3 emploie une certaine forme d'exploration stochastique (ici $\gamma' = 0,2$) nous utiliserons 20 processus indépendants d'apprentissage en ligne. Ainsi, les résultats présentés plus bas pour cette méthode correspondront en fait aux moyennes de ceux observés sur ces 20 processus distincts.

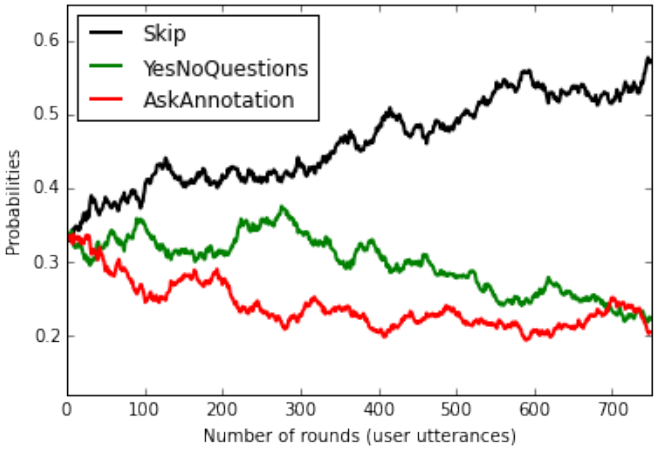


FIGURE 5.6 – *Distribution de probabilité estimée par l'Exp3 au cours du temps sur les différentes actions.*

La figure 5.6 donne l'évolution de la probabilité $p_i(t)$ associée à chaque action i telle qu'estimée par l'algorithme Exp3 ($\gamma = 0,5$). Nous pouvons observer que chaque action est sélectionnée avec une probabilité comparable au début de la procédure d'optimisation, Exp3 explore. Puis, à mesure que le nombre de tours considérés augmente, on observe que l'influence des deux actions YesNoQuestions et Skip croît. On remarque cependant un avantage clair à l'action Skip lorsqu'il devient plus difficile d'obtenir de nouvelles informations eu égard au coût impliqué pour les collecter.

Dans la figure 5.7 on compare maintenant l'effet de γ sur la stratégie apprise par

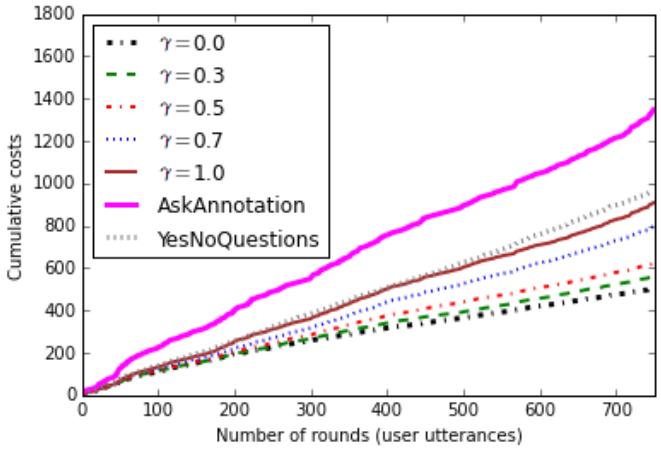


FIGURE 5.7 – *Impact de γ sur l'effort utilisateur (coûts) cumulé.*

Exp3 en terme d'effort utilisateur cumulé. Les stratégies **AskAnnotation** et **YesNoQuestions** (stratégies réalisant la même action à chaque tour) sont introduites ici a des fin de comparaison comme méthodes de référence. Nous considérons les performances pour $\gamma \in \{0,0,3,0,5,0,7,1\}$. Nous pouvons observer que la stratégie **AskAnnotation** est la plus coûteuse, suivie par **YesNoQuestions**. Faire varier le paramètre γ semble avoir l'effet escompté sur l'apprentissage de la stratégie d'adaptation. Ainsi, plus γ est grand, moins le coût a un impact sur l'apprentissage. De ce fait, lorsque celui-ci est totalement ignoré dans la fonction de perte ($\gamma = 1,0$), l'algorithme Exp3 a tendance à favoriser les actions les plus coûteuses car elles permettent de réduire significativement la mesure d'inefficacité du modèle. Ainsi, γ permet de régler le compromis entre l'effort de l'utilisateur et l'efficacité du modèle pour une application donnée.

Enfin, dans la figure 5.8 Exp3 ($\gamma = 0,5$) est comparée à **Askannotation** et **YesNoQuestion** en termes de F-mesure sur les transcriptions du corpus de test DSTC2. Comme prévu **AskAnnotation** obtient les meilleures performances. En effet, l'utilisation des nouvelles annotations permet au modèle ZSSP de couvrir dynamiquement des actes de dialogue supplémentaires grâce à la mise à jour K avec des exemples robustes. En raison du fait que l'objectif de l'algorithme Exp3 est de trouver un compromis entre le fait de réduire l'effort de l'utilisateur et l'efficacité du modèle, cette méthode est capable d'atteindre à plus faible coût des performances proches de celles obtenues avec **AskAnnotation** et bien meilleures que celles observées pour **YesNoQuestion** (cette dernière ne pouvant pas capturer de nouveaux concepts).

5.3.5 Apprentissage supervisé du modèle

Pour faire le lien entre notre proposition et les approches supervisées standards, nous considérons un modèle de CRF utilisant les descripteurs bigrammes et unigramme sur une fenêtre de voisinage de taille 2 autour du mot courant, appris à l'aide de

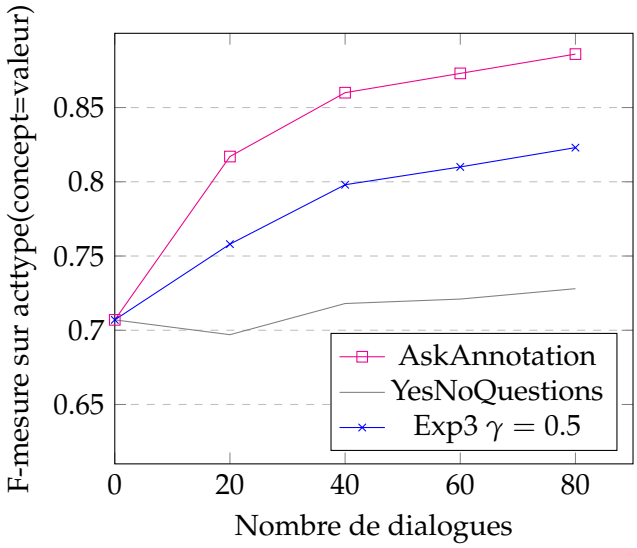


FIGURE 5.8 – *Impact du nombre de dialogues employés sur les différentes techniques d'adaptation en ligne en terme de F-mesure.*

wapiti [11] (Lavergne et al., 2010) sur la totalité des données d'apprentissage relative à chaque tâche (pour DSTC3 ce corpus se limite à 10 dialogues annotés). Ce modèle constituera notre référence pour comparer la performance de notre proposition (il sera noté **CRF-trained** 5.2). Sachant que l'annotation sémantique (séquences d'*acttype(concept=valeur)*) constitue un pré-requis à l'utilisation des CRF et que les corpus considérés ne sont pas alignés aux mots, nous avons eu de nouveau recours à une technique d'alignement non supervisée (Huet et Lefèvre, 2011).

Task	Model	F-mesure	P	R
DSTC2	**CRF-trained**	0.851	0.869	0.835
	word-parser	0.679	0.781	0.600
	chunk-parser	0.786	0.769	0.803
DSTC3	**CRF-trained**	0.606	0.567	0.649
	word-parser	0.552	0.685	0.462
	chunk-parser	0.817	0.786	0.851

TABLE 5.2 – *Evaluation des performances de l'analyseur sémantique basé sur l'apprentissage sans données de référence en terme de F-mesure, **P**récision et **R**appel sur la meilleure hypothèse sémantique.*

Les résultats présentés dans le tableau 5.2 montrent qu'en l'absence de données d'apprentissage de référence, le **chunk-parser** obtient de meilleures performances que le **word-parser** en terme de F-mesure (0,786 contre 0,679 pour DSTC2 et 0,817 contre 0,552 pour DSTC3). Nous montrons également que, malgré ses performances inférieures à un modèle CRF appris sur une grande quantité de données annotées (0,786 contre 0,851 sur DSTC2), le modèle **chunk-parser** donne de meilleurs résultats que

11. https ://wapiti.limsi.fr/

l'approche à base de CRF quand une petite quantité de données d'apprentissage est disponible (0,817 contre 0,606 sur DSTC3).

Même si ces résultats ne sont pas présentés dans le tableau 5.2, le **chunk-parser** obtient instantanément (au démarrage) des performances comparables [12] à celles obtenues par le système à base de règles dont les sorties sont fournies dans les données DSTC (0,786 contre 0,782 sur DSTC2 et 0,817 contre 0,824 sur DSTC3). Il en est de même pour le modèle statistique SLU1 présenté dans (Williams, 2014) (0,786 contre 0,803). Ainsi, l'approche proposée atteint des performances proches de l'état de l'art sans règles spécifiques (coût d'expert humain) ni données d'apprentissage en contexte (coût des annotateurs).

Dans la mesure où le modèle CRF appris sur une grande quantité de données atteint les meilleures performances (**CRF-trained** *DSTC2*), nous avons essayé de montrer l'impact de données d'apprentissage supplémentaires dans les approches proposées. Pour cela, nous proposons d'ajouter progressivement de nouveaux exemples d'apprentissage annotés à la base de connaissances et de ré-estimer dynamiquement le modèle λ_F avec ces nouvelles données contextuelles entrantes. Afin d'avoir un point de comparaison, nous considérons également un modèle CRF appris sur ces même données.

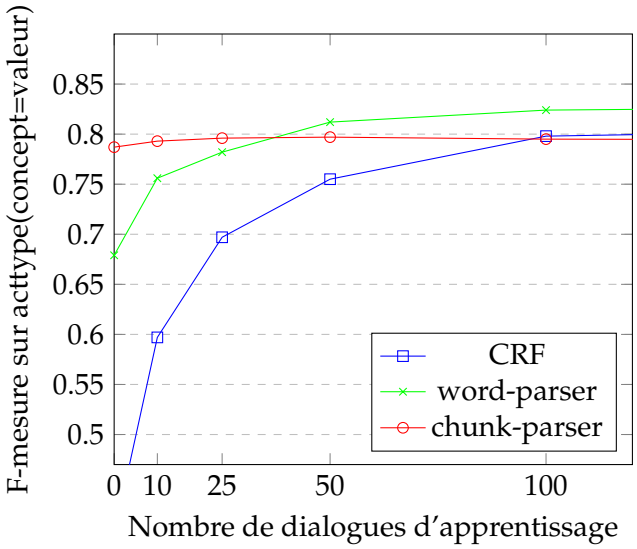


FIGURE 5.9 – *Impact de la configuration de l'analyseur sémantique sur l'apprentissage DSTC2.*

Les résultats présentés dans la figure 5.9 montrent que le *chunk-parser* constitue une bonne option dans une situation où il n'y a pas ou peu de données d'apprentissage. Cependant, les modèles **word-parser** et **CRF** atteignent de meilleures performances après 50 (382 phrases) et 100 dialogues (782 phrases) respectivement. En effet, les descripteurs segmentaux (*chunk*) sont plus rigides et moins fréquents dans les données d'apprentissage que lorsqu'on considère uniquement les mots. Ainsi ces descripteurs nous

12. Il est important de noter que pour les règles et l'analyseur sémantique de Williams la liste des n-meilleures ASR est considérée comme entrée à la place de la meilleure (1-meilleure) dans notre expérience.

permettent de traiter efficacement une situation initiale où l'on manque de données contextuelles, mais diminue la robustesse du modèle lorsque des données deviennent disponibles en quantité suffisante. La raison principale qui peut expliquer la différence de comportent entre l'approche ZSSP standard et sa version supervisée est que la technique d'adaptation en ligne présentée plus haut offre un mécanisme permettant d'intégrer également les informations négatives lors de la mise à jour de K (ce que ne permet pas un modèle CRF). Grâce à cela, elle dispose donc d'un mécanisme implicite favorisant l'exploration et lui permettant de faire usage plus efficace des descripteurs segmentaux sur la durée. L'approche *word-parser* semble cependant pouvoir bénéficier plus longtemps des capacités de généralisation apportées par F au prix de moins bonnes performances en début d'apprentissage. On peut néanmoins constater que plus il y a de données d'apprentissage plus les performances de CRF sont proches de *word-parser* (généralisation marginale) ce qui justifie parfaitement l'utilité des approches supervisées lorsque la quantité de données d'apprentissage est suffisante.

5.4 Bilan

Dans ce chapitre nous avons présenté une approche d'apprentissage sans données de référence pour la compréhension de la parole. Cette dernière repose à la fois sur l'utilisation d'une représentation sémantique riche apprise sur des données généralistes et sur une description ontologique minimale décrivant la tâche de compréhension visée. Nous avons montré que cette approche, bien que peu coûteuse, est tout de même comparable en terme de performances à des modèles statistiques appris sur de grande quantité de données annotées et aussi à un système à bases de règles expertes. De plus, la méthode proposée montre une meilleur tolérance à des valeurs de concept manquantes et donc offre des propriétés de généralisation pouvant être employées notamment dans l'extension de domaine en ligne.

De plus, nous avons montré qu'un processus d'adaptation simple et ajustable en ligne permet de répondre aux deux limites de l'approche, à savoir la qualité de la base de connaissance K et de l'espace sémantique employé F, et ce même lorsque l'effort de supervision se limite à la confirmation des hypothèses faites par le système. Pour dépasser ce cadre d'adaptation simple une approche de bandit contre un adversaire a été employée pour optimiser la stratégie d'adaptation du modèle sans données de références et permettre de résoudre le problème d'une couverture initiale limitée sur la sémantique de domaine spécifique. Il a été montré que cette technique est efficace et à même de fournir un moyen pratique de formaliser un compromis entre l'effort de supervision de l'utilisateur et l'amélioration de l'efficacité du système.

Par ailleurs, nous avons montré qu'il était possible d'établir un formalisme unifié entre cette méthode et une approche supervisée de référence, à savoir les CRF. De ce fait, la méthode peut également exploiter des données apprentissage plus conventionnelle et bénéficier d'un cadre formel qui pourra lui aussi être raffiné par le biais de techniques d'AL.

La généralisation de l'approche d'optimisation proposée (par exemple le nombre

d'actions employées ou encore la prise en compte explicite du contexte) ainsi qu'une comparaison plus poussée avec d'autres algorithmes de bandit fera l'objet de futurs travaux. Une autre piste d'étude est celle de l'intégration de la stratégie d'adaptation proposée dans un système de dialogue bout en bout toujours avec pour objectif d'exploiter au mieux les interactions réalisées avec de vrais utilisateurs. L'objectif visé étant d'étudier à la fois son effet sur la progression globale de dialogue (exécution de la tâche et la satisfaction de l'utilisateur) ainsi que sur l'apprentissage de la stratégie du gestionnaire de dialogue.

Chapitre 6

Application au Dialogue Homme-Robot et apports de l'aspect situé

Sommaire

Dans ce chapitre nous présentons plus en détail les objectifs du projet ANR *MaRDi* ainsi que nos propositions quant à la mise en œuvre d'un système de dialogue Homme-Robot situé. Nous donnons également les résultats de nos premières études sur le sujet.

Dans la section 6.1 nous décrirons les objectifs et la tâche retenue dans le projet ANR *MaRDi*. Puis, dans la section suivante, nous présenterons l'architecture mise en œuvre concrètement pour conduire un dialogue situé entre l'Homme et le Robot. Nous détaillerons dans la section 6.3 les différentes conditions d'apprentissage RL et de tests en

ligne de la politique d'interaction envisagée dans cette thèse. Nous profiterons de cette section pour présenter le simulateur robotique employé dans notre étude pour modéliser en 3D l'environnement physique où se déroule l'interaction. La dernière section de ce chapitre sera consacrée à la présentation des résultats obtenus lors de nos expériences réalisées avec le concours de vrais utilisateurs dans les conditions que nous auront préalablement décrites (apprentissage et tests).

Comme nous allons le voir, nous avons profité du contexte particulier qu'est celui d'un développement d'une nouvelle plateforme de dialogue de zéro pour valider sur un cas pratique la proposition d'apprentissage en ligne socialement inspiré présentée dans la section 4.2. La différence majeure introduite ici est que les signaux de renforcement additionnels seront cette fois issus d'évaluations émises tout au long du dialogue par de vrais utilisateurs. Nous présenterons ainsi les premiers résultats obtenus quant à l'apport de cette technique sur la problématique de l'apprentissage RL en ligne de la politique d'interaction ainsi qu'à ses capacité d'adaptation à un nouveau profil utilisateur (nouveaux comportements).

Nous nous intéressons également dans ce travail à l'intérêt de la prise en compte de l'information située dans la gestion du dialogue, plus exactement de celle la « prise de perspective » du robot du fait de son incarnation physique dans la réalité (son aspect situé). Cette notion, bien connue dans le domaine de la psychologie du développement, se réfère à la capacité qu'a une personne à pouvoir concevoir qu'une autre puisse avoir une vision du monde qui diffère sensiblement de la sienne, mais également d'émettre des hypothèses sur son état mental courant (connaissances factuelles). Ainsi, à l'instar de récents travaux dans le domaine de la robotique, nous cherchons à doter le robot de mécanismes similaires pour tendre vers une gestion de l'interaction plus efficace et naturelle. Nous abordons ce défi en proposant d'étendre le paradigme POMDP HIS (Young et al., 2010) présenté dans la section 3.4.4. De fait, l'extension proposée permet d'intégrer dans le mécanisme de prise de décisions (espaces d'état et d'action) les croyances factuelles des différents agents engagés dans le dialogue. L'idée poursuivie est celle de pouvoir proposer un cadre formel unifié capable à la fois de faire face aux ambiguïtés liées au canal de communication bruitée mais aussi celles dues à une divergence entre les croyances factuelles du robot et de l'utilisateur (nous reviendrons sur ce point dans la section 6.2.2). Nous montrons la pertinence de l'approche en comparant une politique de dialogue apprise avec et sans ces mécanismes de raisonnement avec de vrais utilisateurs.

6.1 *MaRDi* : objectifs et description de la tâche

L'objectif du projet ANR *MaRDi* est d'étudier l'apport d'une approche « située » pour le dialogue Homme-Machine. Ce projet, qui a débuté en octobre 2012, est financé dans le cadre de l'appel à projet Contenu et Interactions et est effectué en collaboration avec le Laboratoire d'Informatique Fondamentale de Lille (LIFL), l'École supérieure d'électricité (Supélec), le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), le groupe Acapela et le Laboratoire Informatique d'Avigon (LIA). La notion de mise en

situation est ici relative à l'incarnation physique du système de dialogue au travers d'une plateforme robotique. Dans cette configuration, la réalité telle que perçue par un robot peut alimenter le contexte de l'interaction pour compléter ou lever des ambiguïtés introduites par le médium vocal.

Pour atteindre ce but, la Machine doit être capable de :

— maintenir un contexte d'interaction suffisant riche pour pouvoir être à même de prendre des décisions sur la suite à donner à celle-ci. Ce contexte intègrera les entrées fournies par l'humain, les informations issues de la perception de l'environnement par le robot, mais aussi ses données proprioceptives (mesure du robot sur lui-même comme par exemple l'angle de rotation de ses armatures, le niveau de charge de sa batterie, etc.) ;
— prendre des décisions pour poursuivre l'interaction sur la base de ce contexte. Il faudra également qu'il tienne compte de son aspect incertain du fait des possibles erreurs introduites par la chaîne de traitement automatique des entrées vocales et visuelles ;
— restituer de façon expressive sa compréhension du contexte (y compris spatial) pour aider l'humain à la réalisation d'une interaction efficace ;
— planifier des mouvements et des actions physiques précises pour répondre aux besoins de utilisateurs (déplacement d'objets, etc.).

Afin de positionner le scénario dans un cadre naturel et fonctionnel, le choix a été fait de faire interagir un robot assistant avec un handicapé dans son appartement (aide à la personne). Du fait de ce handicap, ce dernier interagira avec le robot pour que ce dernier l'aide à manipuler divers objets. Ces derniers auront des propriétés en termes de couleur, de type et de position et se verront attribuer un identifiant unique (par exemple, un livre bleu sur la table de la cuisine avec *idobj*=BLUE_BOOK). Dans la version actuelle du scénario seuls des objets considérés comme facilement manipulables par le robot sont employés (livres, cassette vidéos, tasses, etc.). Pour se faire comprendre par le robot, l'utilisateur usera principalement de la parole, mais pourra également employer des gestes déictiques (comme le fait de pointer un objet particulier).

Un dialogue multimodal sera ici employé pour pouvoir résoudre les possibles ambiguïtés liées à une précision insuffisante de la requête utilisateur, à la qualité des traitements (taux d'erreurs de transcription/compréhension de la parole) voire au contexte physique courant de l'interaction (angle de vue du robot).

Ce dialogue se poursuivra jusqu'à la fin de l'exécution effective de la tâche ou l'échec de l'interaction. Cette dernière situation peut par exemple être due à un désengagement explicite de l'utilisateur, ou encore à l'exécution d'une commande erronée de la part du robot. Un exemple d'un tel dialogue multimodal est donné dans le tableau 6.1.

Dans ce contexte d'étude nous pourrons également jouer sur la définition de divers scénarios (configuration de l'environnement, connaissances initiales du robot, position des agents, etc.) pour rendre l'interaction plus ou moins complexe. Ceci nous permet de pouvoir tester divers cas d'interactions, notamment dans des configurations « limites » comme celles employées lorsque nous étudierons les tâches de fausses croyances (voir

$Robot_1$>	Comment puis-je vous aider ? hello() [Greet]
Usr_1>	Peux-tu mettre le livre sur ma table de chevet ? inform(cmd.action=move, cmd.location=bedroom_bedsidetable)
$Robot_2$>	Désolé, mais où se trouve le livre dont vous me parlez ? confreq(object.type=book,object.location) [TentRQ]
Usr_2>	C'est celui là inform(idobj= ?) + pointsAt BLUE_BOOK = inform(idobj=BLUE_BOOK)
$Robot_3$>	Ok, je vais mettre le livre bleu sur votre table de chevet execute(cmd.action=move,cmd.location=bedroom_bedsidetable, idobj=BLUE_BOOK,object.location=livingroom_table,object.type=book, object.color=blue) [Execute] move(BLUE_BOOK,livingroom_table,bedroom_bedsidetable)

TABLE 6.1 – *Exemple de dialogue multimodal sur la tâche MaRDi.*

section 6.4.3).

FIGURE 6.1 – *Robot PR2 de Willow Garage*

La plateforme robotique retenue pour notre étude est le PR2 [1] (pour *Personnal Robot 2*, voir figure 6.1). Il s'agit d'un robot commercialisé par la société Willow Garage reposant essentiellement sur des logiciels libres (système d'exploitation Linux, *middleware* libre ROS [2]). De taille humaine, ce robot possède deux bras avec sept degrés de liberté qui lui permettent notamment de réaliser des tâches de manipulation fines et d'adopter des postures expressives. Il repose également sur une base mobile afin de pouvoir naviguer dans l'environnement et est équipé de nombreux capteurs (camera, télémètre laser, capteur tactile sur les pinces, etc.). De par ses propriétés et sa large communauté cette plateforme est employée dans de nombreuses équipes de robotique spécialisées dans l'HRI ; dont le LAAS-CNRS qui est notre partenaire dans le projet *MaRDi*. De

1. https ://www.willowgarage.com/pages/pr2/overview
2. http ://www.ros.org

même, l'appartement choisi pour conduire nos expériences est la réplique taille réelle d'un 3 pièces (salon, cuisine, chambre) présent dans leurs locaux et dont une modélisation fidèle a été faite sur simulateur 3D (voir figure 6.2).

FIGURE 6.2 – *Environnement 3D MaRDi*

6.2 Architecture retenue pour le dialogue Homme-Robot

Le système que l'on cherche à développer dans le cadre du projet *MaRDi* doit pouvoir traiter des modalités responsables de l'analyse visuelle de la scène ainsi que de la reconnaissance et la compréhension de la parole. Par défaut, ces modalités fournissent des informations qui peuvent se renforcer, être complémentaires ou incompatibles (voir section 2.1.2).

Parmi elles, certaines sont destinées à fournir des informations situées pour mettre à jour le contexte de l'interaction (positions des objets et des agents, leur état mental, etc.) constituant la base de faits dynamiques, telles que les relations spatiales entre les objets et les agents présents dans l'environnement (par exemple, « la tasse rouge est sur la table »). D'autres informations comme la parole et les gestes déictiques sont analysées conjointement au travers d'un mécanisme de fusion tenant compte du contexte courant pour mener à bien la tâche de dialogue.

En accord avec la tâche décrite dans la section 6.1, trois modalités sont retenues dans le système actuel et associées aux actes communicatifs du robot et de l'utilisateur :

— la parole : expression en langage naturel pour utilisateur et pour le robot ;
— le geste : limité à la désignation d'objets pour l'utilisateur et à la désignation et la manipulation d'objets pour le robot ;
— la vision : acquisition d'informations situées (mises à jour de la base de connaissances dynamiques).

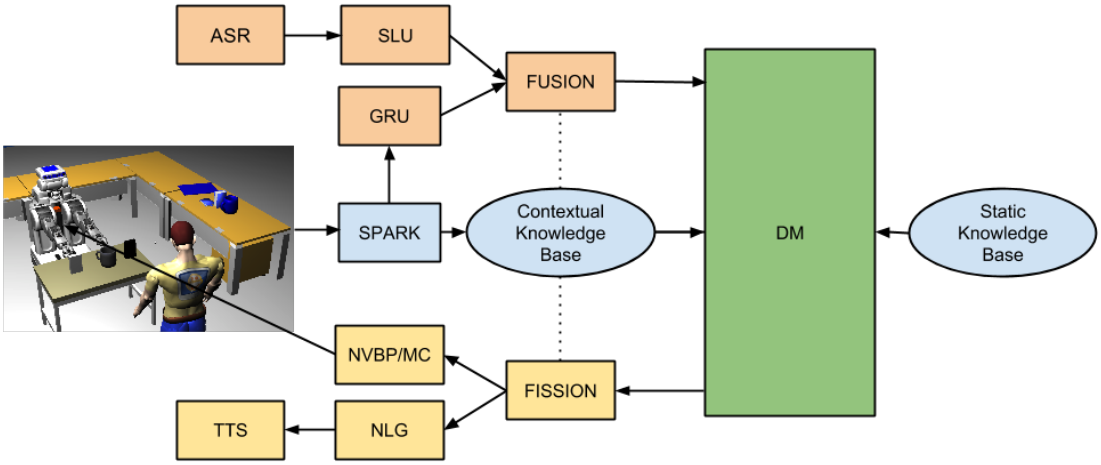


FIGURE 6.3 – *Architecture multimodale pour le dialogue situé.*

La figure 6.3 présente l'architecture du système employée dans le cadre du projet *MaRDi*. Au total, ce sont douze modules qui sont impliqués dans le fonctionnement global de la solution retenue. Nous allons les présenter au travers des quatre prochaines sections qui les regroupent selon leur fonction de « haut niveau ».

