



HAL
open science

Apprentissage par renforcement pour la généralisation des approches automatiques dans la conception des systèmes de dialogue oral

Florian Pinault

► **To cite this version:**

Florian Pinault. Apprentissage par renforcement pour la généralisation des approches automatiques dans la conception des systèmes de dialogue oral. Autre [cs.OH]. Université d'Avignon, 2011. Français. NNT : 2011AVIG0188 . tel-00933937

HAL Id: tel-00933937

<https://theses.hal.science/tel-00933937v1>

Submitted on 24 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Apprentissage par renforcement pour la généralisation des approches automatiques dans la conception des systèmes de dialogue oral

Florian Pinault

► **To cite this version:**

Florian Pinault. Apprentissage par renforcement pour la généralisation des approches automatiques dans la conception des systèmes de dialogue oral. Autre [cs.OH]. Université d'Avignon, 2011. Français. .

HAL Id: tel-00933937

<https://tel.archives-ouvertes.fr/tel-00933937>

Submitted on 21 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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 Agrosociétés »
Laboratoire d'Informatique d'Avignon (EA 4128)

Apprentissage par renforcement
pour la généralisation des approches automatiques
dans la conception des systèmes de dialogue oral

par

Florian PINAULT

Soutenue publiquement le 24 novembre 2011 devant un jury composé de :

M.	François Charpillet	Professeur, LORIA, Nancy	Rapporteur
M.	Olivier Pietquin	Professeur, Supélec, Metz	Rapporteur
M.	Romain Laroche	Docteur, Orange Labs	Examineur
M.	Georges Linarès	Professeur, CERI, Avignon	Examineur
M.	Blaise Thomson	Docteur, Université de Cambridge, UK	Examineur
M.	Fabrice Lefèvre	Professeur, CERI, Avignon	Directeur de thèse

Remerciements

Avant tout merci à mes directeurs de thèse, Fabrice Lefèvre qui m'a fait confiance tout au long de cette thèse. Merci pour les discussions approfondies, pour sa disponibilité et son écoute, j'ai pu développer mes idées tout en profitant de son expérience et de son soutien permanent. Merci à Renato De Mori, en particulier pour les clarifications apportées sur les articles et les bibliographies dont j'ai eu besoin.

Merci à Florian Verdet et Alain Baeckeroot qui ont permis, entre autres, au mathématicien que j'étais de passer du côté obscur, de décrypter perl, python, ruby et bash et de maîtriser le silicium.

Je tiens également à remercier les membres du jury qui ont jugé ce travail et émis des remarques constructives.

Merci aussi à tous ceux qui m'ont permis de trouver la voie, par leur exemple ou leurs conseils, mes parents, ma sœur Laure, Marc Humbert, Lionel Aufranc, Sandra Rozensztajn, Francesco Colonna, C., Y. et N., et bien d'autres.

Enfin, merci à Sabine sans qui ce manuscrit n'aurait pas pu être achevé, pour ses relectures et ses conseils de rédaction, son soutien constant.

Résumé

Les systèmes de dialogue homme-machine actuellement utilisés dans l'industrie sont fortement limités par une forme de communication très rigide imposant à l'utilisateur de suivre la logique du concepteur du système. Cette limitation est en partie due à leur représentation de l'état de dialogue sous la forme de formulaires préétablis.

Pour répondre à cette difficulté, nous proposons d'utiliser une représentation sémantique à structure plus riche et flexible visant à permettre à l'utilisateur de formuler librement sa demande.

Une deuxième difficulté qui handicape grandement les systèmes de dialogue est le fort taux d'erreur du système de reconnaissance vocale. Afin de traiter ces erreurs de manière quantitative, la volonté de réaliser une planification de stratégie de dialogue en milieu incertain a conduit à utiliser des méthodes d'apprentissage par renforcement telles que les processus de décision de Markov partiellement observables (POMDP). Mais un inconvénient du paradigme POMDP est sa trop grande complexité algorithmique. Certaines propositions récentes permettent de réduire la complexité du modèle. Mais elles utilisent une représentation en formulaire et ne peuvent être appliqués directement à la représentation sémantique riche que nous proposons d'utiliser.

Afin d'appliquer le modèle POMDP dans un système dont le modèle sémantique est complexe, nous proposons une nouvelle façon de contrôler sa complexité en introduisant un nouveau paradigme : le POMDP résumé à double suivi de la croyance. Dans notre proposition, le POMDP maître, complexe, est transformé en un POMDP résumé, plus simple. Un premier suivi de croyance (*belief update*) est réalisé dans l'espace maître (en intégrant des observations probabilistes sous forme de listes *n-best*). Et un second suivi de croyance est réalisé dans l'espace résumé, les stratégies obtenues sont ainsi optimisées sur un véritable POMDP.

Nous proposons deux méthodes pour définir la projection du POMDP maître en un POMDP résumé : par des règles manuelles et par regroupement automatique par *k* plus proches voisins. Pour cette dernière, nous proposons d'utiliser la distance d'édition entre graphes, que nous généralisons pour obtenir une distance entre listes *n-best*.

En outre, le couplage entre un système résumé, reposant sur un modèle statistique par POMDP, et un système expert, reposant sur des règles *ad hoc*, fournit un meilleur contrôle sur la stratégie finale. Ce manque de contrôle est en effet une des faiblesses empêchant l'adoption des POMDP pour le dialogue dans l'industrie.