6.2.1 Gestion et compréhension des entrées multimodales de l'utilisateur

Dans notre contexte applicatif, l'utilisateur peut faire l'usage de la parole et/ou de gestes déictiques de façon non contrainte tout au long de l'interaction pour s'adresser au robot. Les quatre modules représentés en orange sur la figure 6.3 ont la charge d'extraire l'information sémantique résultant de l'analyse des différentes modalités à chaque tour de dialogue sous la forme d'une liste unifiée de N-meilleures hypothèses d'actes de dialogue utilisateur.

Représentation sémantique

Avant de poursuivre sur les mécanismes mis en jeu pour effectuer la compréhension des actes dialogiques utilisateur il est utile de faire un point sur la représentation sémantique utilisée dans le cadre du projet *MaRDi*.

Concepts	**Valeurs**
spatial.indicator	near, far, behind, before, right, left, under, on, in
room	livingroom, bedroom, kitchen
color	black, grey, blue, red, etc.
type	mug, book, box, tape, table, bed, coffeetable, etc.
action	put, bring, give, move, locate, etc.

TABLE 6.2 – *Exemple de concepts élémentaires ou « bas niveau » avec leurs valeurs respectives.*

Concepts	Valeurs
object.type	mug, book, box, tape
object.color	black, grey, blue, red, etc.
object.location	livingroom_table, kitchen_table, etc.
cmd.action	give, move
cmd.location	livingroom_table, kitchen_table, etc.

TABLE 6.3 – *Exemple de concepts « haut niveau » avec leurs valeurs respectives.*

Dans cette étude, nous utilisons un formalisme des actes de dialogue proche de celui employé dans le cadre de la tâche *TownInfo* (standard CUED décrit dans l'annexe A.1). Cependant, afin de faciliter l'interprétation sémantique tout en permettant à l'utilisateur une plus grande liberté d'expression, nous avons fait le choix de procéder à une extraction d'éléments sémantiques selon deux niveaux de granularité :

— un premier niveau, appelé par la suite « bas niveau », est constitué d'un ensemble de concepts élémentaires (voir tableau 6.2). Ces derniers sont plus nombreux que ceux effectivement manipulés par le gestionnaire de dialogue et s'apparentent plus à une analyse littérale de l'énoncé utilisateur ;
— un second niveau, dit « haut niveau », contient quant à lui uniquement les concepts ontologiques effectivement manipulés par le gestionnaire de dialogue (voir tableau 6.3). Nous employons ici la notion d'ontologie dans le sens d'une description structurée des éléments sémantiques utilisés pour modéliser le but utilisateur. Un exemple de concept « haut niveau » est *object.location*. Ce dernier représente la position de l'objet « sujet » de la commande de manipulation utilisateur. Sa valeur doit correspondre à une des positions d'intérêts identifiées lors de la quantification spatiale de l'environnement. Dans la version actuelle de la plateforme de dialogue, ces positions ont été définies pour correspondre aux différents meubles de l'appartement (emplacements où peuvent se trouver les objets à manipuler).

Le décodage sémantique d'un énoncé utilisateur va donc se faire en trois temps. En premier lieu le module SLU va extraire le contenu sémantique bas niveau de l'énoncé utilisateur sous forme d'une liste d'hypothèses sémantiques scorées. Une fois les hypothèses des diverses modalités combinées (nous détaillons ce processus plus bas), les concepts bas niveau vont être regroupés en segments pour ensuite être associés à des concepts haut niveau. Enfin, certains d'entre eux devront être résolus, autrement dit il faudra déterminer leur valeur par l'analyse des concepts de bas niveau qui y sont rattachés (segment). C'est par exemple le cas pour *object.location* pour lequel il faudra dans un premier temps identifier un ensemble de relations spatiales puis les soumettre au module de raisonnement spatial pour obtenir un liste ordonnée sur les positions d'intérêts candidates (valeurs possibles pour le concept). Un exemple de ce processus est illustré le tableau 6.4.

Nous décrivons ci-dessous les modules mis en jeu pour réaliser la compréhension de la parole et des gestes utilisateur avant de décrire la solution retenue pour la fusion dans notre étude.

Phrase utilisateur	Passe moi le livre bleu qui est sur la table du salon
Sémantique bas niveau	inform(action=give, type=book, color=blue, spatial.indicator=on, type=table, room=livingroom)
Regroupement	inform(cmd.action=give, object.type=book, object.color=blue object.location=[spatial.indicator=on, type=table, room=livingroom])
Sémantique haut niveau	inform(cmd.action=give, object.type=book, object.color=blue, object.position=livingroom_table)

TABLE 6.4 – *Exemple d'extraction sémantique complète de la tâche MaRDi.*

Compréhension de la parole

La reconnaissance vocale est effectuée grâce à la *Google Web Speech API*[3]. Cette dernière nous offre l'accès à un ASR grand vocabulaire état de l'art en langue française dont le niveau de performance et les temps de réponses (transcriptions incrémentales en temps réel des entrées vocales sur un navigateur web) sont très satisfaisants pour le contexte de notre étude. Ainsi, à chaque tour de parole utilisateur, une liste des N meilleures hypothèses scorées (confiances) de transcription est mise à disposition du système ($N = 5$ dans nos travaux).

Deux implémentations différentes du module SLU ont été pour l'instant étudiées dans le cadre de la tâche *MaRDi*. La première repose sur une grammaire non contextuelle et la seconde sur l'approche sans données de références proposée dans le chapitre 5. Nous avons constaté que cette dernière a nécessité un effort de conception bien moindre pour atteindre un niveau de performance équivalent (en plus de ses capacités de généralisation dans le cas où l'on souhaiterait étendre le domaine). Comme nous l'avons dit précédemment, les hypothèses sémantiques du module SLU se présentent sous la forme d'actes de dialogue faisant intervenir des concepts bas niveau. Il est à noter que le recours à une solution SLU adaptative, à l'instar de ce qui a été proposé dans le chapitre 5, fera l'objet de travaux ultérieurs. Avant cela, il nous faudra tout d'abord évaluer l'impact théorique et pratique d'une amélioration progressive des performances du module SLU sur l'optimisation globale de la gestion de l'interaction. Il est à noter que des approches supervisées pourront également être envisagées une fois plus de données collectées et annotées par nos soins.

Compréhension déictique

Afin de capturer les gestes déictiques émis par l'utilisateur lors de son tour d'interaction, nous employons le module de reconnaissance et compréhension des gestes (*Gesture Recognition and Understanding* - GRU). Dans la configuration standard de la plateforme, les gestes sont détectés et interprétés dynamiquement par le raisonneur spatial SPARK (Milliez et al., 2014). Ce dernier exploite à la fois les coordonnées spatiales des objets et les jointures de l'utilisateur telles que déterminées grâce aux informations is-

3. https ://www.google.com/intl/en/chrome/demos/speech.html

sues des capteurs visuels du robot pour savoir si oui ou non un objet est désigné du doigt par l'utilisateur. Lorsque c'est le cas, un évènement de la forme *AGENT_ID pointsAt OBJECT_ID* est alors généré. Ce dernier est alors associé à un marqueur temporel (temps en secondes depuis le 1er janvier 1970 00 :00) pour simplifier le mécanisme de fusion avec les entrées vocales.

Dans la version actuelle de la plateforme des heuristiques expertes sont employées pour la capture de ces gestes dans SPARK. Cependant, une fois que plus de données auront été collectées, des techniques plus élaborées pourront être envisagées pour les remplacer, comme par exemple celle proposée dans (Rossi et al., 2013) qui fait intervenir un classifieur HMM avec en entrée des données issues d'une caméra RGB-D (coordonnées 3D et angles des jointures du corps de l'utilisateur, état ouvert/fermé de chacune de ses mains, etc.).

Fusion

L'objectif du mécanisme de fusion est de combiner les actes de dialogue extraits du signal de parole utilisateur aux évènements déictiques capturés grâce au module GRU. Pour ce faire, il faut tenir compte à la fois du contexte de l'interaction (positions des objets dans l'environnement physique, etc.), du niveau confiance que l'on porte aux différentes hypothèses unimodales (étant données qu'elles peuvent être erronées) mais également à leur marqueur temporel.

La première étape de ce processus consiste donc à déterminer si les hypothèses en provenance des différentes modalités sont synchrones entre elles et peuvent être fusionnées ou doivent être considérées séparément. Du fait que la parole est considérée dans notre étude comme modalité principale de l'utilisateur, les tours d'interaction seront calés sur celui des entrées vocales. Ainsi, comme dans (Holzapfel et al., 2004), seuls les gestes déictiques détectés dans un segment temporel de 20ms avant et après celui du tour de parole courant seront exploités par le mécanisme de fusion. De ce fait, si des hypothèses SLU apparaissent seules ou que les gestes détectés ne leur sont pas synchrones, elles seront directement considérées comme résultat de la fusion.

La méthode de fusion retenue dans nos travaux repose sur la définition d'un ensemble de règles. Cependant, elle s'attache à intégrer un mécanisme permettant de propager l'incertitude donnée par les capteurs (ASR compris) sur les entrées unimodales. Pour ce faire, elle exploite les scores de confiance obtenues en sortie du SLU et intègre les incertitudes liées à la prise en compte des hypothèses du module de détection de gestes GRU. L'objectif visé est de pouvoir considérer les scores associés aux hypothèses du module de fusion comme des scores de confiance pour les traitements supérieurs (décisionnels). Cette implémentation se concentre surtout sur le problème de la désambiguïsation des hypothèses vocales. Par exemple si l'utilisateur prononce la phrase « prends ça » tout en désignant un objet du doigt la fusion a pour rôle principal d'identifier un candidat valable. Pour ce faire le mécanisme de fusion s'appuie notamment sur la détection de concepts bas niveau qui témoignent d'un besoin de résolution de référents dans l'énoncé utilisateur (le mot « ça » dans l'exemple précédent).

La dernière étape du processus consiste à convertir les hypothèses ainsi produites dans leur représentation sémantique haut niveau pour pouvoir les transmettre au DM. Des heuristiques définies manuellement sont employées pour déterminer les valeurs des concepts de haut niveau identifiés à partir des hypothèses bas niveau.

Bien que des solutions par règles soient ici retenues, leurs limitations théoriques et pratiques constituent pour nous un obstacle à leur maintien dans la plateforme de dialogue sur le long terme. Le recours à des approches supervisées, comme c'est le cas dans (Rossi et al., 2013), ou exploitant des notions empruntées à la logique floue, à l'instar des travaux présentés dans (Reddy et Basir, 2010), pourra être envisagé une fois que plus de données auront été collectées et annotées (ou qu'une solution à partir de zéro aura été élaborée à l'instar de nos travaux en SLU).

6.2.2 Modélisation du contexte

La modélisation du contexte de l'interaction joue un rôle déterminant dans une application HRI. Grâce a elle le robot peut modéliser dynamiquement l'environnement géométrique avec lequel il est en train d'interagir et ainsi en avoir une représentation symbolique adaptée au raisonnement logique. Cette modélisation est rendue possible grâce aux capacités perceptives et de raisonnement à même d'extraire des informations de haut niveau (reconnaissance d'objets, localisation, identification de relations spatiales, etc.).

Dans notre configuration le système dispose à la fois d'une base statique de connaissances contenant la liste de tous les objets connus du robot (même ceux non encore perçus durant l'interaction) et de leurs propriétés statiques (couleur, identifiant, etc.) mais aussi d'une base dynamique de connaissances dans laquelle sont stockées les informations contextuelles. Cette base se présente sous la forme d'un ensemble de faits symboliques représentant les propriétés dynamiques de l'environnement, que ce soit celles dites « géométriques » (positions relatives des différents objets/agent) ou celles déterminées par raisonnement sur les observations visuelles du robot (visibilité/accessibilité des divers objet pour chaque agent). A titre d'illustration le tableau 6.5 donne une liste, non exhaustive, de ces faits.

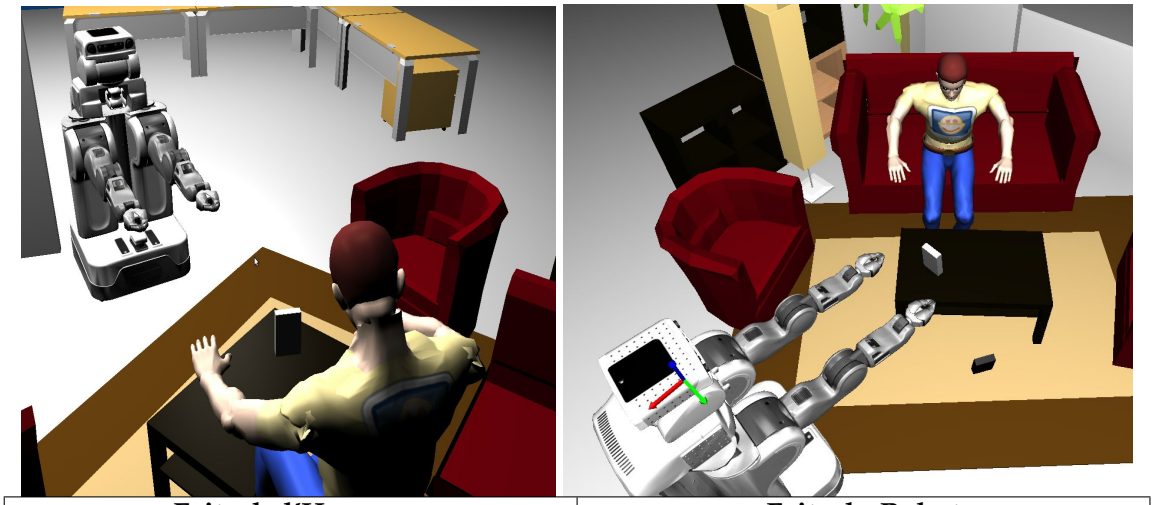
Faits	**Descriptions**
O_1 isOn O_2	L'objet O_1 est sur O_2
O_1 isIn O_2	L'objet O_1 est dans O_2
O_2 isNear O_2	L'objet O_1 est à proximité de O_2
O_1 isAt Z_1	L'entité E_1 (objet/agent) se trouve dans la zone Z_1
O_1 isVisible [true | false]	O_1 est visible (ou non) pour l'agent principal
O_1 isReachable [true | false]	O_1 est atteignable (ou non) pour l'agent principal

TABLE 6.5 – *Exemples de faits symboliques servant à décrire le contexte de l'interaction.*

Pour mener à bien une tâche d'interaction située, le robot doit considérer les utilisateurs à la fois comme des entités physiques (par exemple sur lesquelles il ne faudra pas

rouler), mais aussi et surtout comme des entités intelligentes, dotées d'une individualité et de capacités cognitives qui leur sont propres. Pour pouvoir agir efficacement, le robot doit donc être doté de capacités lui permettant de représenter fidèlement son environnement et de modéliser les perceptions/connaissances présumées des utilisateurs avec lesquels il interagit.

L'une des particularités de notre module de gestion du contexte est qu'il permet de gérer plusieurs modèles symboliques en parallèle, à savoir un par agent (y compris le robot), chaque modèle étant indépendant et cohérent d'un point de vue logique. Cette décomposition permet notamment au robot de pouvoir raisonner sur plusieurs perspectives cognitives du même environnement. Ces perspectives peuvent notamment être incohérentes lorsqu'on les compares deux à deux. Le tableau 6.6 rapporte un exemple dans lequel deux objets sont visibles dans le modèle du robot alors que pour l'homme il n'y en a qu'un seul (l'autre n'étant pas dans son champ de vision). Ainsi, l'objet BLACK_TAPE a simultanément la propriété *isVisible* à la valeur *true* et *false* selon le modèle considéré (situation d'incohérence).



Faits de l'Homme	**Faits du Robot**
GRAY_TAPE isVisible true GRAY_TAPE isOn livingroom_coffeetable BLACK_TAPE isVisible false ...	GRAY_TAPE isVisible true GRAY_TAPE isOn livingroom_coffeetable BLACK_TAPE isVisible true BLACK_TAPE isOn floor ...

TABLE 6.6 – *Exemple d'une situation d'interaction où les faits symboliques ne sont pas les mêmes dans les modèles de l'Homme et du Robot.*

Il est à noter que la gestion dynamique des faits dans le temps peut amener le robot à modéliser des « croyances », par exemple le fait que l'homme ai connaissance de la position d'un objet même si à cet instant précis ce dernier n'est plus visible de son point de vue, car il vient de s'assoir par exemple. Ainsi le point important est que la modélisation par agent des connaissances permet à la machine de prendre en compte différents points de vues (ou perspectives) du monde dans la gestion de l'interaction et

non uniquement des informations ayant pour référentiel le robot.

Dans la plateforme actuelle, le module SPARK (Milliez et al., 2014) employé pour le GRU a également à sa charge l'alimentation et le maintien des faits dynamiques (notamment ceux visuels) dans la base de connaissances grâce à ses capacités avancées en raisonnement spatial.

Le raisonnement sur les perspectives

De nombreux travaux dans la littérature robotique ont montré que prendre en compte la perspective des différents agents qui partagent l'environnement du robot peut sensiblement améliorer les capacités d'interprétation de la situation courante de ce dernier, mais aussi rendre plus efficace sa stratégie de planification des tâches (Breazeal et al., 2006, 2009; Milliez et al., 2014).

Dans le cadre de la théorie de l'esprit (*Theory of Mind* en anglais), la prise de perspective est une capacité largement étudiée dans la littérature de la science du développement. Ce terme général englobe la prise de perspective visuelle (niveau 1) qui consiste à comprendre que d'autres personnes peuvent voir/percevoir le monde différemment, mais également celle conceptuelle (niveau 2) qui consiste à émettre des hypothèses quant aux connaissances, pensées et sentiments des autres personnes (Baron-Cohen et al., 1985).

Dans (Tversky et al., 1999) les auteurs expliquent dans quelle mesure la prise en compte de la perspective dans les actes communicatifs peut améliorer l'efficacité globale du dialogue lorsqu'on la compare à une situation où l'interprétation faite par le robot serait purement « égocentrique ».

Afin de développer des agents plus socialement compétents, de nombreuses recherches se sont concentrées sur le fait de doter les robots de telles capacités. Entre autres, les auteurs de (Breazeal et al., 2006) ont proposé une approche permettant à un robot d'apprendre des séquences particulières d'activation/désactivation de lampes de couleur effectuées par un instructeur humain tout en tenant compte de la perspective de ce dernier pour clarifier de potentielles ambiguïtés. Dans l'expérience réalisée dans cet article, certaines lampes sont occultées du point de vue de l'utilisateur (du fait de la présence d'une planche) mais pas de celui du robot qui lui les a toujours en visuel (création d'une divergence de perspectives). De ce fait lorsque l'utilisateur veut faire la démonstration de la tâche consistant à « allumer toutes les lampes », sans plus d'information, il le fera selon son point de vue, c'est à dire sans allumer les lampes occultées. Ainsi, si le robot veut pouvoir apprendre correctement il doit tenir compte impérativement de la perspective de l'utilisateur.

Dans (Trafton et al., 2005) les auteurs proposent d'exploiter les informations visuelles et les capacités de prise de perspective spatiale des divers agents pour résoudre des référents indiqués vocalement par un partenaire humain afin d'effectuer des commandes de déplacement dans une pièce où plusieurs objets peuvent être occultés à la fois du point de vue de l'humain mais aussi celui du robot. Pour ce faire, le robot peut

également faire l'usage d'actions telles qu'explorer des parties de la pièce, ou demander une clarification à l'utilisateur.

Dans la présente étude, nous nous concentrons particulièrement sur une tâche de détection de fausses croyances (prise de perspective conceptuelle). Initialement introduite dans (Wimmer et Perner, 1983), cette catégorie de tâche nécessite la capacité de reconnaître qu'une autre personne peut avoir des croyances sur le monde qui diffèrent de la réalité observée. Dans le domaine de l'HRI, les auteurs de (Breazeal et al., 2009) ont présenté une des premières implémentations de ce type de compétence dans le but d'identifier le véritable but utilisateur sur une tâche inspirée du test de Sally et Anne (Wimmer et Perner, 1983; Baron-Cohen et al., 1985). Ce dernier a notamment été employé dans des études en relation avec l'autisme et fait office de référence pour illustrer la notion de fausse croyance dans la littérature. Dans sa version originale, des enfants assistent à un spectacle de marionnettes. Dans ce dernier figure deux personnages, Sally et Anne, qui au début du spectacle sont tous deux dans une même pièce. Sally dispose d'un panier et d'un sac de billes, et Anne seulement d'une boîte. Sally tire une bille de son sac, la dépose dans son panier, et quitte la pièce. En son absence, Anne va prendre la bille qui est dans le panier de Anne et la cache dans sa boîte. Sally revient alors dans la pièce et l'on pose à chaque enfant la question suivante : « où Sally va-t-elle aller chercher sa bille ? ». Pour réussir le test, l'enfant doit désigner le panier de Sally mais aussi indiquer où se trouve la bille actuellement. Pour ce faire il doit être capable de concevoir qu'un agent (en l'occurrence Sally) peut avoir un état mental qui en l'état diffère du sien et de la réalité physique observée.

Dans (Milliez et al., 2014), les auteurs font l'usage du module de raisonnement spatial SPARK pour faire passer au robot un test similaire au travers du maintien de modèles distincts sur les croyances factuelles des différents agents. Ainsi, le robot se voit attribuer un rôle équivalent à celui de l'enfant et deux utilisateurs jouent en quelque sorte le rôle de Sally et Anne. Au cours du test, le robot doit être capable de modéliser dynamiquement les états mentaux des utilisateurs (possiblement divergents avec le modèle issu des informations capturées depuis sa perspective du monde) lorsque des modifications survenues dans l'environnement ne sont observables que par un des deux participants (qui est également l'auteur de ces changements). La réussite du test est obtenue seulement si le robot à réussi à modéliser ces situations de divergence dans sa base de faits dynamiques (ce qui a été le cas pour le module SPARK).

Considérant cela, nous pensons qu'une bonne gestion des croyances factuelles de la part du robot est à même d'améliorer la politique du système de dialogue. Pour ce faire, nous proposons donc d'introduire cette nouvelle source d'information dans le mécanisme d'optimisation offert par le cadre formel du POMDP par son intégration dans la définition de l'état de croyance du dialogue et l'ajout de mécanismes de traitement permettant d'en tirer partie au travers de la politique de contrôle apprise (nouvelle action). L'objectif étant de combiner les bonnes propriétés du raisonnement sur les croyances divergentes et celles de la gestion de l'incertitude inhérente aux données que le robot doit manipuler (erreurs de transcription automatique, fausses alarmes sur les gestes, etc.).

Gestion des connaissances factuelles des différents agents

Avant de montrer leur intégration dans le formalisme de notre DM, précisons comment sont évalués les faits constituant les croyances des interlocuteurs. Comme nous l'avons déjà mentionné, le module SPARK (Milliez et al., 2014) a la charge d'alimenter la base de connaissances dynamiques [4] de notre système de par une analyse constante de la situation courante et ses capacités de raisonnement spatial. Pour mener à bien sa tâche, ce module exploite trois sources d'information afin d'obtenir une représentation physique et symbolique de son environnement : celles liées aux objets (position, identifiant, etc.), celles permettant de modéliser l'utilisateur (détection des jointures, etc.) et les proprioceptions du robot (sa propre position, sa posture actuelle, etc.).

Un modèle de l'environnement contenant les positions des objets statiques (par exemple les murs, meubles, etc.) est chargé directement dans la base lors de son initialisation. Ces informations permettent d'établir une première cartographie 3D de l'espace. Les autres objets (tasses, DVD, etc.) sont eux considérés comme mobiles et leur position sont donc recueillies dynamiquement à l'aide des informations en provenance de la vision stéréo du robot. Des capteurs de posture RGB-D tels que ceux présents dans la Kinect ou dans l'ASUS Xtion, sont ici employés pour obtenir la position de l'homme ainsi que ces jointures. Ces données permettent au système de mettre à jour le modèle 3D de l'environnement employé pour procéder au raisonnement spatial.

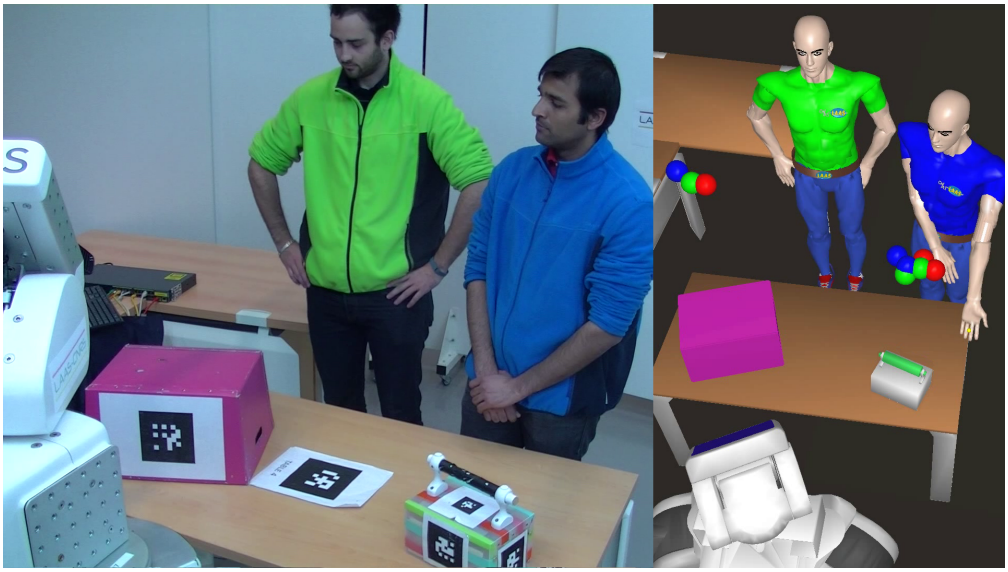


FIGURE 6.4 – *Vrais utilisateurs interagissant avec le robot (à droite) et représentation virtuelle de l'environnement tel que construit par le système (à gauche).*

4. Théoriquement cette base de connaissances dynamiques peut également être mise à jour par l'intermédiaire du dialogue. Elle le sera d'ailleurs, toute proportion gardée, dans notre étude (cas des fausses croyances). C'est par exemple le cas lorsqu'une personne de confiance informe le système de la position courante d'un objet. Cependant compte tenu de l'incertitude sur les entrées vocales, une gestion fine des erreurs devra alors être adoptée.

La figure 6.4 montre un exemple de l'environnement virtuel généré par le système (à droite de l'image) à partir des données recueillies grâce aux capteurs du robot et enrichies par le module de raisonnement spatial. Ce dernier est notamment utilisé pour générer les faits informant sur la position relative des différents objets mais aussi sur les capacités physiques des agents (*affordances* en anglais).