Dans le domaine du renseignement d'informations touristiques et de la réservation de chambres d'hôtel, les résultats sur des dialogues simulés montrent l'efficacité de l'approche par renforcement associée à un système de règles pour s'adapter à un environnement bruité. Les tests réels sur des utilisateurs humains montrent qu'un système optimisé par renforcement obtient cependant de meilleures performances sur le critère pour lequel il a été optimisé.

Table des matières

INTRODUCTION.....	7
CHAPITRE 1: ÉTAT DE L'ART.....	15
1.1 Contexte.....	18
1.1.1 Système de dialogue oral : généralités.....	19
1.1.2 Compréhension de la parole.....	20
1.1.3 Gestion du dialogue.....	29
1.1.4 Interface entre gestion du dialogue et compréhension.....	35
1.2 Apprentissage par renforcement.....	37
1.2.1 Processus de Décision de Markov (MDP).....	38
1.2.2 Processus de Décision de Markov Partiellement Observables (POMDP).....	39
1.2.3 Stratégie optimale.....	41
1.2.4 Représentation de la stratégie.....	42
1.2.5 Algorithmes de résolution.....	43
1.2.6 Résolution approchée par espaces résumés.....	43
1.3 Application au dialogue oral.....	48
1.3.1 MDP pour le dialogue.....	49
1.3.2 Structure d'un POMDP pour le dialogue.....	50
1.3.3 Verrous technologiques.....	54
1.3.4 Solutions proposées.....	55
1.4 Simulation.....	65
1.4.1 Niveau de granularité de la simulation.....	65
1.4.2 Simulation de l'utilisateur.....	66
1.4.3 Simulation des erreurs de compréhension.....	68
1.4.4 Simulation du reste de l'environnement.....	69
1.5 Évaluation.....	70
1.5.1 Simulateurs.....	71
1.5.2 Utilisateurs humains.....	71
1.5.3 Évaluation Paradise.....	73
CHAPITRE 2: CONTRIBUTIONS.....	75
2.1 Module de compréhension et état de dialogue non contraints.....	78
2.1.1 Modèle sémantique : graphes mémoire de frames.....	79
2.1.2 Agrégation de graphes de frames sémantiques.....	80
2.1.3 Accès non contraint à la base de données.....	81
2.2 Résumé d'action système.....	83
2.2.1 Création d'actions maitres.....	83
2.2.2 Sélection de l'action maitre.....	86
2.2.3 Ordre de la génération et de la sélection.....	87
2.2.4 Complétude de la conception (VUID-completeness).....	87
2.3 Paradigme de POMDP résumé : Réduction de l'espace d'état et de croyance...88	
2.3.1 Motivation - Principe.....	89
2.3.2 Espaces maitres.....	90
2.3.3 Espaces résumés.....	92
2.3.4 Prise de décision.....	93
2.3.5 Exemple.....	93
2.4 Conception de fonctions résumées.....	97
2.4.1 Résumés ad-hoc.....	97
2.4.2 Résumés automatiques par clustering.....	99

2.5 Optimisation de stratégie POMDP.....	108
2.5.1 Apprentissage par renforcement sur modèle complet.....	108
2.5.2 Simulations d'utilisateurs avec modèle de frames sémantiques.....	110
2.5.3 Principe de simulations itératives.....	112
CHAPITRE 3: APPLICATION ET RÉSULTATS.....	115
3.1 Méthodologie d'évaluation.....	118
3.1.1 Tâches.....	119
3.1.2 Critères objectifs	121
3.1.3 Critères subjectifs.....	123
3.1.4 Synthèse.....	124
3.2 Étude préliminaire dans les espaces résumés.....	125
3.2.1 Système.....	125
3.2.2 Optimisation de stratégie de dialogue.....	126
3.2.3 Évaluation.....	126
3.2.4 Résultats.....	127
3.2.5 Conclusion et limites.....	128
3.3 Implémentation d'un système couplé avec un POMDP résumé.....	130
3.3.1 Architecture d'un prototype.....	130
3.3.2 Modèle sémantique : mise à jour d'un graphe sémantique.....	132
3.3.3 Modèle sémantique d'ancrage.....	133
3.3.4 Passage aux espaces résumés.....	136
3.3.5 Retour à l'espace maître.....	137
3.3.6 Couplage entre règles et POMDP.....	137
3.4 Optimisation de stratégie POMDP par simulations.....	138
3.4.1 Mise en œuvre des simulations.....	138
3.4.2 Entraînement du POMDP résumé.....	141
3.4.3 Suivi de croyance maître.....	143
3.4.4 Comparaison à un MDP.....	144
3.4.5 POMDP résumés par clustering automatique.....	147
3.4.6 Couplage de systèmes maître-résumé.....	149
3.4.7 Limites.....	150
3.5 Tests utilisateurs	152
3.5.1 Mise en œuvre.....	152
3.5.2 Évaluation objective.....	154
3.5.3 Évaluation subjective.....	156
3.5.4 Conclusion et limites	162
CONCLUSION ET PERSPECTIVES.....	165
ANNEXES.....	171
Annexe A : Détails d'implémentation du prototype.....	172
Annexe B : Protocole expérimental.....	175
Annexe C : Démonstrations mathématiques.....	178
INDEX DES ILLUSTRATIONS.....	182
INDEX DES TABLES.....	184
BIBLIOGRAPHIE.....	186

Introduction

Dans un monde où la technologie tient une part de plus en plus importante, les interactions avec les différentes machines qui nous entourent ne passent plus seulement par le clavier et l'écran. Le développement d'internet et la multiplication des téléphones portables ont rendu possible l'accès à tout moment et en tout lieu à des informations extrêmement variées. La situation d'interaction avec la machine s'est diversifiée, et de la position statique d'un travailleur de bureau, l'utilisateur est devenu nomade. S'adaptant à ce changement, les interfaces sont devenues multimodales, combinant les différents canaux de communication visuelle et auditive lors de l'utilisation de GPS par exemple. La conception d'écrans tactiles a permis d'utiliser également le sens du toucher pour interagir avec la machine.

L'échange verbal étant le mode de communication privilégié entre les êtres humains, l'utilisation de la voix comme canal d'interaction entre l'homme et la machine apparaît de prime abord comme un choix naturel.

En effet, l'approche la plus évidente pour automatiser une tâche réalisée par un être humain (de la consultation d'un annuaire à l'enseignement de notions complexes [Forbes-Riley et Litman 2011], en passant par la gestion des réservations de billets d'avion (ATIS¹) [Price 1990] ou comme dans les travaux de cette thèse du renseignement d'informations touristiques et de la réservation de chambres d'hôtel) semble être de copier le comportement humain, y compris son mode de communication oral.

Cependant, force est de constater que les systèmes de dialogue oral homme-machine ne sont encore que très peu employés. En effet, avec la généralisation d'internet, une nouvelle technologie a vu le jour et le remplissage de formulaires web s'avère très répandu car plus facile à mettre en place sur les machines que ne l'est la communication orale. Cette technologie est plus efficace car il est souvent plus rapide d'utiliser le canal visuel que l'auditif.

Dans certaines situations pourtant, le dialogue oral est nécessaire, voire indispensable. C'est évidemment le cas pour la gestion d'un standard téléphonique, mais aussi dans le

(1) ATIS : *Air Traveler Information System*.

cadre de la conduite automobile afin de minimiser les erreurs dues à l'inattention du conducteur. De plus, selon une enquête du CREDOC¹ (servant de référence à une étude de l'INSEE) sur les conditions de vie et aspirations des français, « en 2005, 71% des 18-24 ans disposent d'au moins un micro-ordinateur à leur domicile, contre 25% des sexagénaires » [Bigot 2005]. Les personnes âgées, souffrant parfois de troubles visuels, peuvent être amenées à utiliser plus facilement les nouvelles technologies à travers l'utilisation de la voix, ce mode d'interaction conduisant de plus à une interaction moins impersonnelle. Cette catégorie d'utilisateurs pourrait donc accéder à l'outil informatique reposant sur ce type d'interface.

Si chacun reconnaît l'importance d'un système de dialogue homme-machine et ses possibilités pour le quotidien des utilisateurs, il faut bien reconnaître que ce type de communication verbale n'en est encore qu'à ses prémises. Le dialogue fluide et naturel entre un homme et une machine se heurte encore à de nombreuses difficultés.

Historiquement, ce dialogue homme-machine est étroitement lié à la recherche en intelligence artificielle qui accorde encore aujourd'hui une place capitale au test de Turing [Turing 1950] reposant essentiellement sur un échange verbal. Ce test consiste, pour un système automatique, à tenter de se faire passer pour un humain lors d'une interaction donnée avec un véritable être humain, généralement un expert en intelligence artificielle qui fait office de juge. Ce test est souvent considéré comme la tâche capable de prouver que l'intelligence artificielle est comparable à celle d'un humain et ne se limite pas à la simple répétition d'un mécanisme prédéfini.

Rapidement, des propositions ont été faites pour réussir le test de Turing. Ainsi, le système de conversation Eliza [Weizenbaum 1966] se fonde sur des modèles de phrases qui réutilisent les informations fournies par l'interlocuteur. Les résultats paraissent tout d'abord impressionnants puisque la machine reconnaît les concepts clés mais elle n'a pas accès au sens de l'information émise et produit des réponses mécaniques. Ce paradigme de système de dialogue est donc difficilement exploitable dans des situations concrètes.

Pour dépasser alors ce type de systèmes qui atteignent vite leurs limites, il a été nécessaire d'élaborer des mécanismes beaucoup plus complexes reposant sur la modélisation commune du système de dialogue oral décomposé généralement selon l'architecture suivante :

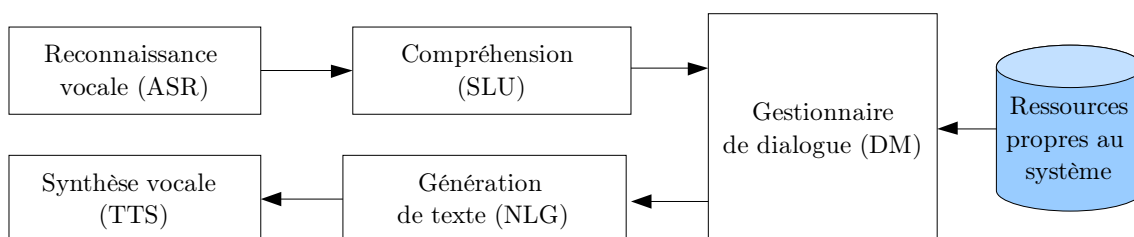


Figure 1: Décomposition d'un système de dialogue en ses différents modules.