Les *affordances* des divers agents comme *isVisible* et *isReachable* décrivent respectivement les capacités de ces derniers à voir et à atteindre les objets de la scène. Les positions relatives comme *isIn*, *isNextTo*, *isOn* sont utilisées dans la gestion du dialogue multimodal pour résoudre les référents dans les énoncés des utilisateurs et transmettre de façon naturelle les positions des objets dans les réponses du robot.

En ce qui concerne les faits relatifs au robot :
— les *isVisible* sont déterminés directement grâce aux informations issues du module en charge de la reconnaissance des objets et de leur position ;
— les *isReachable* sont déterminés en essayant de trouver pour chaque objet une posture dans laquelle le robot est à même de saisir les objets en question avec ses joints de préhension (détection de collisions) grâce à une technique de cinématique inverse (Baerlocher et Boulic, 2004) (simulation dans le modèle 3D de l'environnement établi par SPARK).

Pour ce qui est des faits relatifs à l'homme, un raisonnement similaire à celui appliqué pour le modèle du robot est adopté pour ceux de type *isReachable*, pour *isVisible* le robot devra cette fois ci déterminer quels objets sont actuellement dans le champ de vision de l'homme dans le modèle 3D (cône émergeant de sa tête). Ainsi, si un objet dans son champ de vision peut être directement relié aux joints représentant la tête de l'homme alors le module de raisonnement fera l'hypothèse que l'objet en question est visible pour cet agent et que ce dernier a désormais connaissance de sa position.

L'utilisation d'une capacité de prise de perspective est envisageable dans notre contexte applicatif du fait de la capacité de notre base de faits dynamiques à pouvoir modéliser les états mentaux du robot et de l'utilisateur de façon indépendante [5] mais aussi de les maintenir au cours du temps à partir des captations visuelles du robot et du dialogue. Du fait de leur indépendance logique, les modèles de l'utilisateur de l'utilisateur et du robot peuvent à tout moment diverger, ce serait notamment le cas dans une situation de fausse croyance dans lequel l'utilisateur ne serait pas témoin d'un changement intervenu dans l'environnement physique. Un exemple concret d'une telle situation est donné dans la figure 6.5.

Soit $P(O)$ la propriété de position d'un objet O. Dans un premier temps, le robot détecte que l'utilisateur a dans son champ de vision un livre rouge (RED_BOOK) qui est sur la table de chevet (bedroom_bedsidetable, BT). Il met alors la propriété $P(RED\_BOOK) = BT$ dans le modèle mental utilisateur. Puis, alors que l'utilisateur s'est éloigné (et l'objet n'étant plus dans son champ visuel), ce livre est échangé avec un livre marron (BROWN_BOOK) qui était jusqu'alors sur la table de cuisine (kit-

5. Bien que dans notre étude nous nous concentrons sur des situations d'interaction ne faisant intervenir qu'un robot et un utilisateur, il est à noter qu'il tout à fait possible en l'état de calculer des modèles dans un scénario impliquant plusieurs utilisateurs et/ou robots.

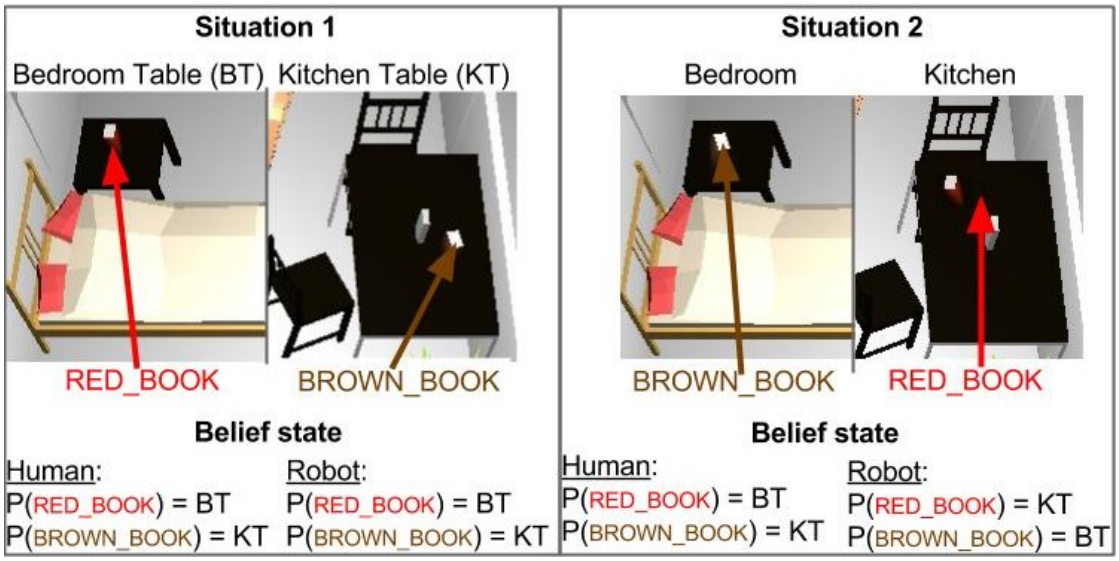


FIGURE 6.5 – *Exemple de croyance factuelle divergente.*

chen_table, *KT*), le robot étant témoin de ce changement. Dans ce contexte le robot mettra à jour son propre modèle avec les nouvelles positions des 2 livres mais laissera celui de l'utilisateur intact. En effet, sans information supplémentaire sur la situation courante, l'utilisateur croit toujours que l'objet RED_BOOK est à sa position d'origine et ce jusqu'à ce qu'il constate par lui-même l'inexactitude de cette information, par exemple si lors d'un déplacement il aperçoit que l'objet RED_BOOK est maintenant sur *KT* ou qu'il n'est plus sur *BT*. Dans ce dernier cas, la valeur de la propriété position de l'objet prendrait alors la valeur *unknown* dans le modèle utilisateur pour permettre au robot d'identifier une situation où il serait bon de lui fournir la valeur de position de l'objet.

Dans cette étude préliminaire nous nous concentrons exclusivement sur les fausses croyances relatives aux positions des différents objets dans l'environnement. Il est à noter qu'un raisonnement similaire pourrait être adopté pour d'autres propriétés telles que l'identité du dernier agent ayant manipulé un objet, son contenu ou encore sa température. Cependant, des études plus poussées seront nécessaires pour faire face à des situations de divergence plus complexes. On pourra donner comme exemple celles liées aux connaissances des agents (nom d'une personne, etc.) ou encore celles obtenues dans une configuration où le robot pourrait être lui-même dans une situation de fausse croyance.

6.2.3 Restitution multimodale des actions du système

Représentés en jaune sur la figure 6.3, quatre modules sont actuellement responsables des sorties du système.

Comme nous avons pu le voir dans la section 2.1.2, le module de fission a en charge

le processus de traduction des décisions abstraites (actes de dialogue haut niveau) du système vers des actions verbales et non-verbales (déplacement, prise de position). Pour l'instant, ce module est basé sur la définition d'un ensemble de règles prenant en compte la nature de la décision du système et le contexte courant (base de faits des différents agents). La solution retenue considère également les flux de sorties comme parallèle (pas de synchronisation fine entre gestes et paroles).

Pour la restitution vocale du système, deux modules interviennent, NLG et TTS. Le premier s'appuie sur des patrons lexicaux similaires à ceux présentés dans section 2.1.1 (voir tableau 2.1.1), le second module a quant à lui été spécialement implémenté par notre partenaire ACAPELA Group dans le cadre du projet *MaRDi*. Son originalité réside dans le fait qu'il repose sur des mécanismes d'interpolations de modèles pour élargir la richesse expressive de la voix employée tout en offrant un contrôle continu pour moduler dynamiquement la voix au cours de la synthèse d'un même énoncé (Astrinaki et al., 2012). Dans sa version actuelle nous pouvons donc jouer sur trois paramètres simultanément, à savoir le style de voix (portée, chuchotée ou normale), l'émotion transmise (ton joyeux, triste ou normal) et la vitesse d'élocution (rapide, lente, normale).

Selon la nature de l'acte de dialogue sélectionné par le système et le contexte interactif, le module de fission va donc attribuer une étiquette sur l'acte vocal pour que le module TTS puisse faire une synthèse expressive de la phrase générée par le NLG. Par exemple, si l'utilisateur et le robot ne sont pas dans la même pièce le module de fission va attribuer l'étiquette indiquant qu'il va falloir que le robot parle plus fort (avec une voix portée), ou encore si le système informe l'utilisateur qu'il ne peut pas réaliser l'action (par exemple si l'objet est hors de porté pour lui) alors il pourra faire jouer une synthèse vocale employant une voix triste.

En ce qui concerne la gestuelle et les actions physiques du robot, elles vont se faire grâce à l'utilisation d'une interface abstraite, NVBP/MC pour *Non-Verbal Behaviour Planner and Motor Control* en anglais. De par son haut niveau d'abstraction, cette dernière nous permet de faire tourner le système de façon similaire que ce soit sur la véritable plateforme robotique ou sur l'outil de simulation 3D décrit dans la section 6.3.2. Dans notre scénario, deux situations distinctes vont impliquer des mouvements de la part du robot. La première est liée à l'exécution de la commande de déplacement d'objet utilisateur, cette dernière intervient toujours en fin d'interaction car l'exécution d'une commande erronée est également synonyme d'échec dans notre scénario. La seconde situation consiste en l'exploration de l'environnement. Elle est utilisée pour acquérir des faits symboliques sur des zones non explorées (par exemple aller voir ce qu'il y a sur la table de la cuisine).

Le module de fission utilise l'interface abstraite pour transmettre les commandes haut niveau, par exemple *move(BLACT_TAPE, kitchen_table,bedroom_bedsidetable)* ou *explore(kitchen_table)*. Dans le cas où la plateforme robotique est employée, ces buts vont être transmis à un superviseur qui va dans un premier temps planifier les actions devant être exécutées par l'intermédiaire d'*HATP* (pour *Human Aware Task Planner*) (Alami et al., 2006), puis procéder à leur exécution d'après le plan ainsi établi. En simulation, l'exécution de ces commandes haut niveau est grandement simplifiée. En effet, elles

sont traduites en séquence d'actions élémentaires selon des patrons prédéfinis dont nous donnerons quelques exemples dans la section 6.3.2.

6.2.4 Gestion de l'interaction

Tout comme pour le système de dialogue *TownInfo*, le DM employé dans notre étude repose sur le paradigme POMDP HIS (voir section 3.4.4). Mais contrairement au premier système étudié dans le chapitre 4, la tâche *MaRDi* ne peut pas être directement assimilable à un problème de recherche d'information standard, il a fallu donc légèrement adapter le paradigme à notre contexte applicatif.

Le but utilisateur consiste ici en une commande de manipulation d'objet que ce dernier souhaite faire exécuter au robot parmi celles réalisables compte tenu des contraintes données par l'utilisateur et du contexte physique de l'interaction. L'ontologie de la tâche est décrite dans le tableau 6.7 (plus de détails sont disponibles dans l'annexe C.1).

Du fait de la nature dynamique des informations contextuelles considérées (base de connaissances dynamiques), la base de données métier n'est plus seulement limitée à des informations statiques comme c'était le cas pour *TownInfo* où les données métier correspondent à une liste d'établissements qui n'a pas vocation à changer durant l'interaction. De fait, la mise à jour de l'état de croyance du système de dialogue s'effectuera à la fois en prenant en compte les actes du dialogue robot et utilisateur, mais également en y intégrant l'information issue de la base des connaissances dynamiques.

```
task -> execute(cmd){1.0} ;
cmd -> manipaction(action, object){1.0} ;
action -> give(){0.5} ;
action -> move(location){0.5} ;
object -> domestic(idobj, type, color, location){1.0} ;
type -> book(title, genre, author){0.3} ;
type -> mug(){0.3} ;
type -> tape(title, genre, director){0.3} ;
type -> box(){0.1} ;
idobj = ("BLUE_BOOK" | "RED_BOOK" | ...)
color = ( blue | red | ...)
location = ( livingroom_coffeetable | livingroom_bedsidetable | ...)
book.title = ( "the lord of the rings 1" | ...)
tape.title = ("very bad trip" | ...)
author = ("J.R.R Tolkien" | ...)
director = ("Todd Phillips" | ...)
genre = ("scifi" | ...)
```

TABLE 6.7 – *Ontologie de la tâche MaRDi.*

Pour mener à bien la tâche *MaRDi* et faciliter la génération de comportement multimodaux il a fallu définir deux nouvelles actions résumées venant compléter le jeu initialement proposé dans (Young et al., 2010). Nous avons donc complété l'ensemble d'actions décrit dans la section 3.4.4 (voir tableau 3.5) par les actions *Explore* et *Execute*.

La première est employée pour procéder à la découverte de l'environnement afin d'acquérir de nouvelles connaissances factuelles. Par exemple, si le robot ne s'est jamais rendu dans la cuisine, une telle action peut être prise pour s'y déplacer et compléter ou mettre à jour ses connaissances sur les objets présents. La seconde action est quant à elle employée pour lancer la procédure d'exécution (si réalisable) de la commande « candidate » la plus probable du point de vue du robot. Elle suppose donc un effet de bord (réalisation de la commande avec toutes ses implications), ce qui typiquement n'existe pas dans des tâches purement recherche d'informations telles que *TownInfo*.

Une des limites du paradigme HIS pour le problème qui nous concerne est qu'il n'offre dans sa version initiale que des mécanismes capables de gérer l'incertitude due aux bruits présents dans le canal de communication (reconnaissance puis compréhension de la parole). Or, nous pensons qu'une autre source possible de l'incertitude peut provenir de situations de fausses croyances où la croyance en des faits erronées (par exemple une position antérieure d'un objet) viendrait bruiter les actes de communicatifs de l'utilisateur. En effet, si l'état mental de l'utilisateur n'est pas modélisé ni pris en compte, seul des mécanismes de résolution classiques de l'incertitude peuvent être appliqués par la politique, par exemple demander à l'utilisateur de confirmer des hypothèses jusqu'à ce que sa demande corresponde à la réalité observée, et ce même dans des situations où il aurait été possible d'identifier une telle situation en amont de par sa modélisation.

Gestion de l'interaction divergente pour répondre à la problématique soulevée ci-dessus nous proposons d'étendre le paradigme HIS pour y intégrer l'analyse sur les croyances factuelles du robot et de l'utilisateur directement dans le mécanisme de prise de décision du système pour pouvoir améliorer la qualité et l'efficacité du dialogue dans des situations où l'utilisateur poursuit son objectif selon des connaissances erronées qui ont pu être identifiées en amont par le système. Ainsi, nous proposons d'augmenter l'état résumé du dialogue avec un état sur la croyance divergente, nommé le *d-status*. Ce dernier est utilisé pour notifier de la présence d'une situation de fausse croyance (divergence) lors de la mise correspondance du but tel que décrit par la partition la plus probable aux connaissances dynamiques du robot et de l'utilisateur.

Dans notre cadre expérimental actuel, ces croyances sont considérées comme parties intégrantes de la ressource de connaissances dynamiques employée et sont donc maintenues de façon indépendante aux mécanismes de gestion de l'état interne du système de dialogue. Nous proposons également dans cette extension d'ajouter une nouvelle action dédiée à la résolution des situations de fausses croyances, *InformDivergentBelief*.

Les modifications apportées au modèle initial sont illustrées par les éléments en orange sur la figure 6.6. Cette dernière fait référence à l'exemple présenté plus haut et dans lequel deux livres ont été interchangés (voir figure 6.5). Soit la situation suivante : l'utilisateur vient de prononcer la phrase « donne moi le livre qui est sur ma table de chevet ». Comme le montre la figure 6.6, logiquement la partition de plus grande probabilité devient celle qui modélise le but utilisateur qui consiste à vouloir lui « apporter un livre situé sur la table de chevet ». Du point de vue du robot (ROBOT FACTS sur

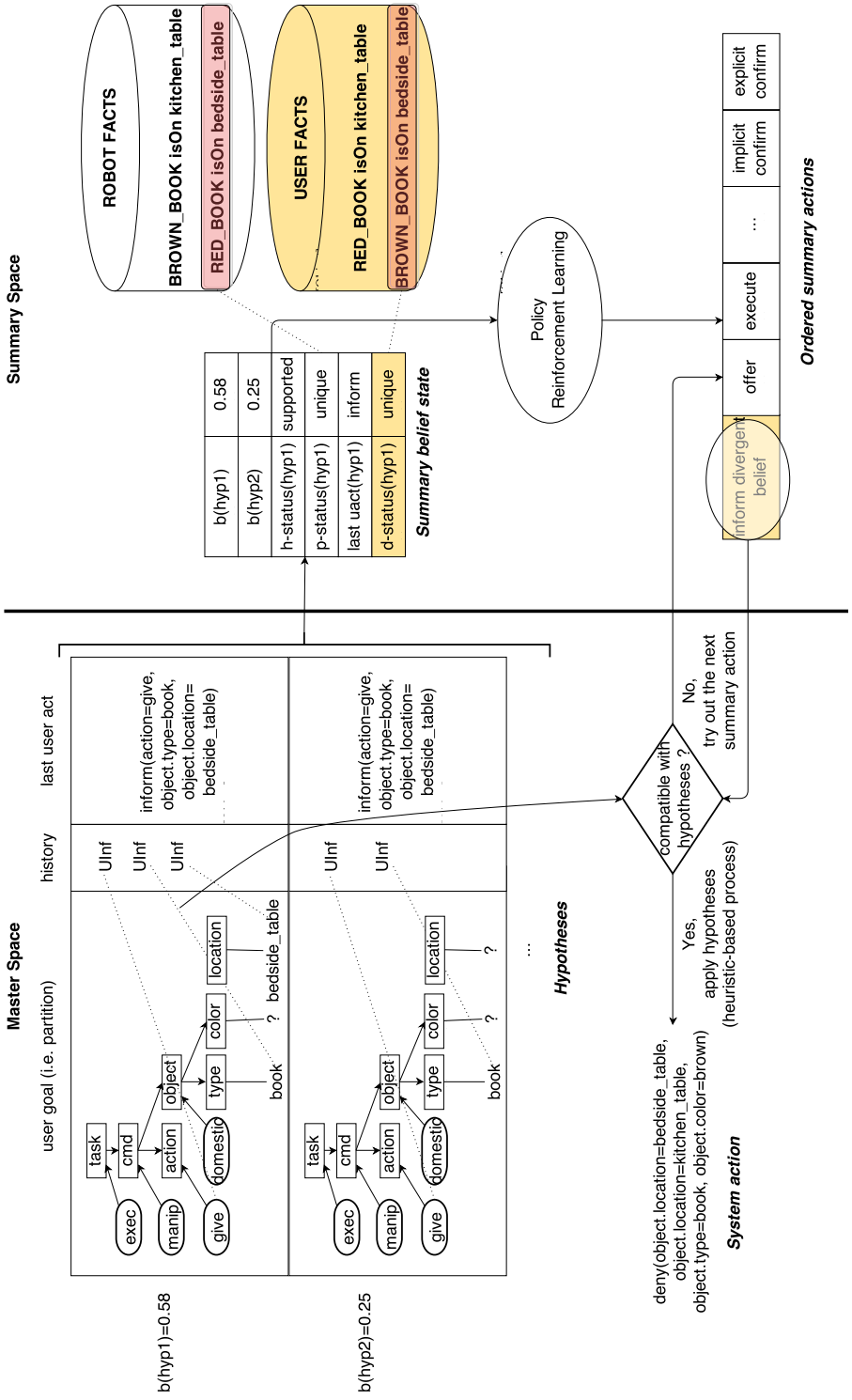


FIGURE 6.6 – *Vue d'ensemble de l'extension HIS pour prendre en compte les croyances factuelles divergentes.*

la figure) ce livre est identifié de façon unique comme étant l'objet RED_BOOK, de ce fait le *p-status* a pour valeur *unique* (voir le tableau 3.3 dans la section 3.4.4). Cependant, du point de vue de l'utilisateur (USER FACTS sur la figure), il est également identifié de façon unique comme étant l'objet BROWN_BOOK. Cette situation est considérée comme divergente et le *d-status* est réglé sur *unique* parce qu'il n'y a qu'un seul objet possible qui correspond à cette description dans le modèle de l'utilisateur et que ce dernier est différent de celui également identifié dans la base de faits dynamiques du robot. Dans nos travaux, le *d-status* ne peut prendre que les deux valeurs *unique* et *other*, nous considérons cependant cet élément comme étant catégoriel et non binaire car nous comptons étendre à terme le nombre des valeurs ainsi considérées (identification de différentes situations comme par exemple le fait que plusieurs objets candidats présentent une position divergente).

En ce qui concerne la nouvelle action résumée *InformDivergentBelief* introduite pour résoudre la situation de divergence, les actes de dialogue qui lui sont associés dans l'espace maître sont déterminés par l'intermédiaire d'heuristiques expertes. Dans cette première version, lorsqu'un cas de divergence est détecté dans la meilleure hypothèse d'état de dialogue, idéalement l'action prise par le système doit permettre d'informer l'utilisateur de manière explicite de la présence et de la nature de cette divergence. Pour ce faire, un acte de dialogue de type *deny* est employé pour informer l'utilisateur sur l'existence d'une divergence quant à la valeur de la position de l'objet « sujet » de la commande. Grâce à l'émission de cet acte de dialogue l'utilisateur va pouvoir mettre à jours ses croyances avant de poursuivre son objectif initial. Ainsi, lorsque le système informera l'utilisateur oralement de la véritable position de l'objet en question, le modèle de croyance utilisateur sera mis à jour en fonction. Ce processus est également illustré dans la figure 6.6 lorsque l'action *InformDivergentBelief* est sélectionnée en tant que prochaine action du système (action maître). L'action finalement exécutée dans l'espace maître sera : *deny(object.location=bedside_table, object.location=kitchen_table, object.type=book, object.color=brown)*, qui sera transformée par le NLG dans l'énoncé système suivant « le livre marron n'est plus sur la table de chevet mais sur la table de la cuisine ».

Pour ce qui est de la politique d'interaction, nous envisageons là encore de réaliser un apprentissage RL en ligne de la politique. Cependant contrairement aux expériences réalisées dans le chapitre 4, nous considérons ici le recours à des interactions faisant intervenir de vrais utilisateurs (que ce soit pour l'apprentissage et les tests). Nous décrivons plus précisément les conditions expérimentales dans la section suivante.

6.3 Conditions d'apprentissage et de tests en ligne de la politique d'interaction

Dans nos travaux, nous nous intéressons tout particulièrement à réaliser une évaluation du système *MaRDi* face à des utilisateurs réels, et ce dès la phase initiale de développement, que se soit pour l'apprentissage par RL en ligne de la politique de l'interaction, mais aussi pour son l'évaluation de ses performances ensuite (phase de test).

L'environnement 3D simulé est utilisé pour ces expériences afin de réduire leur coût et une évaluation complète sur le robot réel est planifiée.

6.3.1 Scénarios d'interaction

Comme mentionné plus haut, les expériences réalisées dans le cadre de *MaRDi* ont toutes été effectuées en faisant interagir le système avec de véritables utilisateurs. Pour cela, il fallu au début de chaque dialogue, fournir à l'utilisateur un but précis (une commande avec des arguments dans notre cas) dont la définition nécessite la prise en compte du contexte courant de l'interaction (positions effectives des différents objets dans la scène, la position du robot, la position de l'humain, etc.) afin de proposer des commandes pertinentes et réalisables par le système.

FIGURE 6.7 – *Exemple de but pour la tâche MaRDi tel que donné à l'utilisateur avant le déroulement de l'interaction.*

La capture d'écran reportée dans la figure 6.7 donne un exemple d'un but tel que proposé à l'utilisateur avant le début du dialogue. Ce but peut se traduire par le fait de « demander au robot d'apporter le mug bleu qui est sur la table basse du salon et s'assurer que l'objet dont le robot fait référence et/ou manipule est celui qui a pour identifiant BLUE_MUG ». L'identifiant des objets est ici employé pour limiter les ambiguïtés liées à une mauvaise calibration de l'expressivité du robot dans la phase de développement préliminaire du système.

La manière dont sont définis ces buts nous permet également de régler la difficulté des scénarios proposés. Ceci sera notamment exploité dans les expériences relatives aux traitements de situations de fausses croyances (voir la section 6.4.3). Dans ce cas de figure, de telles situations seront créées artificiellement grâce à l'introduction d'une valeur intentionnellement erronée pour le concept *object.location* (position de l'objet sujet de la commande) dans la définition du but donné à l'utilisateur. Cette corruption du but permet de reproduire une situation proche de celle décrite pour le test de Sally et Anne, puisqu'on suppose qu'un contexte interactif précédent a conduit l'utilisateur à penser que l'objet est à la position A alors qu'il est désormais en B. L'ajout dans le but permet à l'utilisateur de faire librement l'usage de la position A dans ses actes communicatifs s'il souhaite faire effectuer un commande au robot impliquant l'objet en question.

Il est également possible de faire usage d'instructions supplémentaires pour imposer à l'utilisateur de manifester des traits comportementaux particuliers. Ce sera

par exemple le cas dans l'expérience sur l'adaptation au profil utilisateur dans la section 6.4.2.

6.3.2 Simulation de l'environnement

Bien que la réalisation d'expériences sur la véritable plateforme robotique constitue un des objectifs finaux du projet *MaRDi*, compte tenu du coût élevé que représente la mise en place de telles interactions mais aussi afin de faciliter la réalisation des expériences sur les divers sites partenaires, nous avons adopté une solution dans laquelle le système des dialogue multimodal est couplé avec un logiciel de simulation robotique 3D [6].

Choix du simulateur robotique

Le recours à des logiciels de simulation est souvent nécessaire en robotique. En utilisant des simulateurs, il est possible d'évaluer et de valider certaines propositions et développements avant toute tentative d'intégration coûteuse et risquée sur la véritable plateforme robotique. Ainsi, de nombreux simulateurs sont disponibles. Nous pouvons citer par exemple la suite *Gazebo* (Koenig et Howard, 2004), la plate-forme de simulation intégrée *OpenHRP* (Nakaoka et al., 2007) ou encore la simulateur commercial *V-REP* (Freese et al., 2010). Cependant, peu de solutions sont véritablement adaptées pour l'HRI par exemple en limitant l'intégration de l'homme (contrôle limité) dans l'environnement virtuel. Ceci explique en partie pourquoi les études HRI faisant l'usage de la simulation ont longtemps été menées de façon télé-opérées, où seuls le robot et l'environnement sont modélisés dans l'environnement virtuel. Les simulateurs robotiques *USARSim* (Lewis et al., 2007) et *MORSE* [7] (Echeverria et al., 2011, 2012) sont tous deux utilisés dans plusieurs dizaines de travaux en HRI en raison de leur support explicite de l'homme. Cependant, la seconde solution dispose de nombreux avantages pour notre contexte d'étude qui ont motivé sa sélection dans nos travaux.