Le signal acoustique reçu par le microphone et provenant de l'utilisateur est traité par le module de reconnaissance vocale (ASR²) qui le transforme en mots. Ceux-ci permettent au

(1) CREDOC : Centre de recherche pour l'étude et l'observation des conditions de vie des français.

(2) ASR : *Automatic Speech Recognition*.

module de compréhension (SLU¹) d'élaborer une représentation symbolique de l'intention de l'utilisateur. Le gestionnaire de dialogue (DM²) l'utilise pour définir une réponse appropriée. Celle-ci sera mise en forme grâce aux modules de génération de texte (NLG³) et de synthèse vocale (TTS⁴).

À travers cette modélisation, les chercheurs ont élaboré des réponses complexes comme la constitution d'un ensemble de règles logiques mimant le raisonnement humain. Ces règles permettent idéalement à la machine de produire des connaissances nouvelles à partir des informations, propres au dialogue, fournies par l'interlocuteur et d'une base de connaissance (KB⁵) intégrée au système.

Une autre réponse aux limitations des systèmes de conversation tel que celui d'Eliza consiste à s'atteler à la gestion de l'historique du dialogue et au statut du contenu propositionnel de l'échange : c'est le modèle d'ancrage (*grounding*) proposé par [Traum 1994]. Ce modèle d'ancrage est employé dans la proposition de suivi de l'état d'information (ISU) [Traum et Larsson 2003] qui est à la base des systèmes DIPPER [Bos et al. 2003] et HIS [Young et al. 2010].

Que ce soit les règles logiques ou le modèle d'ancrage, deux difficultés majeures empêchent leur utilisation : d'une part, le manque de robustesse face aux erreurs de reconnaissance vocale, et d'autre part l'insuffisance de flexibilité du système.

Concernant le manque de robustesse, les techniques de reconnaissance vocale, essentielles dans le cadre d'un système de dialogue homme-machine ont réalisé de grands progrès grâce à l'utilisation de méthodes statistiques (chaines de Markov cachées en particulier). Ainsi, le traitement de la parole préparée (celle des journaux télévisés et des discours politiques) est devenu très efficace avec un taux d'erreur mot (WER) inférieur à 10 %. En revanche, la parole spontanée utilisée lors d'un dialogue contient beaucoup de disfluences telles que les répétitions, les mots de remplissage (*fillers*), les phrases inachevées ou reprises, etc. Cette forte agrammaticalité conduit à des taux d'erreur de l'ordre de 30 %.

Les approches de conception de systèmes de dialogue sont généralement déterministes : ils ne font pas intervenir de traitements statistiques. Certaines, fondées sur différents principes linguistiques, proposent d'utiliser des représentations sémantiques riches et flexibles pour modéliser les différentes facettes du processus de dialogue homme-machine. Elles reposent finalement sur un enchaînement d'implications logiques particulièrement sensibles aux erreurs de reconnaissance vocale : une erreur de compréhension se propage par un effet de dominos et devient impossible à corriger.

Les systèmes actuellement utilisés dans l'industrie rencontrent également ce problème de robustesse mais le contournent en implémentant des mécanismes *ad hoc*, ce qui entraîne des choix non-optimaux et des coûts élevés de conception et de maintenance. De plus, l'industrie emploie un cadre théorique rigide à travers le langage VoiceXML [VoiceXML 2004]. Cette rigidité est de deux types. Tout d'abord, la rigidité du gestionnaire de dialogue (DM)

(1) SLU : *Spoken Language Understanding*.

(2) DM : *Dialogue Manager*.

(3) NLG : *Natural Language Generation*.

(4) TTS : *Text-To-Speech*.

(5) KB : *Knowledge Base*.

est induite par un plan prédéfini par le concepteur, elle conduit l'utilisateur à s'adapter à une logique qui n'est pas la sienne. Deuxièmement, la rigidité sémantique du module de compréhension (SLU) est due à l'emploi d'une représentation préétablie sous la forme d'un formulaire constitué d'une liste de couples champ-valeur. Il est alors impossible pour la machine de traiter des situations impliquant des énoncés complexes et ces systèmes se limitent à des tâches simples telles que l'accès à une base de données.

Visant à palier la rigidité du gestionnaire de dialogue évoquée ci-dessus, il a été proposé d'intégrer dans le système un processus de planification. Cela consiste à supposer que l'utilisateur agit selon un plan en vue de réaliser son objectif. Ce plan fait office de dialogue idéal qui, s'il est suivi par le système, conduit à un dialogue parfait du point de vue de l'utilisateur : la tâche commune est réussie et la satisfaction de l'utilisateur est totale. Les systèmes conçus selon les principes de planification cherchent à identifier le plan de l'utilisateur pour s'y conformer. Il s'adapte à l'utilisateur et c'est ce qui leur confère une plus grande flexibilité.

En fait, les systèmes réalisant une planification modélisent le cours futur du dialogue et tentent de prévoir son évolution. Mais ils ne reposent pas sur l'optimisation d'un critère quantitatif, et ne sont donc pas complètement automatisés. Ainsi pour réduire les coûts de conception et de maintenance, les méthodes statistiques issues du domaine de l'apprentissage par renforcement sont employées et donnent un fondement mathématique à la planification : c'est le cas des modèles MDP¹ qui utilisent des chaînes de Markov conditionnées par les actions du système. Les POMDP² y ajoutent de la robustesse en utilisant des variables d'états cachées et modélisent quantitativement l'incertitude due aux erreurs de reconnaissance vocale.

L'intérêt de ces modèles (POMDP) issus de l'apprentissage par renforcement est double. Ils permettent de gérer l'incertitude de façon quantitative et offrent un ensemble de techniques adaptées à la nature interactive du dialogue.

Cependant, ces méthodes statistiques requièrent de grandes quantités de données afin de rendre les modèles appris plus proches du modèle exact et les estimateurs plus précis. De plus, ces techniques statistiques impliquent l'utilisation d'algorithmes dont la complexité grandit exponentiellement avec la taille de la représentation sémantique.

Finalement, plus le modèle sémantique est complexe, moins les méthodes statistiques sont applicables du fait des limitations pratiques (corpus disponibles, puissance de calcul). Ceci implique la nécessité d'utiliser une représentation sémantique structurée de façon particulièrement simple, ce qui amène, dans la plupart des cas, le recours à un formulaire constitué d'une liste préétablie de couples champ-valeur. Ainsi, les propositions d'utiliser des méthodes d'apprentissage par renforcement semblent prisonnières de la rigidité propre au modèle de formulaire et apparaissent comme difficilement applicables pour des tâches nécessitant une plus grande flexibilité.

Pourtant, malgré ces difficultés, nous pensons qu'une représentation sémantique riche peut convenir à l'utilisation de systèmes de dialogue oral pour des tâches plus complexes tout en conservant le bénéfice de la robustesse offerte par les méthodes statistiques.

(1) MDP : *Markov Decision Process*.

(2) POMDP: *Partially Observable Markov Decision Process*.

Contributions

Nous nous intéressons dans cette thèse à concilier deux tendances antagonistes s’opposant sur le degré de complexité de la représentation sémantique à utiliser : d’un côté, plus la représentation sémantique est riche, plus la représentation symbolique de l’intention de l’utilisateur peut être précise et intégrer les nuances d’un dialogue humain. Cette flexibilité du modèle sémantique nécessite un modèle sémantique complexe.

Au contraire, plus la structure de la représentation sémantique est simple, mieux les méthodes statistiques peuvent être appliquées : la simplification du modèle sémantique permet d’appliquer des algorithmes de résolution en un temps raisonnable et avec des quantités de données d’apprentissage acceptables.

Nous proposons donc un paradigme de système de dialogue doté d’une part d’une modélisation riche de la sémantique du dialogue, d’autre part de modèles statistiques de renforcement qui sont robustes aux erreurs ASR et permettent la planification en milieu incertain. Il s’agit du paradigme que nous appelons « POMDP résumé à double suivi de croyance ». Nous reprenons les propositions de [Young et al. 2010] concernant l’utilisation de POMDP pour le dialogue afin de contrôler l’explosion combinatoire mais nous avons choisi de tirer avantage des possibilités du modèle POMDP tant dans l’espace maître que dans l’espace résumé en effectuant un double suivi de croyance. La stratégie résumée optimisée l’est donc sur un véritable modèle POMDP. Notre première contribution consiste à utiliser un modèle sémantique riche, en restant dans le cadre POMDP, tout en intégrant un traitement approprié des scores de confiance *n-best* issus du SLU.

Cependant, les systèmes POMDP résumés ne sont pas parfaits et manifestent parfois un comportement aberrant à cause de l’insuffisance d’information due au passage dans des espaces résumés. Pour résoudre cette difficulté, notre deuxième contribution vise à proposer un couplage entre un système résumé, stochastique, reposant sur un POMDP et un système maître, classique, reposant sur un ensemble de règles logiques.

Pour des raisons de coûts, il est souhaitable de pouvoir automatiser le processus de conception des systèmes de dialogue et de permettre leur adaptation automatique à de nouvelles situations. Dans cette optique, notre troisième contribution consiste à proposer d’appliquer le paradigme de POMDP résumé à double suivi de croyance en définissant les fonctions résumées par *clustering* de graphes et de listes de graphes sémantiques.

Bien sûr, nous ne nous limitons pas aux seules propositions théoriques et nous développons un prototype de système de dialogue permettant de tester ces idées de façon pratique, il s’agit de notre quatrième contribution. Nous présentons ainsi une évaluation à la fois par des simulations et par des utilisateurs réels. Les expériences réalisées valident l’approche de POMDP résumé pour le dialogue.

Organisation du document

Le chapitre 1 vise à cerner le contexte des travaux effectués sur les modèles sémantiques utilisés pour le dialogue, sur l’apprentissage par renforcement, et sur leur utilisation dans des systèmes de dialogue afin d’expliquer le cadre théorique de nos contributions. Cet état de l’art s’étend également aux méthodes utilisées dans la littérature pour simuler un utilisateur et pour évaluer un système de dialogue que nous employons pour l’optimisation de la stratégie de notre prototype.