MORSE la plateforme de simulation open-source *MORSE* est une solution de simulation robotique libre qui a pour contributeur principal le LAAS-CNRS. Cet outil présente l'avantage de pouvoir prendre en charge de nombreux *middleware* (comme ROS , YARP [8]) mais aussi de permettre au robot virtuel d'interagir avec son environnement virtuel grâce à des capteurs/actionneurs modélisés de façon réaliste pour faciliter le déploiement ultérieur vers une véritable plateforme robotique. Cet outil propose également des capteurs/actionneurs de haut niveau pour alléger la tâche du développeur en lui simplifiant la mise en place de certaines chaînes de traitements pour qu'il puisse

6. Il est important de signifier que le terme simulation employé ici ne se réfère pas à la problématique de la « simulation d'utilisateurs » introduite dans la section **??**. En effet, l'objectif visé par l'outil considéré est de reproduire le plus fidèlement possible l'environnement dans lequel va se dérouler l'interaction (de même que le robot et l'utilisateur) et non les comportements d'un utilisateur face aux réponses du système.
7. https ://www.openrobots.org/wiki/morse/
8. http ://wiki.icub.org/yarp/

se concentrer sur des problématiques plus pertinentes. Par exemple, *MORSE* dispose à la fois d'un capteur de type caméra RGB-D et d'une caméra sémantique, alors que le premier capteur fournit une image non traitée en sortie (pixels), la seconde extrait directement le nom et la position des objets apparaissant dans son champ de vision.

La solution *MORSE* repose actuellement sur le *Blender Game Engine* intégré au logiciel libre Blender [9]. Ce faisant, elle propose un rendu graphique 3D avancé (shaders OpenGL) intégrant des principes de physique tel que la gravité par l'intermédiaire du moteur *Bullet Physics* pour offrir une représentation réaliste de l'environnement dans lequel se déroule l'interaction. Dans notre contexte, une réplique virtuelle de l'appartement de trois pièces dans lequel devraient s'effectuer les tests finaux du projet a été mise à notre disposition ainsi que celle du PR2 qui elle fait partie des plateformes robotiques pré-intégrées dans *MORSE*.

Un contrôle immersif d'un avatar humain (voir la figure 6.8) est également mis à disposition. Ce dernier fait usage soit d'une interface classique clavier/souris mais également d'une interface ASUS Xtion/Wiimote. Quel que soit son choix, l'utilisateur au travers de son avatar peut se déplacer [10], observer et interagir à les éléments de l'environnement. Par exemple, il peut saisir et relâcher un objet si ce dernier est manipulable et à sa portée.

Simulation pour la tâche *MaRDi*

Avant toute interaction, un script est chargé de positionner les différents objets dans l'environnement sur les positions d'intérêt (comme *livingroom_coffeetable* ou *kitchen_table*). Il est en serait de même dans des conditions réelles (hors scénarios prédéfinis) sauf que ce serait à l'expérimentateur de préparer l'environnement. Au fil des interactions, et selon la configuration choisie, ces objets vont progressivement changer de position du fait de l'exécution des commandes par le robot. C'est pourquoi un suivi est également fait sur l'environnement courant. Il permet de générer de façon automatique des buts pertinents (ne pas proposer de déplacer un objet à une position qu'il occupe déjà).

Pour le PR2, il peut être démarré soit avec un ensemble de faits symboliques préchargés, soit dans une configuration plus réaliste, où il doit déterminer ces faits de manière autonome grâce à ses capacités de raisonnement spatial.

Pour pouvoir réaliser les actions non-verbales de façon transparente entre la véritable plateforme robotique et le simulateur nous avons choisi de définir la liste d'actions abstraites qui suit :

— **GoToLocation** : déplace le robot vers une position d'intérêt passée en argument (on parlera également de zones de manipulations). Dans notre configuration expérimentale ces zones correspondent aux valeurs des concepts de haut niveau *object.position* et *move.position* dans l'ontologie du domaine *MaRDi* utilisée pour

9. https ://www.blender.org/

10. Dans notre contexte expérimental cette fonctionnalité ne sera pas exploitée car l'utilisateur est supposé être handicapé.

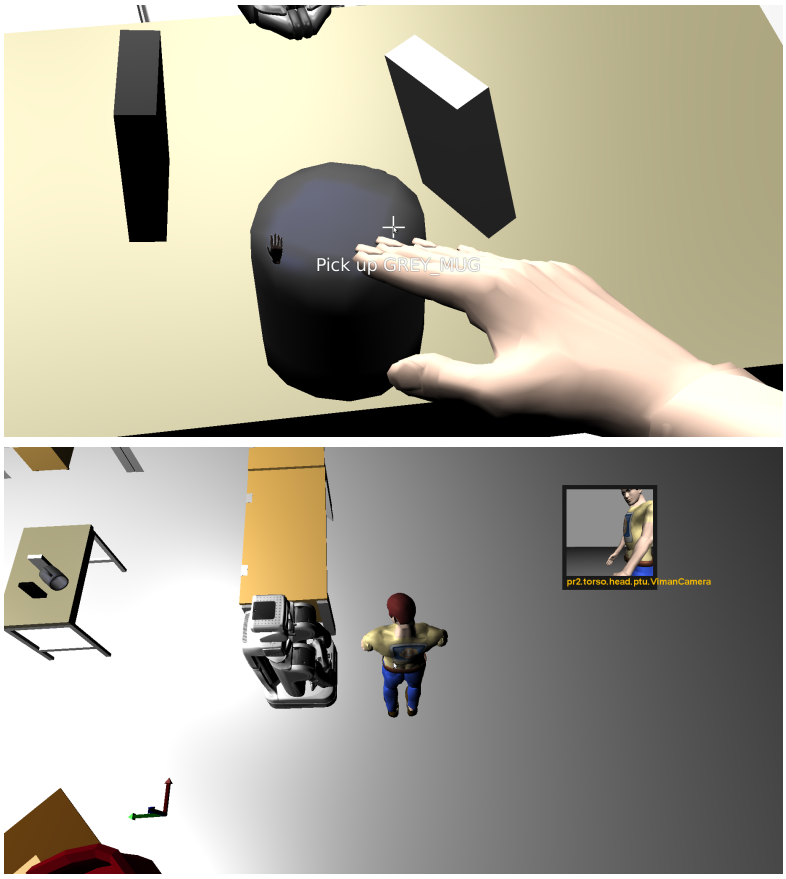


FIGURE 6.8 – *Avatar humain dans le simulateur MORSE en première et troisième personne (resp. en haut et en bas)*

le dialogue. Dans *MORSE* nous utilisons l'actionneur *teleport* qui permet de déplacer instantanément le robot à une position spécifiée, l'objectif étant d'accélérer la durée de l'interaction pour faciliter la phase de collecte. Il est à noter que la plupart des déplacements seront simplifiés en simulation, seule une partie de la motricité fine (ici le déplacement des armatures du robot) est à ce jour réalisée à l'identique sur le robot et sur sa version simulée ;

— **HeadScan** : parcourt visuellement l'environnement grâce à un mouvement de tête circulaire du robot. En simulation, le robot dispose d'une caméra sémantique sur la tête qui lui permet de reconnaître les objets dès qu'il les a en visuel. Sur le PR2, pour simplifier leur détection, les objets sont identifiés par un code QR scotché dessus ;

— **GraspObject** : attrape un objet passé en argument si ce dernier est à portée. En simulation, le service *grasp* est employé. Ce dernier attache automatiquement l'objet (si à portée) à la main gauche du robot ;

— **DropOnLocation** : relâche l'objet que le robot a dans sa main sur la position d'intérêt passée en argument de la commande ;

— **GiveToHuman** : déplace le robot auprès de l'homme avant de tendre le bras vers

lui tout en abaissant la raideur de sa main pour que l'humain puisse se saisir de l'objet qui est à l'intérieur. Pour cela les actionneurs de l'armature liée au bras droit du robot sont utilisés.

De nombreux travaux dans le domaine de l'HRI située utilisent des scénarios d'interaction que l'on pourrait modéliser (si ce n'est déjà le cas) par l'intermédiaire d'un simulateur robotique 3D tel que celui adopté dans nos expériences (Byron et Fosler-Lussier, 2006; Stiefelhagen et al., 2007; Lucignano et al., 2013). Néanmoins, très peu de travaux privilégient à ce jour la simulation pour réaliser un apprentissage RL en ligne de la politique d'interaction qu'ils emploient (même pour un *bootstrap*). En effet, la plupart des travaux en HRI recourent pour cela à des expériences préliminaires en configuration de WoZ (où un utilisateur expert contrôle le robot à distance afin de jouer son rôle). On peut par exemple faire référence aux études réalisées dans (Prommer et al., 2006; Stiefelhagen et al., 2007; Rieser et Lemon, 2008). Cependant, comme nous l'avons déjà mentionné dans les précédents chapitres, ce type d'approche est coûteux lors de sa réalisation (temps, recrutements de sujets) et pose la question de « quels comportements l'expert doit-il faire jouer au robot pour garantir un bon apprentissage ». C'est pourquoi nous proposons directement d'apprendre et de tester les actions possibles en ligne mais dans une configuration virtuelle où les actions prises par le système auront des incidences et des temps d'exécution moindres.

Alternative fonctionnelle au simulateur robotique durant la conduite de nos expériences nous avons constaté que beaucoup de temps avait été perdu lors du lancement et du déroulement des interactions sur le simulateur, et ce malgré les simplifications apportées à ce dernier par rapport à la véritable plateforme robotique. En moyenne, il a été estimé que chaque interaction sur le simulateur prenait une durée allant de 7 à 10 minutes (initialisation de l'environnement, détections d'objets, mouvements du robot, déplacements, etc.).

Nous avons donc cherché une solution de contournement, sans perte en termes de fonctionnalité et de performance, pour une nouvelle fois accélérer l'interaction. L'approche retenue est de recourir à une représentation « fixe » de la scène (une capture d'écran du point de vue de l'humain) dans une interface web multimodale où les objets manipulables et les positions d'intérêts (meubles) sont considérés comme cliquables pour pouvoir simuler les gestes déictiques en provenance de l'utilisateur. Dans cette configuration, la base des faits dynamiques est alimentée par un module dédié capable de générer les faits dynamiques qui auraient été produits par SPARK dans la configuration virtuelle standard tout au long de l'interaction dans le contexte d'interaction chargé au démarrage. Pour proposer un suivi de l'interaction par l'utilisateur lors du dialogue, l'affichage web (capture d'écran de la scène) est mis à jour quand une modification physique de l'environnement intervient (déplacement physique du robot et/ou d'un objet) pour refléter la réalité observable.

6.3.3 Retours utilisateur et critères d'évaluations

Pour réaliser nos expériences avec de vrais utilisateurs, des conditions d'interaction en laboratoire seront envisagées. Bien que ces dernière ne soient pas à proprement parler « optimales », puisque l'on peut légitimement se demander jusqu'à quel point les utilisateurs qui y participent se comportent de façon naturelle lorsqu'ils jouent le rôle qui leur a été attribué par les consignes, elles constituent néanmoins un moyen simple et standard de contrôler précisément les expériences.

Les sujets recrutés sont des utilisateurs ayant des connaissances dans le domaine du NLP (membres du personnel du LIA essentiellement). Nous prendrons le soin de les distinguer dans nos expérience des utilisateurs dits *experts*, plus familiers des problématiques du dialogue et du RL et qui ont de fait la possibilité d'exploiter leurs connaissances pour interagir de manière plus efficace avec le système (membres du groupe Interactions Vocales du LIA).

Les conditions de laboratoire nous facilitent l'accès au critère de la réussite de la tâche qui constitue une information essentielle pour l'apprentissage RL par rapport à un système déployé (Gašić et Young, 2011). En effet, les utilisateurs réels prennent rarement le temps de remplir un formulaire de satisfaction à moins que cela soit dans leur intérêt immédiat ou à court terme (accès à des promotions, amélioration explicite de la qualité de service, etc.). Ainsi, un questionnaire inspiré de celui de employé dans (Walker, 2000) a été employé dans notre étude à la fin de chaque dialogue (voir tableau 6.8).

Critères évalués	**Questions associées**
Réussite de l'interaction $^{(*)}$	Le robot a-t-il exécuté une commande adéquate ?
Performance génération	Les réponses du système étaient-elles facilement compréhensibles ?
Performance compréhension	Le système comprenait-il ce que vous disiez ?
Facilité de la tâche	'Etait-il facile de d'atteindre les objectifs de votre scénario ?
Rythme de l'interaction	La cadence de l'interaction vous a-t-elle paru appropriée ?
Expertise de l'utilisateur	Saviez-vous quoi dire à chaque étape de l'interaction ?
Temps de réponse système	Avez-vous ressenti des latences dans les réponses du système ?
Comportement du système	Le système s'est-il comporté comme vous l'imaginiez ?
Naturel	À quel point jugeriez-vous l'interaction naturelle ?
Utilisation future	D'après cette expérience, seriez-vous prêt à interagir de nouveau avec un tel système ?

TABLE 6.8 – *Exemple de questionnaire utilisé dans MaRDi -* $^{(*)}$ *seul champ obligatoire.*

Outre le critère binaire sur la réussite ou l'échec de la tâche, ce questionnaire évalue de multiples critères sur une échelle de satisfaction allant de 1 à 5 et offre une évaluation

implicite sur les différents composants du système de dialogue (le module de compréhension, le gestionnaire de dialogue) ainsi que sur la qualité de l'implémentation au travers de critères comme le temps de réponse ou encore l'utilisation future.

Cependant, nous avons constaté lors de ces évaluations préliminaires que l'effort nécessaire pour remplir ces formulaires a généralement découragé les utilisateurs. Ces derniers ont eu plutôt tendance au fil des interactions à ne plus répondre qu'au critère de réussite, ce dernier étant le seul obligatoire [11].

Une solution à cette problématique serait d'envisager une approche identique à celle adoptée dans le paradigme d'évaluation PARADISE (Walker et al., 1997) pour automatiser partiellement l'évaluation sur les critères subjectifs en les corrélant à des critères objectifs (réussite, nombre de tours) par une méthode de régression linéaire. Cependant cette solution nécessite le recours à beaucoup de données d'apprentissage pour obtenir un modèle linéaire correct, modèle qui est en soit discutable (Larsen, 2003). Ainsi, compte tenu du petit nombre d'utilisateurs recrutés nous nous sommes limités dans nos travaux aux critères considérés comme « objectifs » pour comparer nos différentes méthodes entre elles.

6.4 Expériences et résultats

Fort des résultats obtenus en simulation avec l'apprentissage socialement inspiré (apprentissage de zéro et adaptation) dans la section 4.3.3. Nous cherchons dans les sections 6.4.1 et 6.4.2 à les valider en pratique et sur une nouvelle tâche avec des dialogues impliquant de vrais utilisateurs.

Puis dans la section 6.4.3 nous évaluerons l'impact de la prise en compte de la perspective et plus exactement des croyances divergentes sur la conduite de l'interaction, de même que sur le processus d'apprentissage.

6.4.1 Apprentissage de zéro de la politique de dialogue

La première expérience réalisée s'attache à valider sur un cas pratique l'apport sur l'apprentissage de l'approche par renforcement socialement inspiré initialement constaté sur simulateur dans le chapitre 4.

Pour ce faire nous considérons deux configurations similaires aux méthodes **BASELINE** et **SOCIAL** employées pour la tâche *TownInfo*. Les différences avec la situation précédente résident dans la prise en compte d'un jeu d'action résumée plus étendue (deux actions résumées supplémentaires : *Explore* et *Execute*) ; et pour le cas particulier de **SOCIAL**, le fait que ψ_{social} soit cette fois déterminée par des jugements subjectifs émis par de vrais utilisateurs.

11. Il a été laissé au libre arbitre des utilisateurs la possibilité de compléter ou non le formulaire de satisfaction dans son intégralité.

Ainsi, dans la configuration retenue, l'utilisateur peut émettre lorsque c'est son tour un jugement subjectif sur l'évolution du cours de l'interaction au travers d'une barre de notation à cinq niveaux associés à ceux de l'échelle de Likert employée pour déterminer la valeur de ψ_{social}. Cette barre de notation est accessible sur l'interface graphique de la plateforme de dialogue (interface web). Comme dans le cadre expérimental mis en œuvre pour la tâche *TownInfo*, le coefficient τ a été fixé à 4. De plus, la même définition de la fonction de récompense immédiate R_{env} est ici employée ($+20$ si réussite, 0 si échec, -1 par tour).

Il est à noter que dans ce premier jeu d'expériences l'extension du cadre formel HIS au cas de la gestion des fausses croyances n'est pas utilisé (pas *d*-status).

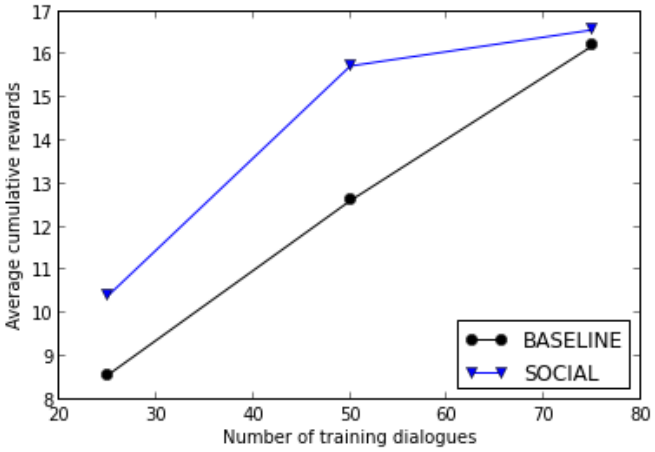


FIGURE 6.9 – *Résultats des configurations de référence et sociale de l'algorithme KTD-Q sur la tâche MaRDi (apprentissage).*

Les résultats obtenus sont présentés dans la figure 6.9 en termes de récompenses (R_{env}) cumulées recueillies durant la phase d'apprentissage de la politique. Pour ces courbes, chaque point est une moyenne sur la performance obtenue lors de l' apprentissage à l'aide d'une fenêtre glissante d'une largeur de 50 et d'un pas 25 dialogues. Ici 100 dialogues d'apprentissage sont considérés pour chacune des méthodes (**BASELINE** et **SOCIAL**). Pour favoriser la comparaison entre les deux courbes d'apprentissage, le même utilisateur expert a effectué les deux apprentissages de la politique suivant la même séquence de buts. À titre indicatif, le WER et le CER ont été estimés à environ 10% après annotation manuelle des 50 premiers tours de dialogue collectés.

Comme le montre la figure 6.9, les performances de **SOCIAL** surpassent celles obtenues avec la technique **BASELINE** sur l'intégralité du processus d'apprentissage. Néanmoins, l'écart est nettement réduit à la fin de la phase d'apprentissage. Ces observations vont dans le même sens que les résultats obtenus lors des expériences sur simulateur sur la tâche *TownInfo*. Il est à noter que le haut niveau de performance atteint par ces deux méthodes au cours de l'apprentissage s'explique principalement par le fait que les contraintes pour définir la commande à exécuter sont moins nombreuses que les contraintes possibles dans *TownInfo*, et donc les dialogues sont généralement

moins ambiguës et donc plus courts.

La politique obtenue en prenant en compte les récompenses sociales dans l'apprentissage a été par la suite testée par 6 nouveaux sujets distincts sur 75 dialogues au total. Dans ce contexte, une moyenne de 16, 1 a été obtenue en terme de récompenses cumulées en contexte de tests (pas d'exploration ni de modification en ligne de la politique de contrôle). Cette fois le WER a été estimé sur 50 tours de dialogue à 10% et le CER à 15%.

Les résultats obtenus dans ces conditions font état d'un niveau de performance légèrement inférieur à celui observé en fin d'apprentissage. Ce constat trouve son explication principale dans le fait que les utilisateurs qui ont participé à cette expérience n'étaient pas informés des capacités effectives de compréhension du système ni des quelques astuces utiles pour débloquer certaine situations dialogiques (contrairement à l'expert ayant conduit l'apprentissage). Ces résultats montrent cependant qu'un système appris en ligne avec un nombre limité de dialogues effectués avec un utilisateur expert est ensuite tout a fait capable de faire face raisonnablement à de nouveaux utilisateurs. De plus, on constate que l'apprentissage est accéléré lorsqu'on considère les jugements subjectifs du dit expert en tant que signaux de renforcement additionnels.

6.4.2 Capacité d'adaptation de la plateforme

Pour évaluer maintenant les capacités d'adaptation de la politique d'interaction, nous considérons au préalable une politique apprise sur 100 dialogue selon l'approche **BASELINE**, puis la réalisation de 30 nouveaux dialogues avec un utilisateur présentant un autre profil pour chacune des configurations **BASELINE** et **SOCIAL** à partir de la même politique.

Comme précédemment, le même utilisateur expert a effectué l'ensemble des dialogues d'adaptation et la même séquence de buts a été utilisée pour faciliter la comparaison entre les deux courbes d'apprentissage. Afin d'assurer que l'utilisateur expert fasse usage de comportements distincts de ceux mis en œuvre dans le cadre de l'apprentissage initial il lui a été donnée comme instruction supplémentaire de ne pas faire mention de certaines propriétés (couleur, emplacement, etc.) tant que celles-ci n'avait pas été explicitement demandées par le robot. De plus, il lui a été demandé d'agir de façon impatiente (mettre fin au dialogue si le système se répète pendant plus de deux tours) et/ou obstiné (insister si le système ne répond pas à ses questions). Pour la méthode **SOCIAL**, nous avons également demandé à l'utilisateur de juger le système en conséquence tout au long de l'interaction.

La figure 6.10 présente les résultats en terme de récompenses moyennes cumulées obtenues lors de l'adaptation de la politique à de nouvelles dynamiques de l'environnement. Pour ces courbes, chaque point est une moyenne sur les performances obtenues au cours de l'apprentissage en appliquant une fenêtre glissante de largeur de 12 et d'un pas de 6 dialogues. Malgré le nombre limité d'échantillons, on peut tout de même constater que l'approche **SOCIAL** semble gérer plus efficacement le changement d'utilisateur que **BASELINE**. En effet, à la vue des courbes les capacités d'adaptation

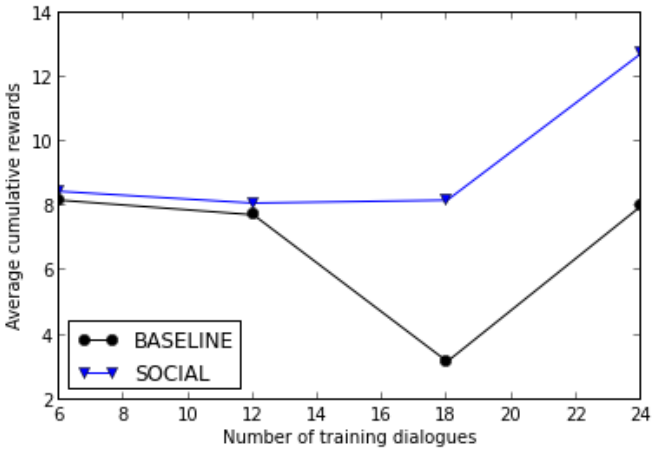


FIGURE *6.10* – *Résultats des configurations de référence et sociale de l'algorithme KTD-Q dans le contexte d'un changement de profil utilisateur sur la tâche MaRDi (apprentissage).*

offerte par l'algorithme KTD semble profiter d'une exploration guidée par les récompenses sociales pour mieux faire face aux nouveaux comportements de l'utilisateur. Par conséquent, ces résultats corroborent là encore ceux observés en simulation sur la tâche *TownInfo*.

6.4.3 La prise de perspective au service de la prise de décision

Dans cette seconde étude, notre objectif est d'évaluer l'apport de la prise en compte d'une prise de perspective conceptuelle (niveau 2) et tout particulièrement de la notion de fausse croyance dans la problématique de la gestion d'un dialogue multimodal situé.

Pour atteindre cet objectif, les scénarios d'intérêts requièrent l'existence de situations de croyances factuelles divergentes entre l'état mental de utilisateur et celui du robot. Pour trouver de telles situations en conditions réelles, il aurait été nécessaire d'avoir recours à une longue période de suivi du contexte de l'interaction (modèle du robot et de l'homme). Pour contourner ce problème, nous avons fait le choix de corrompre le scénario utilisateur, sans que ce dernier en soit informer en début d'interaction. Pour ce faire, une fausse croyance est ajoutée sur la position de l'objet sujet de la commande (uniquement lorsque ce dernier n'est dans le champ de vision direct de l'utilisateur). Bien que la situation soit obtenue de façon artificielle pour les besoins de l'expérience, nous savons d'après (Milliez et al., 2014), qu'elle peut être retrouvée grâce au module de raisonnement spatial si par exemple le robot est témoin d'une modification de la scène opérée par un tiers en l'absence du premier utilisateur.

À des fins de comparaison, nous avons évalué la capacité du robot à réaliser des tâches de manipulation d'objets à la fois dans un contexte classique (CL) ou de fausse croyance (FB). Nous considérons donc un système appris exploitant les croyances des différents agents (noté **BA-BASELINE** ci-après), une version à base de règles expertes (noté **BA-HDC**), ainsi que deux autres systèmes similaires n'exploitant pas la prise de

	HDC			BA-HDC			BASELINE			BA-BASELINE		
TASK	*Avg.R*	*T*	*SuccR*	*Avg.R*	*T*	*SuccR*	*Avg.R*	*T*	*SuccR*	*Avg.R*	*T*	*SuccR*
CL	14,33	4,81	0,85	14,28	4,86	0,86	17,62	2,95	0,93	17,69	2,88	0,93
FB	9,78	6,67	0,72	13,05	5,61	0,83	12,72	5,94	0,83	13,89	4,78	0,83
ALL	12,97	5,36	0,82	13,92	5,08	0,85	16,15	3,85	0,9	16,55	3,45	0,9

TABLE 6.9 – *Performances du système MaRDi sur les tâches classiques (CL), celles faisant intervenir de fausses croyances (FB) et toutes tâches confondues (ALL) en termes de moyennes sur les récompenses cumulées (Avg.R), de nombre de tours (T) et de taux de réussite (SuccR).*

perspective (notés respectivement **BASELINE** et **HDC**).

Les approches par règles **HDC** et **BA-HDC** ont toutes deux pour origine l'approche **HDC** décrite dans le chapitre 4. En effet, nous avons uniquement adapté les règles employées pour la tâche *TownInfo* afin d'y intégrer les deux nouvelles actions *Explore* et *Execute*. Pour le cas particulier de **BA-HDC** cette solution a également été étendue à l'usage du *d-status* et de l'action *InformDivergentBelief* afin de résoudre explicitement les situations de fausses croyances. Là encore, il ne s'agit pas de politiques expertes optimisées mais seulement de stratégies suffisamment robustes pour pouvoir gérer correctement une interaction avec des utilisateurs réels.

Il en va de même pour les configurations **BASELINE** et **BA-BASELINE**. Cependant ici, l'algorithme KTD-SARSA a été employé en lieu et en place de l'algorithme KTD-Q [12].