Le chapitre 2 détaille le paradigme de POMDP résumé à double suivi de croyance permettant la conception de systèmes de dialogue avec un apprentissage par renforcement et une représentation sémantique riche. Il décrit l'essentiel de nos contributions théoriques.

Le chapitre 3 développe une application du paradigme de POMDP résumé à double suivi de croyance ainsi que les résultats obtenus par le prototype sur des tests réalisés avec des utilisateurs simulés et des utilisateurs réels pour des tâches de réservation de chambres d'hôtel.

En annexe figurent les détails d'implémentations, du protocole expérimental ainsi que quelques compléments mathématiques.

Publications associées

Les publications associées à ces travaux de thèse sont les suivantes :

F. Pinault (2008). *Approche probabiliste pour la gestion du dialogue homme-machine*, Journée des doctorants de l'école doctorale I2S ([DOCTISS 2008](#)) .

F. Pinault (2009). *Modélisation stochastique du dialogue par structures sémantiques*, Rencontres Jeunes Chercheurs en Parole ([RJCP 2009](#)).

F. Pinault, F. Lefèvre et R. De Mori (2009). *Feature-based summary spaces for stochastic dialogue modeling with hierarchical semantic frames*, Proceedings of 10th Annual Conference of the International Speech Communication Association ([Interspeech 2009](#)).

F. Pinault et F. Lefèvre (2010). *Structures de frames sémantiques pour le dialogue homme-machine par processus de décision markoviens*, Journée d'étude de la parole ([JEP 2010](#)).

F. Pinault et F. Lefèvre (2011). *Semantic graph clustering for POMDP-based spoken dialog systems*, Proceedings of 12th Annual Conference of the International Speech Communication Association ([Interspeech 2011](#)).

F. Pinault et F. Lefèvre (2011). *Unsupervised clustering of probability distributions of semantic frame graphs for POMDP-based spoken dialogue systems with summary space*, 7th Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI/[KRPDS 2011](#)).

Chapitre 1 : État de l'art

Sommaire

1.1 Contexte.....	18
1.1.1 Système de dialogue oral : généralités.....	19
1.1.2 Compréhension de la parole.....	20
1.1.3 Gestion du dialogue.....	29
1.1.4 Interface entre gestion du dialogue et compréhension.....	35
1.2 Apprentissage par renforcement.....	37
1.2.1 Processus de Décision de Markov (MDP).....	38
1.2.2 Processus de Décision de Markov Partiellement Observables (POMDP).....	39
1.2.3 Stratégie optimale.....	41
1.2.4 Représentation de la stratégie.....	42
1.2.5 Algorithmes de résolution.....	43
1.2.6 Résolution approchée par espaces résumés.....	43
1.3 Application au dialogue oral.....	48
1.3.1 MDP pour le dialogue.....	49
1.3.2 Structure d'un POMDP pour le dialogue.....	50
1.3.3 Verrous technologiques.....	54
1.3.4 Solutions proposées.....	55
1.4 Simulation.....	65
1.4.1 Niveau de granularité de la simulation.....	65
1.4.2 Simulation de l'utilisateur.....	66
1.4.3 Simulation des erreurs de compréhension.....	68
1.4.4 Simulation du reste de l'environnement.....	69
1.5 Évaluation.....	70
1.5.1 Simulateurs.....	71
1.5.2 Utilisateurs humains.....	71
1.5.3 Évaluation Paradise.....	73

Le but de ce chapitre consacré à l'état de l'art de la recherche sur les systèmes de dialogue oral homme-machine est de situer nos travaux dans le contexte qui leur a permis de voir le jour à partir des méthodes actuellement explorées et de leurs éventuelles limitations. Nous tâchons ainsi de mettre en avant quelques propositions pour y répondre, qui ont orienté une partie de cette recherche.

La partie 1.1 présente le contexte des travaux de cette thèse et situe le rôle du gestionnaire de dialogue (DM) par rapport aux autres composants du système de dialogue. En particulier, on s'intéresse à sa relation au modèle sémantique utilisé dans le module de compréhension (SLU).

On trouvera en partie 1.2 une description générale des méthodes de renforcement (MDP et POMDP) ainsi que de certains algorithmes de résolution exacts ou approximatifs.

Leur application actuelle au dialogue est décrite en partie 1.3, mettant en relief leur limitation concernant la flexibilité de la représentation sémantique.

Une description des environnements de simulation de dialogue utilisés pour entraîner les systèmes ou les évaluer se trouve en partie 1.4.

Enfin, les méthodes d'évaluations de systèmes de dialogue proposées dans la littérature sont présentées en partie 1.5.

1.1 Contexte

Résumé : Le gestionnaire de dialogue et le module de compréhension sont deux des composants constituant un système de dialogue. Le gestionnaire de dialogue est très dépendant du modèle sémantique utilisé par le module de compréhension. Si celui-ci est trop simple (par exemple un formulaire pré-établi), il impose une structure rigide au système. S'il est trop complexe, le processus de prise de décision est plus élaboré et il s'avère difficile d'appliquer des méthodes statistiques. La conception de l'interface entre ces deux composants est délicate et conditionne les performances d'un système de dialogue oral homme-machine.