Pour réaliser l'apprentissage en ligne de la politique d'interaction par RL, deux utilisateurs experts ont d'abord réalisés 40 dialogues sur des tâches CL puis 20 dialogues d'adaptation pour les deux méthodes **BASELINE** et **BA-BASELINE**. Pour la première ils s'agissaient de tâches CL et de tâches FB pour la seconde.

En ce qui concerne l'évaluation, 10 dialogues dans chaque configuration du système proposée ci-dessus ont été réalisés par 6 sujets distincts (2 femmes et 4 hommes de 25 ans en moyenne) de sorte que 240 dialogues au total ont été réalisés face au système. En ce qui concerne les politiques apprises, elles ont été fixées durant les tests et ont donc agi de façon gloutonne selon la fonction de qualité. Il est à noter que 30% des dialogues effectués impliquaient une tâches FB. Aucun sujet n'a eu connaissance de la configuration employée par le système.

Le tableau 6.9 présente donc les résultats obtenus lors de tests selon le type de tâche (ALL représentant les performances toutes tâches confondues). Il s'agit donc là de moyennes faites sur les 60 dialogues effectués pour chaque méthode et métrique. Ces résultats sont tout d'abord donnés en terme de moyenne sur les récompenses cumulées (Avg.R). Selon la définition de la fonction de récompense, cette métrique exprime par une valeur réelle unique les deux variables d'amélioration, à savoir le taux de réussite et le nombre de tours jusqu'à la fin de dialogue. Toutefois, les résultats obtenus sur ces deux mesures ont également été reportés dans ce tableau pour en faciliter l'interpréta-

12. Ce choix a été fait arbitrairement à des fins de tests lors de l'élaboration de l'article correspondant (Ferreira et al., 2015), du fait du coût de reproduction de ces expériences nous n'avons malheureusement pas pu les uniformiser avec le reste du manuscrit en employant KTD-Q.

tion dans le cas du problème qui nous intéresse ici, la gestion de divergence.

Les différences observées sur le niveau de performance global (ligne ALL) entre les méthodes **BASELINE** et **BA-BASELINE** mais aussi entre **HDC** et **BA-HDC** illustrent là encore l'intérêt de considérer les méthodes de RL par rapport à des approche à base de règles expertes. En effet, comme il est possible de le constater, un apprentissage réalisé sur 60 dialogues permet de dépasser en termes de performances les deux solutions expertes. Sur les tâches CL la performance entre **BASELINE** et **BA-BASELINE** ainsi qu'entre **HDC** et **BA-HDC** sont similaires. Ainsi, l'ajout du *d-status* et de l'action *InformDivergentBelief* ne semblent pas influencer négativement la qualité de gestion du dialogue lorsque les situations pour lesquelles elles sont prévues n'apparaissent pas. Si pour **BA-HDC** ce constat est logique (en l'absence de fausses croyances, les règles sont identiques à celles utilisés pour **HDC**), pour **BA-BASELINE** une dégradation aurait pu être observée puisque la politique testée est cette fois apprise avec un degré supplémentaire de complexité (plus grand espace état / d'action que pour **BASELINE**).

Lorsqu'on fait cette fois une comparaison sur les tâche FB entre **BASELINE** et **BA-BASELINE** ou entre **HDC** et **BA-HDC**, on remarque que les résultats sont toujours en faveur des systèmes BA. En effet, les deux configurations BA ont un taux de réussite plus élevé et semblent plus efficace (gain moyen de 1 tour de dialogue par rapport à la version non-BA). Toutefois, ce constat n'a pas pu être validé statistiquement en raison d'intervalles de confiance relativement élevés sur les mesures concernées. Par exemple, l'intervalle de confiance sur le taux de réussite sur les tâches FB est proche de $0,2$ pour toutes les configurations. Ceci s'explique par le faible nombre de dialogues considérés dans chacune des configurations du système du fait des coûts expérimentaux élevés, mais aussi par les faibles espoirs de gains attendus lors de la résolution des situations de fausses croyances. En effet, dans un scénario de manipulation d'objets, cette dernière ne peut pas être considérées comme fondamentale mais plus comme un moyen de faire face à un degré supplémentaire (non dominant) de l'incertitude pour améliorer le naturel de l'interaction, et ce faisant l'expérience utilisateur.

Pour avoir une meilleure idée des différences quant à la gestion de l'interaction de ces quatre politiques, nous avons donc choisi de réaliser une étude qualitative complémentaire. Nous avons pu constater que la prise d'actes de demande validation auprès de l'utilisateur (confirmer la valeur d'une contrainte, proposer une entité, etc.) est moins fréquente pour les deux méthodes apprises. Ainsi, lorsque les politiques apprises sont confiantes quant à la meilleure hypothèse sur la commande de manipulation voulue par l'utilisateur, elles en favorisent l'exécution plutôt que d'essayer de vérifier coûte que coûte leur validité auprès de l'utilisateur comme le ferait une approche par règles expertes. Ceci peut s'apparenter à un problème visant à déterminer dynamiquement un seuil de confiance permettant de choisir l'action à exécuter ou de demander une validation de la commande la plus probable.

Dans le tableau 6.10 deux dialogues extraits des données d'évaluation illustrent les différences entre une gestion BA et non-BA sur une même tâche de FB (ici un livre rouge a été échangé avec un livre marron). Si la divergence de croyances factuelles n'est pas explicitement prise en compte, le DM peut avoir à faire face à un niveau supplémentaire

R_1 :	Comment puis-je vous aider ?		
U_1 :	Apporte moi le livre qui est sur ma table de chevet		
		R_2 :	Voulez-vous le livre rouge ?
		U_2 :	Non, le livre marron.
R_2 :	**Le livre marron n'est plus sur la table de chevet mais a été déplacé sur celle de la cuisine**	R_3 :	Il n'y a pas de livre marron dans la chambre mais il y en a un dans la cuisine
U_2 :	Ok, va me le chercher alors	U_3 :	Es-tu sur ? Ok, apporte moi celui-là.
R_3 :	Je vous apporte le livre marron qui est sur la table de la cuisine	R_4 :	Je vous apporte le livre marron qui est sur la table de la cuisine

TABLE 6.10 – *Exemples de dialogue avec (a) et sans (b) raisonnement sur la divergence de croyances factuelles dans le cas où un échange de position a été opéré à l'insu de l'utilisateur entre le livre rouge et le livre marron.*

de confusion (voir (b) à partir de R_2 à U_3). Nous pouvons aussi voir dans (b) que le système non-BA est tout de même en mesure de réussir les tâches de FB. Ceci explique notamment pourquoi la configuration **BASELINE** obtient de bonnes performances sur ce type de tâche. En effet, si l'objet est clairement identifié par l'utilisateur (par exemple avec la couleur et le type), le système peut libérer la contrainte de la fausse position donné par l'utilisateur et donc se trouver en mesure de faire une offre sur la commande « corrigée » impliquant la vraie position de l'objet, si ce n'est de démarrer son exécution.

Concernant les principales différences entre **BA-BASELINE** et **BA-HDC**, nous avons observé une utilisation moins systématique de la nouvelle action *InformDivergentBelief* dans l'approche apprise. Par exemple, **BA-BASELINE** essaie d'abord de parvenir à un haut niveau de certitude sur la présence de l'objet impliqué dans la divergence dans la formulation du but l'utilisateur. En outre, à l'instar de **BASELINE**, **BA-BASELINE** a appris d'autres mécanismes pour mener à bien les tâches de FB telles, par exemple celui de l'exécution directe de la commande lorsque l'information transmise à l'utilisateur semble être suffisante pour identifier l'objet sans ambiguïté.

6.5 Bilan

Cette partie nous a permis de décrire la plateforme de dialogue développée dans le cadre du projet *MaRDi*. Comme nous avons pu le voir il s'agit là d'une plateforme préliminaire qui repose encore sur beaucoup d'expertise. Cependant les résultats obtenus confirment qu'un apprentissage en ligne de la politique en ligne avec de vrais utilisateurs est possible et ce dans des laps de temps très courts (peu de dialogues nécessaire pour atteindre de bonnes performances). Ainsi, cette technique nous parait être une bonne alternative au WoZ pour ce qui est de collecte de données.

La méthode de renforcement social proposée dans le chapitre 4 a là encore montré son efficacité pour accélérer l'apprentissage de zéro et l'adaptation à un profil utilisateur du système. Cependant, les récompenses sociales ont été ici obtenues des utilisateurs de façon simplifiée (interface graphique), des capteurs multimodaux pourraient donc être employés à cette fin dans une configuration plus aboutie de la plateforme.

Nous avons également décrit comment le cadre formel POMDP HIS utilisé pour la gestion de l'interaction peut être couplé efficacement aux mécanismes de suivi de croyances factuelles dynamiques des différents participants. L'évaluation de la méthode proposée avec de véritables utilisateurs a confirmé que cette information supplémentaire était à même de contribuer à une réalisation plus efficace et naturelle des tâches HRI visées dans leur globalité. Cette étude de la prise en compte de l'aspect situé dans les mécanismes de décision mais aussi de la compréhension et de la génération devra être prolongée. Par exemple, le fait d'utiliser les capacités visuelles et physiques de l'utilisateur pour discriminer des hypothèses sur l'état du dialogue dans le DM. Cela pourrait notamment être obtenu en considérant des raisonnements cognitifs supplémentaires comme ceux permettant de reconnaître des situations similaires à celle où l'utilisateur dirait « passe moi le DVD » alors qu'un seul DVD serait dans son champ visuel et hors de sa portée, et dans laquelle il serait judicieux de mettre plus de confiance dans le fait que l'utilisateur fasse référence à cet objet. L'avantage qu'il y aurait alors à employer des mécanismes d'apprentissage RL tels que ceux proposés dans ce manuscrit est qu'ils pourraient être en mesure d'éviter des décisions issues de raisonnements hâtifs du fait d'erreurs dans les canaux de communication (gestion de l'incertitude). Une autre piste d'amélioration possible consisterait à tenir compte des connaissances factuelles dynamiques de l'utilisateur pour que le robot puisse déterminer les meilleurs référents spatiaux permettant d'identifier une zone ou un objet particulier dans ses réponses.

Enfin, il est important de rappeler que ces résultats ont été obtenus en simulant l'environnement physique de l'interaction. L'étape suivante sera donc d'intégrer le système de dialogue multimodal sur la véritable plateforme robotique et d'effectuer de nouvelles évaluations en situation réelle pour voir si elles corroborent une nouvelle fois les résultats présentés dans ce manuscrit. Ce travail constitue d'ailleurs l'un des objectifs des évaluations finales du projet ANR *MaRDi*, qui pour des raisons de contraintes temporelles n'ont pas pu avoir lieu avant la fin de cette thèse.

Chapitre 7

Conclusion et perspectives

Sommaire

Les travaux de ce manuscrit se placent dans le cadre du dialogue Homme-Machine situé. L'objectif global de cette thèse a été de proposer un cadre d'apprentissage en ligne continu permettant la mise en situation directe du système face à de vrais utilisateurs dès les premières étapes de son développement. Pour cela, nous avons choisi d'aborder dans nos travaux trois grandes pistes de recherche.

Dans un premier temps, nous nous sommes intéressés aux mécanismes permettant d'accélérer l'apprentissage en ligne d'une politique d'interaction. L'objectif visé par ces travaux est de rendre possible l'apprentissage en ligne de zéro de la politique de dialogue, c'est à dire directement face à des utilisateurs réels (sans simulateur d'utilisateur ou WoZ). Ce type d'apprentissage présente l'avantage de ne pas dépendre de la pré-existence d'un simulateur d'utilisateurs ou de données d'apprentissage mais également celui de plonger, au plus tôt, le système dans son véritable environnement afin d'y apprendre des politiques de meilleure qualité. Pour ce faire, nous avons proposé l'utilisation de deux sources d'informations disponibles dès le démarrage de l'apprentissage et à même d'introduire un jugement plus fin sur les comportements de l'agent apprenant pour guider au mieux l'apprentissage.

Dans un second temps, nous avons présenté une technique visant à limiter les coûts de développement d'un module de compréhension de la parole pour une nouvelle tâche. Dans cette optique, nous avons proposé d'utiliser les capacités de généralisation d'un espace sémantique appris sur une grande quantité de données, afin d'exploiter au mieux des connaissances initiales limitées voire incomplètes. Nous avons également étudié la possibilité de mettre à jour ces connaissances au travers d'une stratégie d'adaptation en ligne, possiblement apprise, mais également par l'utilisation d'une technique d'apprentissage supervisé plus classique.

Enfin, nous avons présenté la solution retenue dans le projet *MaRDi* pour mener à bien un dialogue multimodal situé entre l'Homme et le Robot. Nous avons également introduit une extension du cadre d'apprentissage RL de la politique de gestion de l'interaction pour pouvoir y intégrer l'information issue du raisonnement spatial du robot, et plus exactement de sa capacité à modéliser la perspective de l'autre. L'objectif visé étant d'améliorer les performances et l'aspect naturel de la politique apprise.

Dans la suite de ce chapitre nous revenons donc en détails sur nos différentes propositions et discutons de leurs possibles extensions.

7.1 Apprentissage de zéro et adaptatif de la politique

Comme nous venons de le rappeler, dans cette thèse nous avons proposé d'utiliser deux sources d'information distinctes pour répondre à la problématique d'un apprentissage en ligne de zéro de la politique de dialogue.

La première source que nous avons considérée est celle des connaissances expertes a priori quant à la gestion de l'interaction (règles expertes) pour leur capacité à pouvoir réduire le laps de temps (dialogues) durant lequel le système doit faire face à une politique de très mauvaise qualité. Cependant, le problème de ces connaissances réside dans leur nature imprécise, voire incomplète, qui fait que leur utilisation exclusive est sous-optimale (à terme) par rapport à l'emploi d'une méthode apprise pour déterminer la meilleure politique d'interaction. Nous avons donc proposé de les intégrer au mécanisme d'apprentissage RL en tant que :

— **conseils experts**, pour guider l'exploration initiale de l'agent apprenant. Cette solution nous a d'ailleurs permis d'atteindre de meilleures performances en début d'apprentissage. Cependant, nous avons pu constater que le fait de retarder d'exploration sur des zones de l'espace de recherche hors de la couverture experte a un impact assez négatif sur la vitesse de convergence globale du processus. L'exploration est un critère essentiel permettant d'atteindre les performances optimales à terme. Cependant, ce type de solution peut présenter l'avantage de pouvoir assurer une qualité de service « minimale » tout au long du processus d'apprentissage. C'est d'ailleurs une des pistes de réflexions que nous approfondirons plus dans la suite de nos travaux. Il s'agit de relier formellement le niveau d'exploration autorisée dans l'apprentissage de la politique au niveau de performance du système pour s'assurer, par exemple, de répartir cette exploration sur un temps plus long mais en limitant la perte de performance (mesurée par les récompenses qui combinent les effets du taux de succès et la longueur des interactions) due à cette exploration à un seuil minimal. Ce qui aurait pour effet de garantir un niveau d'usage acceptable en permanence avec la perspective d'une optimalité à terme ;

— **signal de récompense additionnel**, pour renforcer de façon anticipée le signal de renforcement principal. Comme le montre nos résultats, cette solution présente les avantages de pouvoir accélérer l'apprentissage et de garantir une meilleure tolérance aux conditions bruitées. De plus, la méthode de *reward shaping* em-

ployée dispose de propriétés qui garantissent l'optimalité de la politique apprise et ce, même dans des cas où les signaux de renforcement experts seraient contradictoire avec l'objectif d'apprentissage visé.

La seconde source d'information, vise à la prise en compte de jugements subjectifs en provenance d'utilisateurs. Ces derniers sont utilisés en tant que signaux de renforcement additionnels dans une approche de *reward shaping* similaire à celle employée dans la proposition précédente. L'exploitation de ces nouvelles récompenses, dites socialement inspirées, a permis d'accélérer sensiblement l'apprentissage de la politique, et d'améliorer la tolérances des politiques apprises au bruit. Nous nous sommes également intéressés dans notre étude aux facultés d'adaptation de la politique aux différents profils utilisateurs, pouvant bénéficier de ce type d'évaluations subjectives intermédiaires. Là encore, nous avons pu montrer que la prise en compte de ces signaux pouvaient faciliter la tâche d'adaptation en ligne de la politique. Ces observations ont également pu être réitérées dans une configuration d'apprentissage faisant intervenir de vrais utilisateurs sur la tâche *MaRDi*. L'extension naturelle de nos travaux consistera à tester cette proposition au travers de l'acquisition de « véritables » indices multimodaux. Pour ce faire, il sera nécessaire d'employer des techniques automatiques pour apprendre sur la base de données ces signaux de renforcement additionnels. Contrairement aux données issues de l'expertise, les indices sociaux, que l'ont cherche alors à détecter, peuvent présenter l'avantage d'être moins dépendants de la tâche visée, pour peu que les capteurs multimodaux employés soient les mêmes. De ce fait, il pourra être pertinent de considérer à cette fin des données plus généralistes, telles que celles employées dans le *Interspeech Computational Paralinguistics Challenge*, voire de collecter des données d'apprentissage sur plusieurs tâches.

7.2 Apprentissage sans données de référence pour la compréhension

Dans cette thèse, nous avons également présenté une approche d'apprentissage sans données de référence pour la compréhension de la parole. Celle-ci, repose à la fois, sur l'utilisation d'une représentation sémantique continue riche apprise sur des données généralistes mais également, sur une description ontologique minimale décrivant la tâche de compréhension visée. Nous avons montré que cette approche, bien que très peu coûteuse, est tout de même comparable en terme de performance à des méthodes statistiques apprises sur de grandes quantités de données annotées et aussi à un système à base de règles expertes. De plus, la méthode proposée montre une meilleur tolérance à des valeurs de concept manquantes et donc, offre des propriétés de généralisation pouvant être employées notamment dans l'extension de domaine en ligne.

De plus, nous avons pu montré qu'un processus d'adaptation en ligne simple permet de répondre aux deux limites de l'approche, à savoir, la qualité de la base de connaissance et de l'espace sémantique employé. Nous avons également proposé d'optimiser la stratégie d'adaptation en ligne avec des métriques sur l'effort de supervision fourni part de l'utilisation et sur l'amélioration du modèle qui en résulte. Bien

que dans ce manuscrit nous proposons l'utilisation d'un algorithme simple de Bandit contre un adversaire, des solutions plus évoluées, comme le bandit contextuel ou l'apprentissage par renforcement, pourront être mises en œuvre dans de futur travaux. De plus, de par l'extension de l'approche au cadre formel des CRF, il est sera également possible de combiner cette technique avec des approches supervisées plus classiques et des techniques d'apprentissage actif. Le principal problème rencontré à ce niveau est celui du croisement des courbes de performances. La solution CRF ne prend en effet le pas sur l'approche proposée ZSSP qu'à partir de plusieurs dizaines de dialogues collectés. Donc aucune des 2 méthodes n'est meilleure que l'autre en toute circonstance. Ainsi l'optimalité que nous devons rechercher permettra le démarrage avec une solution plus souple et économe vis-à-vis de ses données (ZSSP) et le basculement sur une technique plus gourmande (CRF) lorsque la quantité (et la qualité, liée à la procédure d'apprentissage actif) de données du domaine sera devenu suffisante. Notamment un des intérêts majeurs de ZSSP dans la première phase de développement est d'offrir un mécanisme permettant d'intégrer y compris les informations négatives (ce que ne permet pas CRF) et d'offrir ainsi un mécanisme implicite d'exploration au sein du modèle.

Une autre piste de travail est celle visant à étudier le lien existant entre le problème de raffinement du modèle de compréhension ainsi introduit et celui plus global de la gestion de l'interaction. La recherche d'une solution d'apprentissage conjoint de la politique d'interaction et du module SLU, voire même de la chaîne fonctionnelle complète SLU-DM-NLG, constitue pour nous la suite logique de nos travaux, et ce malgré les très grandes difficultés théoriques qu'elle soulève (notamment liées à l'évolution de la dynamique des états). L'objectif visé étant celui de proposer une solution de « bout en bout » dotée de mécanismes permettant au système de dialogue de pouvoir s'adapter à de nouvelles conditions (nouveau concept, nouvel utilisateur, etc.) en permanence.

7.3 Exploiter l'aspect situé de l'interaction

Pour finir, nous avons proposé une architecture multimodale capable de prendre en compte l'aspect situé d'un système de dialogue incarné dans un robot. Pour faciliter le déploiement du système de dialogue auprès de nos partenaires du consortium, nous avons privilégié dans nos travaux son intégration préalable dans un outil de simulation robotique en 3D. Cette solution a notamment facilité le fait de pouvoir valider la qualité de la solution retenue auprès des partenaires du projet (physiquement distants). Son portage final vers la plateforme robotique cible est cependant prévu pour début 2016.

Grâce à cette configuration particulière, les conditions d'apprentissage ont été réunies très tôt dans le projet, au point que nous avons eu l'opportunité de faire intervenir de vrais utilisateurs dans la boucle d'apprentissage et ainsi tester en pratique la solution d'apprentissage RL en ligne de zéro proposée dans ce manuscrit. La prochaine étape sera donc de déployer le système de dialogue multimodal ainsi proposé sur le robot physique et d'effectuer des évaluations en situation réelle pour valider les résultats que nous avons à ce jour obtenus.

Dans ce contexte, nous avons également proposé une extension du modèle POMDP

HIS utilisé pour la gestion du dialogue afin de prendre en compte la perspective de l'utilisateur, telle qu'estimée par le robot, dans l'apprentissage de la politique d'interaction (ajout d'une nouvelle action et d'une nouvelle composante dans l'état résumé du dialogue). L'évaluation de cette proposition a été réalisée auprès de vrais utilisateurs et nous avons pu observer que dans certaines conditions (ici celles des fausses croyances) il était possible d'apprendre une politique d'interaction plus efficace et naturelle. Une extension de nos travaux consisterait à étendre les capacités du robot à gérer l'aspect situé, en augmentant la couverture de la méthode employée pour gérer les fausses croyances (distinguer plusieurs *d-status* et définir de nouvelles actions) pour y intégrer toutes les formes de prise de perspective (niveau 1 et 2) et ainsi permettre à la politique de pouvoir raisonner efficacement dans plusieurs vues du monde concurrentes.

Au travers de ces trois pistes de recherche, nous avons proposé des solutions dont l'objectif sur le long terme est de pouvoir établir une plateforme de dialogue entièrement auto-apprenante. Dans notre vision des choses, cette dernière serait capable de démarrer avec un nombre a priori limité de connaissances et grâce à ses mécanismes d'optimisation efficaces, mais aussi à ses capacités cognitives de haut niveau (signaux sociaux, prises de perspectives, etc.), pourrait acquérir de nouvelles informations pour raffiner l'ensemble de ses modules en ligne. Cette volonté affichée s'explique par le constat que le développement d'un système est un processus très coûteux, qui nécessite le recours à l'expertise humaine (à minima le développeur du système). Notre idée est donc de bénéficier au maximum de cette situation où un humain « coopérant » est disponible en ancrant le processus d'apprentissage dès la phase conception de la chaîne du dialogue. Cependant, beaucoup de problématiques pour aller dans ce sens n'ont pas pu être abordées dans ce manuscrit et devront faire l'objet de travaux ultérieurs. Entre autres, quel cadre d'apprentissage permettrait une optimisation conjointe de l'ensemble des modules (y compris ASR et TTS qui sont eux aussi sujets aux erreurs) ? Comment serait-il possible d'identifier la ou les véritables sources d'erreurs ? Serait-il bénéfique de proposer une telle capacité d'adaptation dans les systèmes finaux proposés aux clients et quels gardes-fous faudrait-il établir pour garantir la complétude de la solution pour l'industrie ? On pourrait alors également se questionner sur l'intérêt d'une unification progressive des approches retenues dans les divers modules, à l'instar des avancées récentes obtenues avec l'utilisation des réseaux de neurones, afin de converger vers un cadre formel unique et global pour l'apprentissage de l'intégralité de la chaîne du dialogue.

Liste des illustrations

Liste des tableaux

Bibliographie

(Ai et al., 2007) H. Ai, J. R. Tetreault, & D. J. Litman, 2007. Comparing user simulation models for dialog strategy learning. Dans les actes de *NAACL HLT*.

(Ai et Weng, 2008) H. Ai & F. Weng, 2008. User simulation as testing for spoken dialog systems. Dans les actes de *SIGDIAL*.

(Alami et al., 2006) R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, & R. Chatila, 2006. Toward human-aware robot task planning. Dans les actes de *AAAI Spring Symposium : To Boldly Go Where No Human-Robot Team Has Gone Before*.

(Allen et al., 1996) J. F. Allen, B. W. Miller, E. K. Ringger, & T. Sikorski, 1996. A robust system for natural spoken dialogue. Dans les actes de *ACL*.

(Allen et al., 1995) J. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio, et al., 1995. The trains project : A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence 7*(1), 7–48.

(Anastasakos et Deoras, 2014) T. Anastasakos & A. Deoras, 2014. Task specific continuous word representations for mono and multi-lingual spoken language understanding. Dans les actes de *ICASSP*.

(Astrinaki et al., 2012) M. Astrinaki, N. D'alessandro, B. Picart, T. Drugman, & T. Dutoit, 2012. Reactive and continuous control of hmm-based speech synthesis. Dans les actes de *SLT*.

(Atrey et al., 2010) P. K. Atrey, M. A. Hossain, A. El Saddik, & M. S. Kankanhalli, 2010. Multimodal fusion for multimedia analysis : a survey. *Multimedia systems 16*(6), 345–379.

(Auer et al., 2002) P. Auer, N. Cesa-Bianchi, Y. Freund, & R. E. Schapire, 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing 32*, 48–77.

(Baerlocher et Boulic, 2004) P. Baerlocher & R. Boulic, 2004. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer 20*(6), 402–417.

(Baker et al., 1998) C. F. Baker, C. J. Fillmore, & J. B. Lowe, 1998. The berkeley framenet project. Dans les actes de *COLING*.

(Baron-Cohen et al., 1985) S. Baron-Cohen, M. Leslie, A, & U. Frith, 1985. Does the autistic child have a 'theory of mind' ? *Cognition 21*(1), 37–46.

(Barto et Mahadevan, 2003) A. G. Barto & S. Mahadevan, 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems 13*(1-2), 41–77.

(Bayer et Riccardi, 2013) A. Bayer & G. Riccardi, 2013. On-line adaptation of semantic models for spoken language understanding. Dans les actes de *ASRU*.

(Becker et al., 2006) T. Becker, N. Blaylock, C. Gerstenberger, I. Kruijff-Korbayová, A. Korthauer, M. Pinkal, M. Pitz, P. Poller, & J. Schehl, 2006. Natural and intuitive multimodal dialogue for in-car applications : The sammie system. Dans les actes de *PAIS*.

(Bellman, 1957a) R. Bellman, 1957a. Dynamic programming. *Princeton UniversityPress*.

(Bellman, 1957b) R. Bellman, 1957b. A markovian decision process. *Journal of Mathematical Mechanics 6*, 679–684.

(Bengio et Heigold, 2014) S. Bengio & G. Heigold, 2014. Word embeddings for speech recognition. Dans les actes de *INTERSPEECH*.

(Bennacef et al., 1996) S. Bennacef, L. Devillers, S. Rosset, & L. Lamel, 1996. Dialog in the railtel telephone-based system. Dans les actes de *ICSLP*.

(Beringer et al., 2002) N. Beringer, U. Kartal, K. Louka, F. Schiel, U. Türk, et al., 2002. Promise- a procedure for multimodal interactive system evaluation. Dans les actes de *Workshop on Multimodal Resources and Multimodal Systems Evaluation*.

(Bertsekas, 1995) D. Bertsekas, 1995. *Dynamic programming and optimal control*, Volume 1. Athena Scientific Belmont, MA.

(Bian et al., 2014) J. Bian, B. Gao, & T. Liu, 2014. Knowledge-powered deep learning for word embedding. Dans les actes de *ECML*.

(Black et al., 2011) A. W. Black, S. Burger, A. Conkie, H. Hastie, S. Keizer, O. Lemon, N. Merigaud, G. Parent, G. Schubiner, B. Thomson, et al., 2011. Spoken dialog challenge 2010 : Comparison of live and control test results. Dans les actes de *SIGDIAL*.

(Black et Eskenazi, 2009) A. W. Black & M. Eskenazi, 2009. The spoken dialogue challenge. Dans les actes de *SIGDIAL*.

(Bohus et Horvitz, 2010) D. Bohus & E. Horvitz, 2010. Facilitating multiparty dialog with gaze, gesture, and speech. Dans les actes de *ICMI-MLMI*.

(Bohus et Rudnicky, 2006) D. Bohus & A. Rudnicky, 2006. A 'k hypotheses + other' belief updating model. Dans les actes de *AAAI Workshop on Statistical and Empirical Methods in Spoken Dialogue Systems*.

(Bohus et Rudnicky, 2003) D. Bohus & A. I. Rudnicky, 2003. Ravenclaw : Dialog management using hierarchical task decomposition and an expectation agenda. Dans les actes de *EUROSPEECH*.

(Bohus et Rudnicky, 2005) D. Bohus & A. I. Rudnicky, 2005. Error handling in the ravenclaw dialog management framework. Dans les actes de *HLT/EMNLP*.

(Bolt, 1980) R. A. Bolt, 1980. *"Put-that-there" : Voice and gesture at the graphics interface*, Volume 14. ACM.

(Bos et al., 2003) J. Bos, E. Klein, O. Lemon, & T. Oka, 2003. Dipper : Description and formalisation of an information-state update dialogue system architecture. Dans les actes de *4th SIGdial Workshop on Discourse and Dialogue*, 115–124.

(Boularias et al., 2010) A. Boularias, H. Chinaei, & B. Chaib-draa, 2010. Learning the reward model of dialogue pomdps from data. Dans les actes de *NIPS Workshop of Machine Learning for Assistive Techniques*.

(Breazeal et al., 2006) C. Breazeal, M. Berlin, A. Brooks, J. Gray, & A. Thomaz, 2006. Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems 54*(5), 385–393.

(Breazeal et al., 2009) C. Breazeal, J. Gray, & M. Berlin, 2009. An embodied cognition approach to mindreading skills for socially intelligent robots. *International Journal of Robotics Research 28*(5), 656–680.

(Broekens et Haazebroek, 2007) J. Broekens & P. Haazebroek, 2007. Emotion and reinforcement : Affective facial expressions facilitate robot learning. Dans les actes de *Artificial Intelligence for Human Computing*, Volume 4451 de *Lecture Notes in Computer Science*, 113–132.

(Bubeck et Cesa-Bianchi, 2012) S. Bubeck & N. Cesa-Bianchi, 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning 5*(1), 1–122.

(Bui et al., 2009) T. H. Bui, M. Poel, A. Nijholt, & J. Zwiers, 2009. A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering 15*(2), 273–307.

(Burges, 2010) C. J. Burges, 2010. From ranknet to lambdarank to lambdamart : An overview. Rapport technique, Microsoft Research Technical Report MSR-TR-2010-82.

(Byron et Fosler-Lussier, 2006) D. K. Byron & E. Fosler-Lussier, 2006. The osu quake 2004 corpus of two-party situated problem-solving dialogs. Dans les actes de *LREC*.

(Camelin et al., 2011) N. Camelin, B. Detienne, S. Huet, D. Quadri, & F. Lefèvre, 2011. Unsupervised concept annotation using latent dirichlet allocation and segmental methods. Dans les actes de *EMNLP Workshop on Unsupervised Learning in NLP*.

(Cassandra et al., 1997) A. Cassandra, M. L. Littman, & N. L. Zhang, 1997. Incremental pruning : A simple, fast, exact method for partially observable markov decision processes. Dans les actes de *UAI*.

(Celikyilmaz et al., 2011) A. Celikyilmaz, G. Tur, & D. Hakkani-Tur, 2011. Leveraging web query logs to learn user intent via bayesian latent variable model. Dans les actes de *ICML*.

(Chandramohan et al., 2011) S. Chandramohan, M. Geist, F. Lefèvre, & O. Pietquin, 2011. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. Dans les actes de *INTERSPEECH*.

(Chandramohan et al., 2014) S. Chandramohan, M. Geist, F. Lefèvre, & O. Pietquin, 2014. Co-adaptation in spoken dialogue systems. Dans les actes de *Natural Interaction with Robots, Knowbots and Smartphones*, 343–353. Springer.

(Chinaei et al., 2012) H. R. Chinaei, B. Chaib-draa, & L. Lamontagne, 2012. Learning observation models for dialogue pomdps. Dans les actes de *Advances in Artificial Intelligence*, 280–286. Springer.

(Chowdhury et al., 2014) S. A. Chowdhury, A. Ghosh, E. A. Stepanov, A. O. Bayer, G. Riccardi, & I. Klasinas, 2014. Cross-language transfer of semantic annotation via targeted crowdsourcing. Dans les actes de *INTERSPEECH*.

(Clinchant et Perronnin, 2013) S. Clinchant & F. Perronnin, 2013. Aggregating continuous word embeddings for information retrieval. Dans les actes de *CVSC*.

(Cohen et Perrault, 1979) P. R. Cohen & C. R. Perrault, 1979. Elements of a plan-based theory of speech acts*. *Cognitive science* *3*(3), 177–212.

(Corradini et al., 2005) A. Corradini, M. Mehta, N. O. Bernsen, J. Martin, & S. Abrilian, 2005. Multimodal input fusion in human-computer interaction. Dans les actes de *NATO-ASI conference on Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*.

(Cuayáhuitl et al., 2005) H. Cuayáhuitl, S. Renals, O. Lemon, & H. Shimodaira, 2005. Human-computer dialogue simulation using hidden markov models. Dans les actes de *ASRU*.

(Cuayáhuitl et al., 2007) H. Cuayáhuitl, S. Renals, O. Lemon, & H. Shimodaira, 2007. Hierarchical dialogue optimization using semi-markov decision processes. Dans les actes de *INTERPSEECH*.

(Custers et Henk, 2005) R. Custers & A. Henk, 2005. Positive affect as implicit motivator : On the nonconscious operation of behavioral goals. *Personality and Social Psychology* *89*(2), 129–142.

(Daubigney et al., 2011) L. Daubigney, M. Gašić, S. Chandramohan, M. Geist, O. Pietquin, & S. Young, 2011. Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system. Dans les actes de *INTERSPEECH*.

(Daubigney et al., 2012) L. Daubigney, M. Geist, S. Chandramohan, & O. Pietquin, 2012. A comprehensive reinforcement learning framework for dialogue management optimization. *IEEE Selected Topics in Signal Processing* *6*(8), 891–902.

(Daubigney et al., 2013) L. Daubigney, M. Geist, & O. Pietquin, 2013. Particle swarm optimisation of spoken dialogue system strategies. Dans les actes de *INTERSPEECH*.

(Dauphin et al., 2014) Y. Dauphin, G. Tur, D. Hakkani-Tur, & L. Heck, 2014. Zero-shot learning and clustering for semantic utterance classification. *arXiv preprint arXiv :1401.0509*.

(De Mori et al., 2007) R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, & G. Tur, 2007. Spoken language understanding : a survey. Dans les actes de *ASRU*.

(Denecke, 2002) M. Denecke, 2002. Rapid prototyping for spoken dialogue systems. Dans les actes de *COLING*.

(Denecke et al., 2004) M. Denecke, K. Dohsaka, & M. Nakano, 2004. Learning dialogue policies using state aggregation in reinforcement learning. Dans les actes de *INTERSPEECH*.

(Deng et al., 2013) L. Deng, G. Hinton, & B. Kingsbury, 2013. New types of deep neural network learning for speech recognition and related applications : An overview. Dans les actes de *ICASSP*.

(Deng et al., 2003) Y. Deng, M. Mahajan, & A. Acero, 2003. Estimating speech recognition error rate without acoustic test data. Dans les actes de *INTERSPEECH*.

(Deoras et Sarikaya, 2013) A. Deoras & R. Sarikaya, 2013. Deep belief network based semantic taggers for spoken language understanding. Dans les actes de *INTERSPEECH*.

(Devillers et al., 2002) L. Devillers, H. Maynard, & P. Paroubek, 2002. Méthodologies d'évaluation des systèmes de dialogue parlé : réflexions et expériences autour de la compréhension. *Traitement automatique des langues 43*(2), 155–184.

(Doshi et Roy, 2008) F. Doshi & N. Roy, 2008. Spoken language interaction with model uncertainty : an adaptive human–robot interaction system. *Connection Science 20*(4), 299–318.

(Duchnowski et al., 1994) P. Duchnowski, U. Meier, & A. Waibel, 1994. See me, hear me : integrating automatic speech recognition and lip-reading. Dans les actes de *ICSLP*.

(Dumas et al., 2009) B. Dumas, D. Lalanne, & S. Oviatt, 2009. Multimodal interfaces : A survey of principles, models and frameworks. Dans les actes de *Human Machine Interaction*, 3–26. Springer.

(Dutech, 2012) A. Dutech, 2012. Self-organizing developmental reinforcement learning. Dans les actes de *SAB*.

(Dutech et Samuelides, 2003) A. Dutech & M. Samuelides, 2003. Un algorithme d'apprentissage par renforcement pour les processus decisionnels de markov partiellement observés : apprendre une extension sélective du passé. *Revue d'Intelligence Artificielle 17*(4), 559–589.

(Dybkjaer et al., 2004) L. Dybkjaer, N. O. Bernsen, & W. Minker, 2004. Evaluation and usability of multimodal spoken language dialogue systems. *Speech Communication 43*(1), 33–54.

(Echeverria et al., 2011) G. Echeverria, N. Lassabe, A. Degroote, & S. Lemaignan, 2011. Modular open robots simulation engine : Morse. Dans les actes de *ICRA*.

(Echeverria et al., 2012) G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, & S. Stinckwich, 2012. Simulating complex robotic scenarios with morse. Dans les actes de *SIMPAR*.

(Eck et al., 2015) A. Eck, L.-K. Soh, S. Devlin, & D. Kudenko, 2015. Potential-based reward shaping for finite horizon online pomdp planning. *Autonomous Agents and Multi-Agent Systems*, 1–43.

(Eckert et al., 1997) W. Eckert, E. Levin, & R. Pieraccini, 1997. User modeling for spoken dialogue system evaluation. Dans les actes de *ASRU*.

(El Asri et al., 2012) L. El Asri, R. Laroche, & O. Pietquin, 2012. Reward function learning for dialogue management. Dans les actes de *STAIRS*.

(El Asri et al., 2013) L. El Asri, R. Laroche, & O. Pietquin, 2013. Reward shaping for statistical optimisation of dialogue management. Dans les actes de *SLSP*.

(Engel et al., 2003) Y. Engel, S. Mannor, & R. Meir, 2003. Bayes meets bellman : The gaussian process approach to temporal difference learning. Dans les actes de *ICML*, Volume 20, 154.

(Ferguson et al., 1998) G. Ferguson, J. F. Allen, et al., 1998. Trips : An integrated intelligent problem-solving assistant. Dans les actes de *AAAI/IAAI*.

(Ferreira et al., 2015) E. Ferreira, G. Milliez, F. Lefèvre, & R. Alami, 2015. Users' belief awareness in reinforcement learning-based situated human-robot dialogue management. Dans les actes de *IWSDS*.

(Freese et al., 2010) M. Freese, S. Singh, F. Ozaki, & N. Matsuhira, 2010. Virtual robot experimentation platform v-rep : a versatile 3d robot simulator. Dans les actes de *Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots*, SIMPAR'10, Berlin, Heidelberg, 51–62. Springer-Verlag.

(Gales et Young, 2008) M. Gales & S. Young, 2008. The application of hidden markov models in speech recognition. *Foundations and trends in signal processing 1*(3), 195–304.

(Gandhe et Traum, 2008) S. Gandhe & D. Traum, 2008. An evaluation understudy for dialogue coherence models. Dans les actes de *SIGDIAL*.

(Gao et al., 2005) Y. Gao, L. Gu, & H. Kuo, 2005. Portability challenges in developing interactive dialogue systems. Dans les actes de *ICASSP*.

(Gašic et al., 2013) M. Gašic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, & S. Young, 2013. Pomdp-based dialogue manager adaptation to extended domains. Dans les actes de *SIGDIAL*.

(Gašić et al., 2011) M. Gašić, F. Jurčíček, B. Thomson, K. Yu, & S. Young, 2011. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. Dans les actes de *ASRU*.

(Gašic et al., 2014) M. Gašic, D. Kim, P. Tsiakoulis, C. Breslin, M. Henderson, M. Szummer, B. Thomson, & S. Young, 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains. Dans les actes de *INTERSPEECH*.

(Gašić et al., 2009) M. Gašić, F. Lefèvre, F. Jurčiček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, & S. Young, 2009. Back-off action selection in summary space-based pomdp dialogue systems. Dans les actes de *ASRU*.

(Gašić et Young, 2011) M. Gašić & S. Young, 2011. Effective handling of dialogue state in the hidden information state pomdp-based dialogue manager. *ACM Transactions on Speech and Language Processing* *7*(3), 4.

(Gašić et al., 2010) M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, & S. Young, 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. Dans les actes de *SIGDIAL*.

(Geist et Pietquin, 2010) M. Geist & O. Pietquin, 2010. Kalman temporal differences. *Artificial Intelligence Research* *39*(1), 483–532.

(Geist et Pietquin, 2011) M. Geist & O. Pietquin, 2011. Managing uncertainty within the ktd framework. Dans les actes de *Workshop on Active Learning and Experimental Design*.

(Geist et al., 2009) M. Geist, O. Pietquin, & G. Fricout, 2009. Tracking in reinforcement learning. Dans les actes de *ICONIP*.

(Georgila et al., 2005) K. Georgila, J. Henderson, & O. Lemon, 2005. Learning user simulations for information state update dialogue systems. Dans les actes de *INTERSPEECH*.

(Gibbon et al., 2001) D. Gibbon, I. Mertins, & R. K. Moore, 2001. *Handbook of multimodal and spoken dialogue systems : resources, terminology and product evaluation*, Volume 565. Springer Science & Business Media.

(Goddeau et al., 1996) D. Goddeau, H. Meng, J. Polifroni, S. Seneff, & S. Busayapongchai, 1996. A form-based dialogue manager for spoken language applications. Dans les actes de *ICSLP*.

(Goddeau et Pineau, 2000) D. Goddeau & J. Pineau, 2000. Fast reinforcement learning of dialog strategies. Dans les actes de *ICASSP*.

(Gotab et al., 2010) P. Gotab, G. Damnati, F. Béchet, & L. Delphin-Poulat, 2010. Online slu model adaptation with a partial oracle. Dans les actes de *INTERSPEECH*.

(Graves et Jaitly, 2014) A. Graves & N. Jaitly, 2014. Towards end-to-end speech recognition with recurrent neural networks. Dans les actes de *ICML*.

(Green, 1986) M. Green, 1986. A survey of three dialogue models. *ACM Transactions on Graphics 5*(3), 244–275.

(Grice et al., 1975) H. P. Grice, P. Cole, & J. L. Morgan, 1975. Syntax and semantics. *Logic and conversation 3*, 41–58.

(Hahn et al., 2010) S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, & G. Riccardi, 2010. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE/ACM Transactions on Audio Speech and Language Processing 19*(6), 1569–1583.

(Hakkani-Tur et al., 2011) D. Hakkani-Tur, L. Heck, & G. Tur, 2011. Exploiting query click logs for utterance domain detection in spoken language understanding. Dans les actes de *ICASSP*.

(He et Young, 2006) Y. He & S. Young, 2006. Spoken language understanding using the hidden vector state model. *Speech Communication 48*(3), 262–275.

(Heck et Hakkani-Tur, 2012) L. Heck & D. Hakkani-Tur, 2012. Exploiting the semantic web for unsupervised spoken language understanding. Dans les actes de *SLT*.

(Henderson et al., 2008) J. Henderson, O. Lemon, & K. Georgila, 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics 34*(4), 487–511.

(Henderson et al., 2014a) M. Henderson, B. Thomson, & J. Williams, 2014a. The second dialog state tracking challenge. Dans les actes de *SIGDIAL*.

(Henderson et al., 2014b) M. Henderson, B. Thomson, & J. Williams, 2014b. The third dialog state tracking challenge. Dans les actes de *SLT*.

(Henderson et al., 2013) M. Henderson, B. Thomson, & S. Young, 2013. Deep neural network approach for the dialog state tracking challenge. Dans les actes de *SIGDIAL*.

(Henderson et al., 2014a) M. Henderson, B. Thomson, & S. Young, 2014a. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. Dans les actes de *SLT*.

(Henderson et al., 2014b) M. Henderson, B. Thomson, & S. Young, 2014b. Word-based dialog state tracking with recurrent neural networks. Dans les actes de *SIGDIAL*.

(Hinton et al., 2012) G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., 2012. Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal Processing Magazine 29*(6), 82–97.

(Hirschman et Thompson, 1997) L. Hirschman & H. S. Thompson, 1997. Overview of evaluation in speech and natural language processing. *Survey of the State of the Art in Human Language Technology*.

(Hoey et al., 2007) J. Hoey, A. Von Bertoldi, P. Poupart, & A. Mihailidis, 2007. Assisting persons with dementia during handwashing using a partially observable markov decision process. Dans les actes de *ICVS*.

(Holzapfel et al., 2004) H. Holzapfel, K. Nickel, & R. Stiefelhagen, 2004. Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3d pointing gestures. Dans les actes de *ICMI*.

(Hone et Graham, 2000) K. S. Hone & R. Graham, 2000. Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering 6*(3&4), 287–303.

(Huet et Lefèvre, 2011) S. Huet & F. Lefèvre, 2011. Unsupervised alignment for segmental-based language understanding. Dans les actes de *EMNLP Workshop on Unsupervised Learning in NLP*.

(Hunt et Black, 1996) A. J. Hunt & A. W. Black, 1996. Unit selection in a concatenative speech synthesis system using a large speech database. Dans les actes de *ICASSP*.

(Jabaian et al., 2013) B. Jabaian, L. Besacier, & F. Lefèvre, 2013. Comparison and Combination of Lightly Supervised Approaches for Language Portability of a Spoken Language Understanding System. *IEEE/ACM Transactions on Audio Speech and Language Processing 21*(3), 636–648.

(Jabaian et al., 2014) B. Jabaian, F. Lefèvre, & L. Besacier, 2014. A unified framework for translation and understanding allowing discriminative joint decoding for multilingual speech semantic interpretation. *Computer Speech and Language 35*, 185–199.

(Johnston et al., 2002) M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, & P. Maloor, 2002. Match : An architecture for multimodal dialogue systems. Dans les actes de *ACL*.

(Jung et al., 2009) S. Jung, C. Lee, K. Kim, M. Jeong, & G. G. Lee, 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech & Language 23*(4), 479–509.

(Jurcicek et al., 2010) F. Jurcicek, B. Thomson, S. Keizer, F. Mairesse, M. Gasic, K. Yu, & S. Young, 2010. Natural belief-critic : a reinforcement algorithm for parameter estimation in statistical spoken dialogue systems. Dans les actes de *INTERSPEECH*.

(Kaelbling et al., 1998) L. Kaelbling, M. Littman, & A. Cassandra, 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence Journal 101*(1&2), 99–134.

(Kalman, 1960) R. Kalman, 1960. A new approach to linear filtering and prediction problems. *Basic Engineering 82*, 35–45.

(Keizer et al., 2010) S. Keizer, M. Gašić, F. Jurčíček, F. Mairesse, B. Thomson, K. Yu, & S. Young, 2010. Parameter estimation for agenda-based user simulation. Dans les actes de *SIGDIAL*.

(Khouzaimi et al., 2014) H. Khouzaimi, R. Laroche, & F. Lefèvre, 2014. An easy method to make dialogue systems incremental. Dans les actes de *SIGDIAL*.

(Khouzaimi et al., 2015) H. Khouzaimi, R. Laroche, & F. Lefèvre, 2015. Dialogue efficiency evaluation of turn-taking phenomena in a multi-layer incremental simulated environment. Dans les actes de *HCI International*.

(Kim et Banchs, 2014) S. Kim & R. E. Banchs, 2014. Sequential labeling for tracking dynamic dialog states. Dans les actes de *SIGDIAL*.

(Koenig et Howard, 2004) N. Koenig & A. Howard, 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. Dans les actes de *IROS*.

(Koons et al., 1993) D. B. Koons, C. J. Sparrell, & K. R. Thorisson, 1993. Integrating simultaneous input from speech, gaze, and hand gestures. *Intelligent Multi-Media Interfaces*, 257–276.

(Kunda, 1999) Z. Kunda, 1999. *Social cognition : Making sense of people*. MIT press.

(Lafferty et al., 2001) J. Lafferty, A. McCallum, & F. Pereira, 2001. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. Dans les actes de *ICML*.

(Lagoudakis et Parr, 2003) M. G. Lagoudakis & R. Parr, 2003. Least-squares policy iteration. *The Journal of Machine Learning Research 4*, 1107–1149.

(Lalanne et al., 2009) D. Lalanne, L. Nigay, P. Robinson, J. Vanderdonckt, J.-F. Ladry, et al., 2009. Fusion engines for multimodal input : a survey. Dans les actes de *ICMI*.

(Lamel et al., 2000) L. Lamel, S. Rosset, J.-L. Gauvain, S. Bennacef, M. Garnier-Rizet, & B. Prouts, 2000. The limsi arise system. *Speech Communication 31*(4), 339–353.

(Laroche et al., 2008) R. Laroche, B. Bouchon-Meunier, & P. Bretier, 2008. Uncertainty management in dialogue systems. Dans les actes de *IPMU*.

(Larochelle et al., 2008) H. Larochelle, D. Erhan, & Y. Bengio, 2008. Zero-data learning of new tasks. Dans les actes de *AAAI*.

(Larsen, 2003) L. B. Larsen, 2003. Issues in the evaluation of spoken dialogue systems using objective and subjective measures. Dans les actes de *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, 209–214. IEEE.

(Larsson et Traum, 2000) S. Larsson & D. R. Traum, 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering 6*(3&4), 323–340.

(Lavergne et al., 2010) T. Lavergne, O. Cappé, & F. Yvon, 2010. Practical very large scale crfs. Dans les actes de *ACL*.

(Lavergne et al., 2011) T. Lavergne, J. M. Crego, A. Allauzen, & F. Yvon, 2011. From n-gram-based to crf-based translation models. Dans les actes de *WSMT*.

(Lcock, 2012) G. W. Lcock, 2012. Wikitalk : A spoken wikipedia-based open-domain knowledge access system. Dans les actes de *ICCL*.

(Lee et al., 2014) B.-J. Lee, W. Lim, D. Kim, & K.-E. Kim, 2014. Optimizing generative dialog state tracker via cascading gradient descent. Dans les actes de *SIGDIAL*.

(Lee, 2013) S. Lee, 2013. Structured discriminative model for dialog state tracking. Dans les actes de *SIGDIAL*.

(Lefèvre, 2007) F. Lefèvre, 2007. Dynamic bayesian networks and discriminative classifiers for multi-stage semantic interpretation. Dans les actes de *ICASSP*.

(Lefèvre et Bonneau-Maynard, 2002) F. Lefèvre & H. Bonneau-Maynard, 2002. Issues in the development of a stochastic speech understanding system. Dans les actes de *INTERSPEECH*.

(Lefèvre et de Mori, 2007) F. Lefèvre & R. de Mori, 2007. Unsupervised state clustering for stochastic dialog management. Dans les actes de *ASRU*.

(Lefèvre et al., 2009) F. Lefèvre, M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, & S. Young, 2009. k-nearest neighbor monte-carlo control algorithm for pomdp-based dialogue systems. Dans les actes de *SIGDIAL*.

(Lefèvre et al., 2010) F. Lefèvre, F. Mairesse, & S. Young, 2010. Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation. Dans les actes de *INTERSPEECH*.

(Lefèvre et al., 2012) F. Lefèvre, D. Mostefa, L. Besacier, Y. Esteve, M. Quignard, N. Camelin, B. Favre, B. Jabaian, & L. Rojas-Barahona, 2012. Robustness and portability of spoken language understanding systems among languages and domains : the PORTMEDIA project. Dans les actes de *LREC*.

(Lemon, 2011) O. Lemon, 2011. Learning what to say and how to say it : Joint optimisation of spoken dialogue management and natural language generation. *Computer Speech & Language* *25*(2), 210–221.

(Lemon et al., 2006) O. Lemon, K. Georgila, & J. Henderson, 2006. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users : the talk towninfo evaluation. Dans les actes de *SLT*.

(Levin et Pieraccini, 2000) E. Levin & R. Pieraccini, 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing* *8*(1), 11–23.

(Levin et al., 1997) E. Levin, R. Pieraccini, & W. Eckert, 1997. Learning dialogue strategies within the markov decision process framework. Dans les actes de *ASRU*.

(Lewis et al., 2007) M. Lewis, J. Wang, & S. Hughes, 2007. Usarsim : Simulation for the study of human-robot interaction. *Cognitive Engineering and Decision Making 1*(1), 98–120.

(Li et al., 2009) L. Li, J. D. Williams, & S. Balakrishnan, 2009. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. Dans les actes de *INTERSPEECH*, 2475–2478.

(Ling et al., 2015) Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, & L. Deng, 2015. Deep learning for acoustic modeling in parametric speech generation : A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine 32*(3), 35–52.

(Lison, 2010) P. Lison, 2010. Towards relational pomdps for adaptive dialogue management. Dans les actes de *ACL*.

(Litman et al., 2000) D. J. Litman, M. S. Kearns, S. Singh, & M. A. Walker, 2000. Automatic optimization of dialogue management. Dans les actes de *COLING*.

(Littman et al., 1995) M. L. Littman, A. R. Cassandra, & L. P. Kaelbling, 1995. Efficient dynamic-programming updates in partially observable markov decision processes. Rapport technique, Computer Science Technical Report CS-95-19, Brown University.