Cette partie présente le contexte des travaux de cette thèse et situe notamment le rôle du gestionnaire de dialogue sur lequel porte cette thèse par rapport aux autres composants du système de dialogue. En particulier, on s'intéresse à sa relation au modèle sémantique utilisé dans le module de compréhension car un des enjeux de nos travaux est précisément d'envisager d'utiliser un modèle sémantique plus complexe que dans les systèmes de l'état de l'art.

On présente dans cette partie le contexte de la recherche actuelle sur les deux modules concernés :

- Compréhension : axée sur une représentation sémantique riche avec la présentation des modèles sémantiques utilisés pour le dialogue oral homme-machine.
- Gestion du dialogue : présentation des systèmes de dialogue classiques non statistiques. Étant donné la place essentielle donnée aux systèmes statistiques (MDP et POMDP), ces derniers seront développés dans des parties spécifiques (partie 1.2 pour leur aspect général et 1.3 pour leur application au dialogue).

Un système de dialogue repose en premier lieu sur un modèle sémantique donné. C'est ce modèle qui permet d'exprimer le contenu sémantique des énoncés du dialogue et c'est donc par ce modèle que le système sera capable d'appréhender la réalité.

Ensuite, on utilise un modèle de dialogue adapté au modèle sémantique pour réaliser le suivi du dialogue au cours de son déroulement. C'est dans ce modèle de dialogue que la stratégie est définie manuellement ou optimisée par des méthodes statistiques.

Les possibilités et les limitations du modèle sémantique influent directement sur celle du système de dialogue final. Différents modèles sémantiques ont été proposés dans la littérature. Certains, issus du domaine de la linguistique, sont très puissants pour représenter les nuances du dialogue humain mais sont complexes voire impossibles à mettre en œuvre dans

un système réel. D'autres modèles, visant à répondre au besoin d'une tâche donnée, sont plus directement applicables mais impliquent certaines contraintes fortes pour le dialogue homme-machine (par exemple les modèles de formulaires préétablis).

En somme, selon la tâche envisagée et le niveau d'initiative que l'on souhaite donner à l'utilisateur, il est possible de choisir des modèles sémantiques plus ou moins contraints. Il s'agit donc de se choisir un système de dialogue adapté. [Dahlbäck 1997] propose une taxonomie intéressante, quoique déjà ancienne, des systèmes de dialogue. On pourra s'y référer en ce qui concerne l'influence du type de tâche considéré sur le choix du paradigme de conception d'un système de dialogue.

1.1.1 Système de dialogue oral : généralités

Modèle classique

Comme illustré par la figure 1.1, les systèmes de dialogue actuellement étudiés dans le monde académique se décomposent en différents modules. Les uns sont des modules d'entrée et visent à extraire de l'information à partir d'observations de l'environnement extérieur. D'autres ont pour fonction de transformer l'action du système en un signal correctement formaté vers l'extérieur. A la jointure entre les deux se situe le gestionnaire de dialogue qui prend la décision de l'action à suivre.

L'étape d'entrée d'information est constituée de deux modules. Le module de reconnaissance vocale (ASR¹) qui transforme un signal acoustique en une transcription textuelle. Et le module de compréhension (SLU²) qui élabore une représentation abstraite des informations disponibles sous une forme utilisable par le gestionnaire de dialogue.

L'étape de sortie est elle aussi constituée de deux modules : le générateur de langage naturel (NLG³) crée du texte et le synthétiseur de voix (TTS⁴) le transforme en signal acoustique.

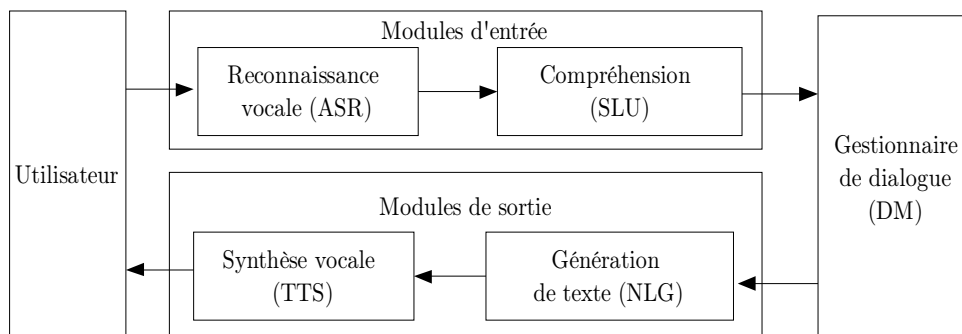


Figure 1.1: Boucle de fonctionnement d'un système de dialogue homme-machine.

- (1) ASR : *Acoustic Signal Recognition*.
- (2) SLU : *Speech Language Understanding*.
- (3) NLG : *Natural Language Generation*.
- (4) TTS : *Text To Speech*.