(Lopez et al., 2011) V. Lopez, V. Uren, M. Sabou, & E. Motta, 2011. Is question answering fit for the semantic web ? a survey. *Semantic Web 2*(2), 125–155.

(Lorenzo et al., 2013) A. Lorenzo, L. Rojas-Barahona, & C. Cerisara, 2013. Unsupervised structured semantic inference for spoken dialog reservation tasks. Dans les actes de *SIGDIAL*.

(Lucignano et al., 2013) L. Lucignano, F. Cutugno, S. Rossi, & A. Finzi, 2013. A dialogue system for multimodal human-robot interaction. Dans les actes de *ICMI*.

(Macherey et al., 2009) K. Macherey, O. Bender, & H. Ney, 2009. Applications of statistical machine translation approaches to spoken language understanding. *IEEE Transactions on Audio, Speech, and Language Processing 17*(4), 803–818.

(Mairesse et al., 2009) F. Mairesse, M. Gašić, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, & S. Young, 2009. Spoken language understanding from unaligned data using discriminative classification models. Dans les actes de *ICASSP*.

(Mairesse et al., 2010) F. Mairesse, M. Gašić, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, & S. Young, 2010. Phrase-based statistical language generation using graphical models and active learning. Dans les actes de *ACL*.

(Mairesse et Young, 2014) F. Mairesse & S. Young, 2014. Stochastic language generation in dialogue using factored language models. *Computational Linguistics 40*, 763–799.

(Mangu et al., 2000) L. Mangu, E. Brill, & A. Stolcke, 2000. Finding consensus in speech recognition : word error minimization and other applications of confusion networks. *Computer Speech & Language* *14*(4), 373–400.

(Mariani, 1990) J. Mariani, 1990. Reconnaissance automatique de la parole : progrès et tendances. *Traitement du signal* *7*(4), 239–266.

(Mariani et Paroubek, 1999) J. Mariani & P. Paroubek, 1999. Human language technologies evaluation in the european framework. Dans les actes de *DARPA Broadcast News Workshop*.

(McTear, 2004) M. McTear, 2004. *Spoken dialogue technology : Towards the conversational user interface*. Springer Science & Business Media.

(McTear, 1998) M. F. McTear, 1998. Modelling spoken dialogues with state transition diagrams : experiences with the cslu toolkit. *development* *5*, 7.

(Mesnil et al., 2013) G. Mesnil, X. He, L. Deng, & Y. Bengio, 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. Dans les actes de *INTERSPEECH*.

(Meurs et al., 2009) M.-J. Meurs, F. Lefèvre, & R. De Mori, 2009. Spoken language interpretation : On the use of dynamic bayesian networks for semantic composition. Dans les actes de *ICASSP*.

(Mikolov et al., 2013a) T. Mikolov, K. Chen, G. Corrado, & J. Dean, 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

(Mikolov et al., 2013b) T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, & J. Dean, 2013b. Distributed representations of words and phrases and their compositionality. Dans les actes de *NIPS*.

(Mikolov et al., 2013c) T. Mikolov, W. Yih, & G. Zweig, 2013c. Linguistic regularities in continuous space word representations. Dans les actes de *NAACL HLT*.

(Milliez et al., 2014) G. Milliez, M. Warnier, A. Clodic, & R. Alami, 2014. A framework for endowing interactive robot with reasoning capabilities about perspective-taking and belief management. Dans les actes de *RO-MAN*.

(Minker, 1998) W. Minker, 1998. *Speech Understanding for Spoken Language Systems : Portability Across Domains and Languages*. Hänsel-Hohenhausen.

(Minker et Bennacef, 2004) W. Minker & S. Bennacef, 2004. *Speech and human-Machine dialog*. Springer Science & Business Media.

(Minker et al., 1996) W. Minker, S. Bennacef, & J.-L. Gauvain, 1996. A stochastic case frame approach for natural language understanding. Dans les actes de *ICSLP*.

(Misu et al., 2012) T. Misu, E. Mizukami, H. Kashioka, S. Nakamura, & H. Li, 2012. A bootstrapping approach for slu portability to a new language by inducting unannotated user queries. Dans les actes de *ICASSP*.

(Möbius, 2000) B. Möbius, 2000. Corpus-based speech synthesis : methods and challenges. Dans les actes de *Forum phoneticum*.

(Mori, 1997) R. D. Mori, 1997. *Spoken dialogues with computers*. Academic Press, Inc.

(Morin et Bengio, 2005) F. Morin & Y. Bengio, 2005. Hierarchical probabilistic neural network language model. Dans les actes de *AISTATS*.

(Murveit et al., 1993) H. Murveit, J. Butzberger, V. Digalakis, & M. Weintraub, 1993. Large-vocabulary dictation using sri's decipher speech recognition system : Progressive search techniques. Dans les actes de *ICASSP*.

(Nakaoka et al., 2007) S. Nakaoka, S. Hattori, F. KANEHIRO, S. Kajita, & H. Hirukawa, 2007. Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms. Dans les actes de *IROS*.

(Ng et al., 1999) A. Ng, D. Harada, & S. Russell, 1999. Policy invariance under reward transformations : Theory and application to reward shaping. Dans les actes de *ICML*.

(Ng et al., 2000) A. Y. Ng, S. J. Russell, et al., 2000. Algorithms for inverse reinforcement learning. Dans les actes de *ICML*.

(Nigay et Coutaz, 1993) L. Nigay & J. Coutaz, 1993. A design space for multimodal systems : concurrent processing and data fusion. Dans les actes de *INTERACT/CHI*.

(Nigay et Coutaz, 1995) L. Nigay & J. Coutaz, 1995. A generic platform for addressing the multimodal challenge. Dans les actes de *SIGCHI*.

(Paek, 2001) T. Paek, 2001. Empirical methods for evaluating dialog systems. Dans les actes de *ACL/EACL Workshop on Evaluation for Language and Dialogue Systems*.

(Paek, 2006) T. Paek, 2006. Reinforcement learning for spoken dialogue systems : A strengths and weaknesses for practical deployment. Dans les actes de *INTERSPEECH Dialog-on-Dialog Workshop*.

(Paek et Pieraccini, 2008) T. Paek & R. Pieraccini, 2008. Automating spoken dialogue management design using machine learning : An industry perspective. *Speech communication 50*(8), 716–729.

(Palatucci et al., 2009) M. Palatucci, D. Pomerleau, G. E. Hinton, & T. M. Mitchell, 2009. Zero-shot learning with semantic output codes. Dans les actes de *Advances in Neural Information Processing Systems 22*, 1410–1418.

(Papineni et al., 2002) K. Papineni, S. Roukos, T. Ward, & W.-J. Zhu, 2002. Bleu : a method for automatic evaluation of machine translation. Dans les actes de *ACL*.

(Parr et Russell, 1995) R. Parr & S. Russell, 1995. Approximating optimal policies for partially observable stochastic domains. Dans les actes de *IJCAI*.

(Pavlovic et Huang, 1999) V. I. Pavlovic & T. S. Huang, 1999. *Dynamic bayesian networks for information fusion with applications to human-computer interfaces*. University of Illinois at Urbana-Champaign.

(Pérez et al., 2004) P. Pérez, J. Vermaak, & A. Blake, 2004. Data fusion for visual tracking with particles. *Proceedings of the IEEE 92*(3), 495–513.

(Perrault et Allen, 1980) C. R. Perrault & J. F. Allen, 1980. A plan-based analysis of indirect speech acts. *Computational Linguistics 6*(3-4), 167–182.

(Pfleger, 2004) N. Pfleger, 2004. Context based multimodal fusion. Dans les actes de *ICMI*.

(Pieraccini et Huerta, 2005) R. Pieraccini & J. Huerta, 2005. Where do we go from here ? research and commercial spoken dialog systems. Dans les actes de *SIGDIAL*.

(Pieraccini et Levin, 1995) R. Pieraccini & E. Levin, 1995. A spontaneous-speech understanding system for database query applications. Dans les actes de *ESCA Workshop on Spoken Dialogue Systems-Theories and Applications*.

(Pieraccini et al., 2009) R. Pieraccini, D. Suendermann, K. Dayanidhi, & J. Liscombe, 2009. Are we there yet ? research in commercial spoken dialog systems. Dans les actes de *TSD*.

(Pieraccini et al., 1992) R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.-H. Lee, & J. G. Wilpon, 1992. A speech understanding system based on statistical representation of semantics. Dans les actes de *ICASSP*.

(Pietquin, 2004) O. Pietquin, 2004. *A framework for unsupervised learning of dialogue strategies*. Presses univ. de Louvain.

(Pietquin et Dutoit, 2006) O. Pietquin & T. Dutoit, 2006. A probabilistic framework for dialog simulation and optimal strategy learning. *Audio, Speech, and Language Processing, IEEE Transactions on 14*(2), 589–599.

(Pietquin et al., 2011) O. Pietquin, M. Geist, S. Chandramohan, & H. Frezza-Buet, 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing 7*(3), 7.

(Pietquin et Hastie, 2013) O. Pietquin & H. Hastie, 2013. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review 28*(1), 59–73.

(Pietquin et Renals, 2002) O. Pietquin & S. Renals, 2002. Asr system modeling for automatic evaluation and optimization of dialogue systems. Dans les actes de *ICASSP*.

(Pinault et Lefèvre, 2011a) F. Pinault & F. Lefèvre, 2011a. Semantic graph clustering for pomdp-based spoken dialog systems. Dans les actes de *INTERSPEECH*.

(Pinault et Lefèvre, 2011b) F. Pinault & F. Lefèvre, 2011b. Unsupervised clustering of probability distributions of semantic graphs for pomdp based spoken dialogue systems with summary space. Dans les actes de *KRPDS*.

(Pinault et al., 2009) F. Pinault, F. Lefèvre, & R. De Mori, 2009. Feature-based summary space for stochastic dialogue modeling with hierarchical semantic frames. Dans les actes de *INTERSPEECH*, 284–287.

(Pineau et al., 2003) J. Pineau, G. Gordon, S. Thrun, et al., 2003. Point-based value iteration : An anytime algorithm for pomdps. Dans les actes de *IJCAI*, Volume 3, 1025–1032.

(Poupart, 2005) P. Poupart, 2005. *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. Thèse de Doctorat, University of Toronto.

(Powell, 1987) M. J. Powell, 1987. Radial basis functions for multivariable interpolation : a review. Dans les actes de *Algorithms for approximation*.

(Price et Boutilier, 2003) B. Price & C. Boutilier, 2003. A bayesian approach to imitation in reinforcement learning. Dans les actes de *IJCAI*.

(Prommer et al., 2006) T. Prommer, H. Holzapfel, & A. Waibel, 2006. Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction. Dans les actes de *INTERSPEECH*.

(Puterman, 1994) M. L. Puterman, 1994. *Markov decision processes : discrete stochastic dynamic programming*. John Wiley & Sons.

(Radová et Psutka, 1997) V. Radová & J. Psutka, 1997. An approach to speaker identification using multiple classifiers. Dans les actes de *ICASSP*.

(Ralaivola et al., 2011) L. Ralaivola, B. Favre, P. Gotab, F. Béchet, & G. Damnati, 2011. Applying multiclass bandit algorithms to call-type classification. Dans les actes de *ASRU*.

(Ramshaw et Marcus, 1995) L. A. Ramshaw & M. P. Marcus, 1995. Text chunking using transformation-based learning. *arXiv preprint cmp-lg/9505040*.

(Raymond et Riccardi, 2007) C. Raymond & G. Riccardi, 2007. Generative and discriminative algorithms for spoken language understanding. Dans les actes de *INTERSPEECH*, 1605–1608.

(Reddy et Basir, 2010) B. S. Reddy & O. A. Basir, 2010. Concept-based evidential reasoning for multimodal fusion in human–computer interaction. *Applied Soft Computing 10*(2), 567–577.

(Richmond et al., 1991) V. P. Richmond, J. C. McCroskey, & S. K. Payne, 1991. *Nonverbal behavior in interpersonal relations*. Prentice Hall Englewood Cliffs, NJ.

(Rieser et Lemon, 2008) V. Rieser & O. Lemon, 2008. Learning effective multimodal dialogue strategies from wizard-of-oz data : Bootstrapping and evaluation. Dans les actes de *ACL*.

(Rieser et Lemon, 2010) V. Rieser & O. Lemon, 2010. Natural language generation as planning under uncertainty for spoken dialogue systems. Dans les actes de *Empirical methods in natural language generation*, 105–120. Springer.

(Rieser et Lemon, 2011) V. Rieser & O. Lemon, 2011. Learning and evaluation of dialogue strategies for new applications : Empirical methods for optimization from small data sets. *Computational Linguistics 37*(1), 153–196.

(Rieser et al., 2014) V. Rieser, O. Lemon, & S. Keizer, 2014. Natural language generation as incremental planning under uncertainty : Adaptive information presentation for statistical dialogue systems. *IEEE/ACM Transactions on Audio Speech and Language Processing 22*(5), 979–994.

(Roque et Traum, 2008) A. Roque & D. Traum, 2008. Degrees of grounding based on evidence of understanding. Dans les actes de *SIGDIAL*.

(Rossi et al., 2013) S. Rossi, E. Leone, M. Fiore, A. Finzi, & F. Cutugno, 2013. An extensible architecture for robust multimodal human-robot communication. Dans les actes de *IROS*.

(Rossignol et al., 2011) S. Rossignol, O. Pietquin, & M. Ianotto, 2011. Training a bn-based user model for dialogue simulation with missing data. Dans les actes de *IJCNLP*, 598–604.

(Roy et al., 2000) N. Roy, J. Pineau, & S. Thrun, 2000. Spoken dialogue management using probabilistic reasoning. Dans les actes de *ACL*.

(Rudnicky et Xu, 1999) A. Rudnicky & W. Xu, 1999. An agenda-based dialog management architecture for spoken language systems. Dans les actes de *ASRU*.

(Russell, 1998) S. Russell, 1998. Learning agents for uncertain environments. Dans les actes de *COLT*.

(Sarikaya, 2008) R. Sarikaya, 2008. Rapid bootstrapping of statistical spoken dialogue systems. *Speech Communication 50*(7), 580–593.

(Schatzmann et al., 2005) J. Schatzmann, M. Stuttle, K. Weilhammer, & S. Young, 2005. Effects of the user model on simulation-based learning of dialogue strategies. Dans les actes de *ASRU*.

(Schatzmann et al., 2007b) J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, & S. Young, 2007b. Agenda-based user simulation for bootstrapping a pomdp dialogue system. Dans les actes de *NAACL HLT*.

(Schatzmann et al., 2007a) J. Schatzmann, B. Thomson, & S. Young, 2007a. Error simulation for training statistical dialogue systems. Dans les actes de *ASRU*, 526–531. IEEE.

(Schegloff et Sacks, 1973) E. A. Schegloff & H. Sacks, 1973. Opening up closings. *Semiotica 8*(4), 289–327.

(Schmitt et al., 2011) A. Schmitt, B. Schatz, & W. Minker, 2011. Modeling and predicting quality in spoken human-computer interaction. Dans les actes de *SIGDIAL*.

(Schuller et al., 2013) B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, et al., 2013. The interspeech 2013 computational paralinguistics challenge : social signals, conflict, emotion, autism. Dans les actes de *INTERSPEECH*.

(Searle, 1969) J. R. Searle, 1969. *Speech acts : An essay in the philosophy of language*, Volume 626. Cambridge university press.

(Shafer et al., 1976) G. Shafer et al., 1976. *A mathematical theory of evidence*, Volume 1. Princeton university press Princeton.

(Shannon, 1951) C. E. Shannon, 1951. Prediction and entropy of printed english. *Bell system technical journal 30*(1), 50–64.

(Singh et al., 2002) S. Singh, D. Litman, M. Kearns, & M. Walker, 2002. Optimizing dialogue management with reinforcement learning : Experiments with the njfun system. *Journal of Artificial Intelligence Research 16*, 105–133.

(Singh et al., 1999) S. P. Singh, M. J. Kearns, D. J. Litman, & M. A. Walker, 1999. Reinforcement learning for spoken dialogue systems. Dans les actes de *NIPS*.

(Skantze et Schlangen, 2009) G. Skantze & D. Schlangen, 2009. Incremental dialogue processing in a micro-domain. Dans les actes de *EACL*.

(Smets et Kennes, 1994) P. Smets & R. Kennes, 1994. The transferable belief model. *Artificial intelligence 66*(2), 191–234.

(Sondik, 1971) E. J. Sondik, 1971. *The Optimal Control of Partially Observable Markov Processes*. Thèse de Doctorat, Stanford University.

(Spaan et Vlassis, 2005) M. T. Spaan & N. Vlassis, 2005. Perseus : Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research*, 195–220.

(Spall, 2005) J. C. Spall, 2005. *Introduction to stochastic search and optimization : estimation, simulation, and control*. John Wiley & Sons.

(Stiefelhagen et al., 2007) R. Stiefelhagen, H. K. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, & A. Waibel, 2007. Enabling multimodal human–robot interaction for the karlsruhe humanoid robot. *IEEE Transactions on Robotics 23*(5), 840–851.

(Stiefelhagen et al., 2004) R. Stiefelhagen, C. Fügen, P. Gieselmann, H. Holzapfel, K. Nickel, & A. Waibel, 2004. Natural human-robot interaction using speech, head pose and gestures. Dans les actes de *IROS*.

(Strobel et al., 2001) N. Strobel, S. Spors, & R. Rabenstein, 2001. Joint audio-video object localization and tracking. *Signal Processing Magazine, IEEE 18*(1), 22–31.

(Strzalkowski et Harabagiu, 2006) T. Strzalkowski & S. Harabagiu, 2006. *Advances in open domain question answering*, Volume 32. Springer Science & Business Media.

(Stuttle et al., 2004) M. N. Stuttle, J. D. Williams, & S. Young, 2004. A framework for dialogue data collection with a simulated asr channel. Dans les actes de *INTERSPEECH*.

(Su et al., 2015) P.-H. Su, D. Vandyke, M. Gašic, N. Mrkšic, T.-H. Wen, & S. Young, 2015. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. Dans les actes de *SIGDIAL*.

(Sungjin et Eskenazi, 2012) L. Sungjin & M. Eskenazi, 2012. Incremental sparse bayesian method for online dialog strategy learning. *IEEE Selected Topics in Signal Processing 6*, 903–916.

(Sutton et Barto, 1998) R. Sutton & A. Barto, 1998. Reinforcement learning : An introduction. *IEEE Transactions on Neural Networks 9*(5), 1054–1054.

(Sutton et al., 2007) R. Sutton, A. Koop, & D. Silver, 2007. On the role of tracking in stationary environments. Dans les actes de *ICML*.

(Sutton et al., 1996) S. Sutton, D. G. Novick, R. Cole, P. Vermeulen, J. De Villiers, J. Schalkwyk, & M. Fanty, 1996. Building 10,000 spoken dialogue systems. Dans les actes de *ICSLP*.

(Szita et al., 2006) I. Szita, V. Gyenes, & A. Lőrincz, 2006. Reinforcement learning with echo state networks. Dans les actes de *ICANN*.

(Taylor et Stone, 2009) M. E. Taylor & P. Stone, 2009. Transfer learning for reinforcement learning domains : A survey. *The Journal of Machine Learning Research 10*, 1633–1685.

(Taylor, 2009) P. Taylor, 2009. *Text-to-speech synthesis*. Cambridge university press.

(Thomson et al., 2010) B. Thomson, F. Jurčíček, M. Gašić, S. Keizer, F. Mairesse, K. Yu, & S. Young, 2010. Parameter learning for pomdp spoken dialogue models. Dans les actes de *SLT*.

(Thomson et Young, 2010) B. Thomson & S. Young, 2010. Bayesian update of dialogue state : A pomdp framework for spoken dialogue systems. *Computer Speech and Language 24*(4), 562–588.

(Thomson et al., 2008) B. Thomson, K. Yu, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, & S. Young, 2008. Evaluating semantic-level confidence scores with multiple hypotheses. Dans les actes de *INTERSPEECH*.

(Thorndike, 1932) E. L. Thorndike, 1932. The fundamentals of learning.

(Trafton et al., 2005) J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, & A. Schultz, 2005. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics 25*(4), 460–470.

(Traum et Larsson, 2003) D. Traum & S. Larsson, 2003. The information state approach to dialogue management. Dans les actes de *Current and New Directions in Discourse and Dialogue*, Volume 22 de *Text, Speech and Language Technology*, 325–353. Springer.

(Traum, 1994) D. R. Traum, 1994. *A Computational Theory of Grounding in Natural Language Conversation.* Thèse de Doctorat, University of Rochester.

(Traum, 1999) D. R. Traum, 1999. Speech acts for dialogue agents. Dans les actes de *Foundations of rational agency*, 169–201. Springer.

(Tsitsiklis et Van Roy, 1997) J. N. Tsitsiklis & B. Van Roy, 1997. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control 42*(5), 674–690.

(Tur et al., 2011) G. Tur, D. Hakkani-tur, D. Hillard, & A. Celikyilmaz, 2011. Towards unsupervised spoken language understanding : Exploiting query click logs for slot filling. Dans les actes de *INTERSPEECH*.

(Tur et al., 2005) G. Tur, D. Hakkani-Tur, & R. Schapire, 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication 45*(2), 171–186.

(Tur et al., 2003) G. Tur, G. Rahim, & D. Hakkani-Tur, 2003. Active labeling for spoken language understanding. Dans les actes de *EUROSPEECH*.

(Turunen et al., 2006) M. Turunen, J. Hakulinen, & A. Kainulainen, 2006. Evaluation of a spoken dialogue system with usability tests and long-term pilot studies : similarities and differences. Dans les actes de *INTERSPEECH*.

(Tversky et al., 1999) B. Tversky, P. Lee, & S. Mainwaring, 1999. Why do speakers mix perspectives ? *Spatial Cognition and Computation 1*(4), 399–412.

(Ultes et al., 2015) S. Ultes, M. Kraus, A. Schmitt, & W. Minker, 2015. Quality-adaptive spoken dialogue initiative selection and implications on reward modelling. Dans les actes de *SIGDIAL*, 374.

(Vinciarelli et al., 2009) A. Vinciarelli, M. Pantic, & H. Bourlard, 2009. Social signal processing : Survey of an emerging domain. *Image and Vision Computing 27*(12), 1743–1759.

(Řehuuřek et Sojka, 2010) R. Řehuuřek & P. Sojka, 2010. Software framework for topic modelling with large corpora. Dans les actes de *LREC*.

(Vukotic et al., 2015) V. Vukotic, C. Raymond, & G. Gravier, 2015. Is it time to switch to word embedding and recurrent neural networks for spoken language understanding ? Dans les actes de *INTERSPEECH*.

(Wahlster, 2002) W. Wahlster, 2002. Smartkom : Fusion and fission of speech, gestures, and facial expressions. Dans les actes de *IWMMS*.

(Wahlster, 2006) W. Wahlster, 2006. *SmartKom : foundations of multimodal dialogue systems*, Volume 12. Springer.

(Walker et al., 1997) M. Walker, D. Litman, C. Kamm, & A. Abella, 1997. Paradise : a framework for evaluating spoken dialogue agents. Dans les actes de *ACL*.

(Walker, 2000) M. A. Walker, 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Artificial Intelligence Research 12*, 387–416.

(Walker et al., 1998) M. A. Walker, D. J. Litman, C. A. Kamm, & A. Abella, 1998. Evaluating spoken dialogue agents with paradise : Two case studies. *Computer Speech & Language 12*(4), 317–347.

(Walker et al., 2007) M. A. Walker, A. Stent, F. Mairesse, & R. Prasad, 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research 30*, 413–456.

(Wallace, 2003) R. Wallace, 2003. The elements of aiml style. *Alice AI Foundation*.

(Wang et al., 2003) Y. Wang, T. Tan, & A. K. Jain, 2003. Combining face and iris biometrics for identity verification. Dans les actes de *AVBPA*.

(Wang et Acero, 2006) Y.-Y. Wang & A. Acero, 2006. Discriminative models for spoken language understanding. Dans les actes de *INTERSPEECH*.

(Wang et al., 2005) Y.-Y. Wang, L. Deng, & A. Acero, 2005. Spoken language understanding. *Signal Processing Magazine, IEEE 22*(5), 16–31.

(Wang et al., 2000) Y.-Y. Wang, M. Mahajan, & X. Huang, 2000. A unified context-free grammar and n-gram model for spoken language processing. Dans les actes de *ICASSP*.

(Ward, 1994) W. Ward, 1994. Extracting information in spontaneous speech. Dans les actes de *ICSLP*.

(Ward et Issar, 1994) W. Ward & S. Issar, 1994. Recent improvements in the cmu spoken language understanding system. Dans les actes de *HLT*.

(Watanabe et al., 1998) T. Watanabe, M. Araki, & S. Doshita, 1998. Evaluating dialogue strategies under communication errors using computer-to-computer simulation. *IEICE transactions on information and systems 81*(9), 1025–1033.

(Watkins et Dayan, 1992) C. J. Watkins & P. Dayan, 1992. Q-learning. *Machine learning 8*(3-4), 279–292.

(Wen et al., 2015) T.-H. Wen, M. Gašic, D. Kim, N. Mrkšic, P.-H. Su, D. Vandyke, & S. Young, 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. Dans les actes de *SIGDIAL*.

(Williams et al., 2013) J. Williams, A. Raux, D. Ramachandran, & A. Black, 2013. The dialog state tracking challenge. Dans les actes de *SIGDIAL*.

(Williams, 2007) J. D. Williams, 2007. Using particle filters to track dialogue state. Dans les actes de *ASRU*.

(Williams, 2008a) J. D. Williams, 2008a. The best of both worlds : unifying conventional dialog systems and pomdps. Dans les actes de *INTERSPEECH*.

(Williams, 2008b) J. D. Williams, 2008b. Integrating expert knowledge into pomdp optimization for spoken dialog systems. Dans les actes de *AAAI Workshop on Advancements in POMDP Solvers*.

(Williams, 2014) J. D. Williams, 2014. Web-style ranking and slu combination for dialog state tracking. Dans les actes de *SIGDIAL*.

(Williams et al., 2005) J. D. Williams, P. Poupart, & S. Young, 2005. Factored partially observable markov decision processes for dialogue management. Dans les actes de *KRPDS*.

(Williams et Young, 2006) J. D. Williams & S. Young, 2006. Scaling pomdps for dialog management with composite summary point-based value iteration (cspbvi). Dans les actes de *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, 37–42.

(Wimmer et Perner, 1983) H. Wimmer & J. Perner, 1983. Beliefs about beliefs : Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition 13*(1), 103–128.

(Wu et al., 2005) Z. Wu, L. Cai, & H. Meng, 2005. Multi-level fusion of audio and visual features for speaker identification. Dans les actes de *Advances in Biometrics*, 493–499. Springer.

(Yao et al., 2013) K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, & F. Gao, 2013. Recurrent conditional random fields. Dans les actes de *NIPS*.

(Yao et al., 2014) k. Yao, B. Peng, G. Zweig, D. Yu, X. Li, & F. Gao, 2014. Recurrent conditional random field for language understanding. Dans les actes de *ICASSP*.

(Yao et al., 2013) K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, & D. Yu, 2013. Recurrent neural networks for language understanding. Dans les actes de *INTERSPEECH*.

(Young, 2002) S. Young, 2002. Talking to machines (statistically speaking). Dans les actes de *INTERSPEECH*.

(Young, 2007) S. Young, 2007. Cued standard dialogue acts. Rapport technique, Cambridge University Engineering Dept.

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

(Young et Proctor, 1989) S. Young & C. Proctor, 1989. The design and implementation of dialogue control in voice operated database inquiry systems. *Computer Speech & Language 3*(4), 329–353.

(Young, 2000) S. J. Young, 2000. Probabilistic methods in spoken–dialogue systems. *Philosophical Transactions of the Royal Society of London A : Mathematical, Physical and Engineering Sciences 358*(1769), 1389–1402.

(Zou et al., 2013) W. Zou, R. Socher, D. Cer, & C. Manning, 2013. Bilingual word embeddings for phrase-based machine translation. Dans les actes de *EMNLP*.

(Zou et Bhanu, 2005) X. Zou & B. Bhanu, 2005. Tracking humans using multi-modal fusion. Dans les actes de *CVPR*.

Bibliographie personnelle

Revues internationales avec comité de sélection

FERREIRA E. ET LEFÈVRE F. « Reinforcement-Learning Based Dialogue System for Human-Robot Interactions with Socially-inspired Rewards » *dans Computer Speech & Language*. 2015

Conférences d'audience internationale avec comité de sélection

FERREIRA E., REIFFERS MASSON A. ET JABAIAN B. ET LEFÈVRE F. « Adversarial bandit for online interactive active learning of zero-shot spoken language understanding » *dans ICASSP*, 2016

FERREIRA E., JABAIAN B. ET LEFÈVRE F. « Zero-shot semantic parser for spoken language understanding » *dans INTERSPEECH*, 2015

FERREIRA E., JABAIAN B. ET LEFÈVRE F. « Online adaptative zero-shot learning spoken language understanding using word-embedding » *dans ICASSP*. 2015

FERREIRA E., MILLIEZ G., LEFÈVRE F ET ALAMI R. « Users' Belief Awareness in Reinforcement Learning-based Situated Human-Robot Dialogue Management » *dans IWSDS*. 2015

MILLIEZ G., FERREIRA E., ALAMI R. ET LEFÈVRE F « Simulating human-robot interactions for dialogue strategy learning » *dans SIMPAR*. 2014

FERREIRA E. ET LEFÈVRE F. « On the use of social signal for reward shaping in reinforcement learning for dialogue management » *dans SemDIal*. 2013

FERREIRA E. ET LEFÈVRE F. « Expert-based reward shaping and exploration scheme for boosting policy learning of dialogue management » *dans ASRU*. 2013

FERREIRA E. ET LEFÈVRE F. « Social signal and user adaptation in reinforcement learning-based dialogue management » *dans MLIS*. 2013

FERREIRA E., NOCERA P., GOUDI M. ET. THI N. « YAST: A scalable ASR toolkit especially designed for under-resourced languages » *dans IALP*. 2012

BOUALLEGUE M., FERREIRA E., MATROUF D., LINARES G., GOUDI M. ET NOCERA P. « Acoustic modeling for under-resourced languages based on vectorial HMM-states representation using Subspace Gaussian Mixture Models » *dans SLT*. 2012

BENKHELLAT Z., FERREIRA E., NOCERA P. ET GUERTI M. « Automatic speech recognition system for under-resourced languages based on Speeral: application to berber language » *dans SLTU*. 2012

Conférences d'audience nationale avec comité de sélection

FERREIRA E., JABAIAN B. ET LEFÈVRE F. « Compréhension automatique de la parole sans données de référence » *dans TALN*, 2015

Défis scientifique d'audience internationale

COSSU JV., JANOD K., FERREIRA E., GAILLARD J. ET EL-BÈZE M. « LIA@ Replab 2014: 10 methods for 3 tasks » *in Replab*, 2014

Annexes

Annexes A

Actes de dialogue

Les actes de dialogue sont une des formes envisageables pour la représentation du contenu sémantique d'un énoncé, qu'il provienne de soit utilisateur ou soit émis par le système. Dans la littérature, plusieurs taxonomies ont été proposées pour définir cette notion. Dans cette thèse cependant, nous avons fait l'usage exclusif du standard proposé par le groupe CUED de l'université de Cambridge.

Dans ce formalisme un acte de dialogue est défini comme étant la combinaison d'une étiquette identifiant l'intention dialogique portée par la phrase dont on veut extraire le sens (e.g. une demande d'information ou de confirmation, une affirmation d'un fait, etc.) et d'une séquence (optionnelle) d'arguments traduisant les informations sémantiques transmises par l'acte de dialogue :

$$acttype(\underbrace{a = x, b = y, ...}_{arguments})$$

L'*acttype* correspond au type de l'acte de dialogue. Les différents types d'actes de dialogue possibles peuvent être considérés comme indépendants de la tâche visée. On peut les diviser en 4 grands groupes :

— ceux ayant pour but de transmettre de l'information (`inform`);
— ceux représentant différents types de requête (`request, reqalts, reqmore`);
— ceux relatifs aux confirmations (`confirm, affirm, negate, deny`);
— et enfin la dimension dialogique, avec principalement les formules de politesse (`hello, thankyou, bye`).

Les arguments sont optionnels et sont majoritairement des paires concept-valeur, ou simplement des concepts ou des valeurs. Par exemple, dans le contexte d'un système de dialogue ayant pour objectif de donner des informations sur les établissements d'une ville, un exemple d'acte de dialogue avec un couple concept-valeur peut être *inform(food=french)*. Cet acte permet d'indiquer le fait que l'utilisateur informe le système qu'il désire un établissement servant de la « nourriture française ». Un autre exemple d'acte de dialogue avec un seul concept peut être *request(phone)* employé pour traduire le cas où l'utilisateur demande le numéro de téléphone d'un établissement au système.

Il est à noter qu'il peut également y avoir des valeurs particulières, comme *dontcare*. Cette dernière est utilisée pour indiquer que l'utilisateur relâche une contrainte (par exemple *food=dontcare* qui peut se traduire littéralement par « peu importe le type de nourriture »).

Dans notre étude, nous considérons cependant 2 variantes de ce standard standard d'annotation selon la tâche adressée. Nous distinguons donc le standard d'annotation utilisé pour les tâches *TownInfo* et *MaRDi* et celui des tâches *DSTC2* et *DSTC3*.

A.1 Standard d'annotation sémantique des tâches *TownInfo* et *MaRDi*

Dans ce formalisme chaque énoncé utilisateur est représenté sous la forme d'un seul acte de dialogue dont l'*acttype* est l'intention dominante de l'énoncé et les arguments sont tous les couples concept-valeur ou concepts détectés. Par exemple l'énoncé « je recherche un restaurant français dans le nord et j'aimerais son numéro de téléphone » va par exemple être annoté par *request(phone, task=find, type=restaurant,food=french,area=north)*.

Le Tableau A.1 donne la liste des actes de dialogue système et utilisateur employés dans les tâches *TownInfo* et *MaRDi*. Le Tableau A.2, montre un exemple d'annotation sémantique sur un dialogue effectué pour la tâche *TownInfo*.

A.2 Standard d'annotation sémantique des tâches *DSTC2* et *DSTC3*

Pour ce qui est du format d'annotation sémantique employé sur les tâches *DSTC2* et *DSTC3*, on peut constater qu'il existe une légère différence avec la représentation sémantique présentée ci-dessus. En effet, les actes de dialogue sont dans ce cas précis décomposés en formes unitaires afin de couvrir plus finement la sémantique de la phrase. Par exemple la phrase « ok quel est le numéro de téléphone » qui aurait été annotée en *request(phone)* selon le format précédent est cette fois ci annotée par *ack()* | *request(phone)*. De même la phrase « je voudrais le numéro de téléphone et l'adresse » va être cette fois annotée *request(phone)* | *request(address)* au lieu de *request(phone,address)*. Il y a aussi quelques différences sur les noms donnés aux types d'actes de dialogue, notamment ceux des actes utilisés pour annoter les énoncés du système.

Le tableau A.3 donne une nouvelle liste des actes de dialogue système et utilisateur et le tableau A.4, et reprend le même exemple de dialogue que précédemment mais cette fois annoté selon le formalisme employé dans *DSTC2* et *DSTC3*. Il est à noter qu'avec quelques règles de conversion simples il est possible de passer sans perte de ce nouveau format vers l'ancien (ce qui ne sera pas le cas dans l'autre sens).

Actes	**Sys**	**Usr**	**Description**
hello()	✓	✓	initie le dialogue
hello(a=x,b=y, . . .)	✗	✓	initie le dialogue avec a=x, b=y, ...
silence()	✗	✓	l'utilisateur reste silencieux
thankyou()	✗	✓	l'utilisateur remercie le système
ack()	✗	✓	l'utilisateur manifeste sa compréhension de la réponse du système
bye()	✓	✓	message de clôture du dialogue
hangup()	✗	✓	l'utilisateur raccroche
inform(a=x, b=y, . . .)	✓	✓	informe que a=x, b=y, ...
inform(name=none)	✓	✗	informe qu'aucune entité de la base de données ne répond aux critères spécifiés par l'utilisateur
inform(a !=x, . . .)	✗	✓	informe que a n'est pas x
inform(a=dontcare, . . .)	✗	✓	l'utilisateur ne se soucie pas de la valeur de a
request(a)	✓	✓	demande la valeur de a
request(a, b=x, . . .)	✓	✓	demande la valeur de a sachant b=x,...
reqalts()	✗	✓	demande d'alternative à la solution actuelle
reqalts(a=x, . . .)	✗	✓	demande d'alternative avec a=x,...
reqalts(a=dontcare, . . .)	✗	✓	demande d'alternative en relâchant la contrainte sur a
reqmore()	✓	✗	demande si l'utilisateur désire d'autres informations
reqmore()	✗	✓	demande des informations supplémentaires sur la solution courante
reqmore(a=dontcare)	✓	✗	demande à l'utilisateur s'il souhaite relâcher la contrainte mise sur a
reqmore(a=x,b=y, . . .)	✗	✓	demande plus d'informations au système en donnant comme contraintes supplémentaires a=x, b=y, ...
confirm(a=x,b=y, . . .)	✓	✓	demande de confirmation explicite sur a=x, b=y, ...
confirm(a !=x, . . .)	✓	✓	demande de confirmation explicite sur a !=x, ...
confirm(name=none)	✗	✓	demande de confirmation sur le fait qu'aucune entité ne peut être trouvé
confreq(a=x,...,c=z,d)	✓	✗	demande de confirmation implicite sur a=x, ... , c=z et demande de la valeur de d
select(a=x, a=y)	✓	✗	demande à l'utilisateur de choisir a=x ou a=y
affirm()	✓	✓	simple affirmation
affirm(a=x,b=y, . . .)	✓	✓	affirme et donne plus d'informations (a=x, b=y, ...)
negate()	✓	✓	simple négation
negate(a=x)	✓	✓	nie la valeur courante de a et informe que a=x
negate(a=x,b=y, . . .)	✓	✓	negate(a=x) et donne plus d'informations (b=y, ...)
deny(a=x,b=y)	✗	✓	nie a=x et donne plus d'informations (b=y, ...)
repeat()	✓	✓	demande à l'autre participant de répéter sa phrase
help()	✗	✓	demande d'aide
restart()	✗	✓	demande au système de redémarrer depuis le début
null()	✗	✓	quelque chose hors du cadre de compréhension du système

TABLE *A.1* – *Liste des actes de dialogue employés dans TownInfo et MaRDi.*

Tours	Phrases	Annotations sémantiques
Sys1 :	Bonjour comment puis-je vous aider ?	hello()
Usr1 :	Je suis à la recherche d'un bar	inform(task=find, type=bar)
Sys2 :	Ok un bar. Dans quelle partie de la ville ?	confreq(type=bar, area)
Usr2 :	Dans le centre près du cinéma	inform(area=central, near=cinema)
Sys3 :	The parrot est un bon bar à deux pas du cinéma	inform(name="the parrot", type=bar, near=cinema)
Usr3 :	Quelle est sa gamme de prix ?	request(pricerange)
Sys4 :	The parrot est un bar dans la gamme de prix modérée	inform(name="the parrot", type=bar, pricerange=moderate)
Usr4 :	Ok quel est son numéro de téléphone	request(phone)
Sys5 :	Son numéro est le 728-66-32	inform(phone=728-66-32)
Usr5 :	ok	ack()
Sys6 :	Avez-vous besoin d'autres choses ?	reqmore()
Usr6 :	Non merci	negate()
Sys7 :	Au revoir et merci d'avoir utilisé notre système	bye()

TABLE *A.2* – *Exemple d'annotation sémantique dans le format employé dans TownInfo et MaRDi.*

Actes	Sys	Usr	Description
hello()	✗	✓	le message employé pour saluer le système
welcomemsg()	✓	✗	le message utilisé pour saluer l'utilisateur
silence()	✗	✓	l'utilisateur reste silencieux
thankyou()	✗	✓	l'utilisateur remercie le système
ack()	✗	✓	l'utilisateur manifeste sa compréhension de la réponse du système
bye()	✓	✓	message de clôture du dialogue
hangup()	✗	✓	l'utilisateur raccroche
inform(a=x)	✓	✓	informe que a vaut x
offer(name=x)	✓	✗	le système propose l'établissement x à l'utilisateur
canthelp(a = x, b = y, ...)	✓	✗	informe l'utilisateur que la combinaison des contraintes a = x, b = y, ... ne correspond à aucune entité dans la base de données
inform(a=dontcare)	✗	✓	l'utilisateur ne se soucie pas de la valeur de a
request(a)	✓	✓	demande la valeur de a
reqalts()	✗	✓	demande de recherche d'alternative
reqmore()	✓	✗	demande si l'utilisateur désire autre chose
reqmore()	✗	✓	demande plus d'information concernant la solution courante
confirm(a=x)	✗	✓	demande de confirmation sur a=x de la part de l'utilisateur
expl-conf(a=x)	✓	✗	demande de confirmation explicite de la part du système
impl-conf(a=x)	✓	✗	demande de confirmation implicite de la part du système
select(a=x)	✓	✗	demande à l'utilisateur de choisir une valeur parmi plusieurs sur un même attribut. Il apparaît toujours avec au moins un autre acte de dialogue de type *select* avec le même a mais avec une valeur différente de x
affirm()	✓	✓	simple affirmation
negate()	✓	✓	simple négation
deny(a=x)	✗	✓	nie que a=x
repeat()	✓	✓	demande à l'autre participant de répéter sa phrase
help()	✗	✓	demande d'aide
restart()	✗	✓	demande au système de redémarrer depuis le début
null()	✗	✓	quelque chose hors du cadre de compréhension du système

TABLE *A.3* – *Liste des actes de dialogue employés dans DSTC2 et DSTC3.*

Tours	Phrases	Annotations sémantiques
Sys1 :	Bonjour comment puis-je vous aider ?	welcomemsg()
Usr1 :	Je suis à la recherche d'un bar	inform(task=find) | inform(type=bar)
Sys2 :	Ok un bar. Dans quelle partie de la ville ?	impl-conf(type=bar) | request(area)
Usr2 :	Dans le centre près du cinéma	inform(area=central) | inform(near=cinema)
Sys3 :	The parrot est un bon bar à deux pas du cinéma	inform(name="the parrot") | inform(type=bar) | inform(near=cinema)
Usr3 :	Quelle est sa gamme de prix ?	request(pricerange)
Sys4 :	The parrot est un bar dans la gamme de prix modérée	inform(name="the parrot") | inform(type=bar) | inform(pricerange=moderate)
Usr4 :	Ok quel est son numéro de téléphone ?	ack() | request(phone)
Sys5 :	Son numéro est le 728-66-32	inform(phone=728-66-32)
Usr5 :	ok	ack()
Sys6 :	Avez-vous besoin d'autres choses	reqmore()
Usr6 :	Non merci	negate() | thankyou()
Sys7 :	Au revoir et merci d'avoir utilisé notre système	bye()

TABLE *A.4* – *Exemple d'annotation sémantique dans le format employé dans DSTC2 et DSTC3.*

Annexes B

Métriques d'évaluation usuelles

On peut distinguer deux types de métrique pour l'évaluation des sorties ASR et SLU :

— celle qui tient compte de la séquences des étiquettes sorties par le système ;
— celle qui en fait une évaluation plus globale.

B.1 Le taux d'erreur de mots (WER)

Le taux d'erreur de mots [1] (WER) est l'unité de mesure généralement employé pour mesurer les performances d'un ASR. Le WER est dérivé de la distance de Levenshtein, en travaillant au niveau des mots au lieu des caractères. Il peut donc être défini comme le ratio de la somme des mots incorrectement reconnus (omis, insérés et substitués) sur le nombre de mots total dans le texte de référence. Plus le taux est faible (minimum 0) plus la reconnaissance est bonne. D'après sa définition (donnée ci-dessous) le taux maximum n'est pas borné et peut donc dépasser 1 sur des cas limites (beaucoup d'insertions).

$$WER = \frac{S_w + D_w + I_w}{N_w} \tag{B.1}$$

où S_w est le nombre de mots qui ont été substitués dans l'hypothèse de transcription par rapport à la référence, D_w celui des mots omis dans l'hypothèse de transcription et I_w celui de ceux insérés. Enfin N_w correspond au nombre total de mots dans la référence. Il est à noter que ce taux est souvent donné en pourcentage dans la littérature.

B.2 Le taux d'erreur de concepts (CER)

Lorsque l'on considère des énoncés avec des références étiquetées en concept au niveau des mots il peut être intéressant d'employer une métrique d'évaluation du SLU

1. en anglais Word Errror Rate

capable de préserver l'importance de l'ordre des dites séquences. De façon similaire au WER, il est donc possible de définir le CER qui lui travaillera, comme son nom l'indique, au niveau des concepts (étiquettes sémantique). Il requiert quant à la lui une référence qui aligne chaque mot ou séquence de mots de la phrase à la bonne étiquette sémantique.

$$CER = \frac{S_c + D_c + I_c}{N_c} \tag{B.2}$$

où S_c est le nombre de concepts qui ont été substitués dans l'hypothèse de compréhension par rapport à la référence, D_c celui des concepts omis et I_c celui de ceux insérés. Enfin N_c correspond au nombre total de concepts dans la référence.

B.3 La F-mesure (F-score)

La F-mesure (*F-score*) peut être utilisée en tant que métrique traduisant de la qualité globale de l'hypothèse de compréhension faite par le SLU. Cette mesure est la moyenne harmonique de la **Précision** et du **Rappel** de la meilleure hypothèse sémantique.

$$\text{F-mesure} = \frac{2 \cdot (\text{Précision} \cdot \text{Rappel})}{\text{Précision} + \text{Rappel}} \tag{B.3}$$

avec

$$\text{Précision} = \frac{\text{nombre de concepts corrects trouvés}}{\text{nombre de concepts trouvés}} \tag{B.4}$$

et

$$\text{Rappel} = \frac{\text{nombre de concepts corrects trouvés}}{\text{nombre de concepts à trouver}} \tag{B.5}$$

Il est à noter que ces indicateurs, contrairement au CER, ne tiennent pas compte de l'aspect séquentiel du processus de décodage sémantique. Ainsi, les références employées pour son calcul pourront ne pas être alignées sémantiquement au niveau des mots.

Annexes C

Ontologie du domaine

Afin de réaliser la tâche qui lui incombe, le système de dialogue doit en avoir une représentation formelle. Cette représentation prend généralement la forme d'une ontologie mentionnant les différents concepts manipulés, leurs propriétés et éventuellement leurs relations. Ces données sont également mises en relation avec une source de connaissance externe (base de données, web) afin de pouvoir répondre concrètement à la tâche. Dans cette thèse 4 tâches (4 ontologies) ont été couvertes et nous en donnons ici le détail.

C.1 Description de l'ontologie d'un domaine

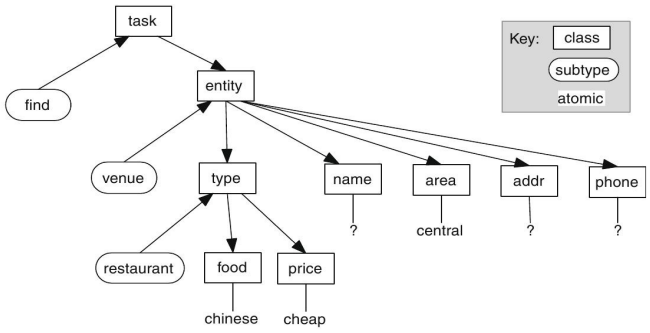


FIGURE *C.1* – *Formalisme de description de l'ontologie dans notre système.*

Suivant les propositions dans le cadre du paradigme HIS (Young et al., 2010), dans notre DM, un but utilisateur est représenté sous la forme d'une structure arborescente similaire à celle reprise sur la figure C.1. Cette dernière est composée de classes abstraites, d'instances (subtypes), de concepts (ou classes terminales) et de valeurs atomiques. Les classes abstraites sont des structures qui regroupent d'autres classes abstraites ou concepts liés. Les concepts sont des classes terminales associées à des valeurs atomiques. Les instances correspondent quant à elles aux différentes variantes d'une

même classe abstraite et conditionnent les concepts et les classes abstraites qui y sont effectivement associés.

task	→	find(entity)	{0.4}
entity	→	venue(name, type, area)	{0.8}
venue	→	bar(drinks, music)	{0.4}
venue	→	restaurant(food, pricerange)	{0.3}
area	=	(south | north | east | west | central)	
food	=	(french | italian | chinese | ...)	
...			

TABLE *C.1* – *Exemple d'ontologie du domaine telle qu'exploitée dans notre système.*

L'espace de tous les buts possibles (tous les arbres possibles) est défini au travers d'une ontologie du domaine, dont un exemple est donné dans le tableau 3.1. Dans cette ontologie sont listés l'ensemble des instances possibles pour chaque classe abstraite (ligne avec le symbole →). Comme on peut le voir, la classe abstraite*venue* peut se décliner sous les deux formes suivantes *bar* et *restaurant* selon une certaine probabilité a priori (donnée entre crochet). Selon l'instance effectivement mentionnée par l'utilisateur au cours de l'interaction on peut voir que ce ne sont pas les mêmes concepts qui vont être mis en jeu, par exemple le concept *food* est ici spécifique à l'instance *restaurant* de la classe abstraite *venue*. De même dans cette ontologie sont identifiées l'ensemble des valeurs atomiques pouvant être assignées à un même concept (ligne avec le =).

Nous donnons dans le tableau C.2 les ontologies employées pour les tâches *TownInfo* et *MaRDi*.

C.2 Ontologies *DSTC2* et *DSTC3*

Dans les tâches *DSTC2* et *DSTC3*, l'ontologie du domaine se présente sous la forme d'un formulaire. Ce dernier identifie uniquement les concepts et leurs valeurs associés mais ne modélise aucune relation de dépendance entre ces concepts. Ainsi, se distingue ceux dont l'utilisateur peut spécifier une valeur particulière en guise de contrainte sur leur recherche de ceux pour lesquels il n'est pas autorisé à le faire. Par exemple, l'utilisateur peut fournir une valeur pour le concept *pricerange*, mais cela ne lui est pas autorisé pour le concept *phone* (numéro de téléphone), ce qui traduit simplement le fait que système ne peut pas procéder à une recherche d'établissement par le numéro de téléphone.

Le tableau C.3 établit la liste complète des concepts manipulés dans les tâches *DSTC2* et *DSTC3* et indique si oui ou non l'utilisateur est autorisé à en indiquer la valeur et si oui combien de valeurs sont prises en compte par le système. Dans ces deux tâches, l'utilisateur est autorisé à demander la valeur pour tous les concepts manipulés par le système. Par exemple, pour un restaurant donné, l'utilisateur peut demander au système son numéro du téléphone ou encore sa gamme de prix (valeur du concept *pricerange*).

```
task -> find (entity){1.0} ;
entity -> venue(+type,+area, near, -name, -addr, -phone, -comment){1.0} ;
type -> restaurant(+food, +pricerange, -price, music, drinks, stars){0.33} ;
type -> hotel(+pricerange, stars, -price, -drinks ){0.33} ;
type -> bar(+drinks, music, pricerange){0.33} ;
type -> amenity(){0.01} ;
area = ("central" | "east" | "north" | "riverside" | "south" | "west" ) ;
near = ("Castle" | "Cinema" | "Fountain" | "Main Square" | "Museum" | ...) ;
name = ("Alexander Hotel" | "Art House Hotel" | .... ) ;
food = ("Chinese" | "English" | "French" | "Indian" | "Italian" | "Russian" | "fish" | "snacks" ) ;
pricerange = ("cheap" | "expensive" | "moderate") ;
music = ("Classical" | "Ethnic" | "Folk" | "Jazz" | "Pop" | "Rock") ;
drinks = ("beer" | "cocktails" | "soft drinks" | "wine") ;
stars = ("1" | "2" | "3" | "4" | "5") ;
addr = () ;
phone = () ;
price = () ;
comment = () ;
```

(a)

```
task -> execute(cmd){1.0} ;
cmd -> manipaction(action, object){1.0} ;
action -> give(){0.5} ;
action -> move(location){0.5} ;
object -> domestic(idobj, type, color, location){1.0} ;
type -> book(title, genre, author){0.3} ;
type -> mug(){0.3} ;
type -> tape(title, genre, director){0.3} ;
type -> box(){0.1} ;
idobj = ("BLUE_BOOK" | "RED_BOOK" | ...)
color = ( blue | red | ....)
location = ( livingroom_coffeetable | livingroom_bedsidetable | ....)
book.title = ( "the lord of the rings 1" | ...)
tape.title = ("very bad trip" | ...)
author = ("J.R.R Tolkien" | ...)
director = ("Todd Phillips" | ...)
genre = ("scifi" | ...)
```

(b)

TABLE *C.2* – *Ontologie des tâche TownInfo (a) MaRDI (b).*

Concepts	Tailles	Valeurs que peut spécifier l'utilisateur
area	5	north, south, east, central, south
food	91	french, english, snacks, etc.
name	113	pizza hut, etc.
pricerange	3	cheap, moderate, expensive
addr	0	-
phone	0	-
postcode	0	-
signature	0	-

(a)

Concepts	Tailles	Valeurs que peut spécifier l'utilisateur
area	15	chesterton, etc.
food	28	french, english, snacks, etc.
name	163	burger king, etc.
pricerange	4	free, cheap, moderate, expensive
near	52	queens college, etc.
type	3	restaurant, pub, coffeeshop
children allowed	2	yes, no
has tv	2	yes, no
has internet	2	yes, no
addr	0	-
phone	0	-
postcode	0	-
price	0	-

(b)

TABLE *C.3* – *Ontologies des tâches DSTC2 (a) et DSTC3 (b).*