Il existe toutefois un handicap majeur pour les systèmes de dialogue : le fort taux d'erreur du module de reconnaissance vocale (ASR) qui reste une véritable gageure technologique à l'heure actuelle. De grands progrès ont été réalisés pour obtenir des transcriptions sans erreur et le taux d'erreur sur des corpus de parole préparée (celle des journaux télévisés et des discours politiques) est devenu très faible, inférieur à 10 %. Cependant, en ce qui concerne la parole spontanée, les phénomènes de disfluences propres à des conditions moins rigides d'élaboration du langage conduisent à une agrammaticalité forte, à une élocution moins soignée sans oublier qu'ils sont souvent associés à des contextes de bruit peu favorables. Dans ce cadre, des taux d'erreur de l'ordre de 30 % sont couramment observés.

Méthodes statistiques (*Fat pipeline*)

Pour obtenir des systèmes plus robustes aux erreurs des modules d'entrée (ASR et SLU), une des voies d'amélioration des techniques de traitement automatique du langage naturel (TALN) consiste à intégrer des méthodes statistiques aux systèmes actuels. Les techniques statistiques ont effectivement donné de bons résultats dans d'autres domaines du TALN, en particulier pour l'ASR avec des modèles de Markov cachés (HMM) appris sur une grande quantité de données. Les canaux de transmission entre les différents modules tendent donc à devenir plus riches en information par l'emploi d'objets probabilistes. Ainsi, les transcriptions textuelles entre l'ASR et le SLU sont remplacées avec profit par des listes *n-best* de transcriptions voire par des treillis de mots. On cherche aussi à utiliser des représentations symboliques fournies au gestionnaire de dialogue par le SLU sous la forme de listes *n-best* comportant des scores de confiance de plusieurs hypothèses concurrentes [Williams 2008a].

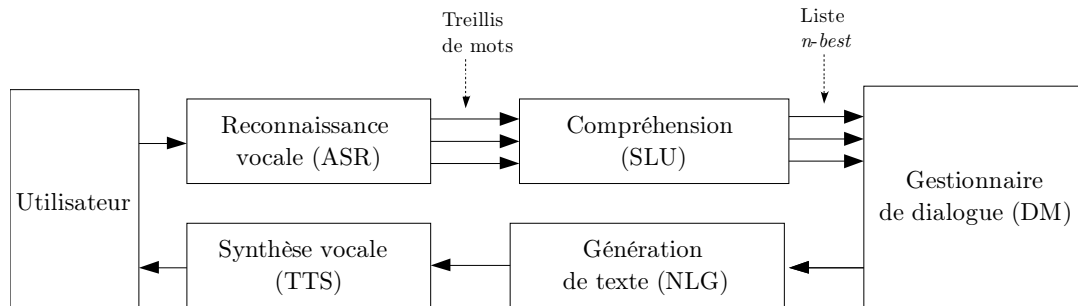


Figure 1.2: Élargissement des canaux de communications entre certains modules (*fat pipeline*).

1.1.2 Compréhension de la parole

Les modèles sémantiques du module de compréhension sont utilisés pour modéliser le sens d'un tour de parole dans le cadre d'un dialogue. On peut les classer de la façon suivante :

Premièrement, les modèles complexes issus du domaine de la linguistique, voir par exemple [McKevitt et al. 1992].

Deuxièmement, les approches utilisant la logique du premier ordre pour implémenter des principes linguistiques. Flexibles et puissants, ils disposent de bases plus mathématiques, ce qui les rend plus faciles à implémenter directement. Mais leur complexité pose le même problème : seuls des experts peuvent les adapter et les faire évoluer.

Troisièmement, on trouve les modèles de formulaires. Ils peuvent être vus comme une simplification à l'extrême de l'une ou l'autre des deux approches précédentes. Leur principale limitation vient de leur rigidité qui ne permet pas de modéliser des situations complexes non prévues par le concepteur. Du fait de leur simplicité, ils sont particulièrement appropriés à une utilisation dans un cadre statistique, c'est à dire avec une modélisation reposant sur des paramètres ou des probabilités qui sont estimés à partir de données. Cela explique que les modèles de formulaire soient ceux qui sont effectivement utilisés dans l'industrie.

Enfin, une approche unifiée est proposée par les projets tels que FrameNet [Baker et al. 1998] qui offre un dictionnaire de *frames* permettant de modéliser le sens d'un énoncé. Il est à mettre en relation avec Penn TreeBank [Marcus et al. 1993] ou WordNet [Miller et al. 1990] qui lui sont similaires. Une extension du paradigme FrameNet pour un domaine restreint a été proposée par les projets tels que MEDIA [Bonneau-Maynard et al. 2005] et LUNA [Duvert et al. 2008] à travers les modèles sémantiques riches par graphes de *frames*.

1.1.2.1 Généralités

i Spécificité du langage oral

En analysant un corpus ou en comparant les transcriptions de dialogues entre êtres humains, on constate que le langage oral est très différent de l'écrit. Il est impossible de transposer directement les modèles utilisés pour l'écrit pour modéliser un dialogue oral. En particulier parce que le langage oral est caractérisé par des phénomènes de disfluences tels que les hésitations, les reprises, et possède une agrammaticalité qui rend difficile la conception et l'application de règles logiques d'analyse de contenu.

ii Modèles d'actes de dialogue

La théorie des actes de langage est née des travaux d'Austin [Austin 1962], puis Searle [Searle 1969] dans le but de décrire plus complètement les actes de langage en distinguant les dimensions locutoire, illocutoire et perlocutoire. Les théories de Grice [Grice 1975] ont étendu ces idées et proposé d'utiliser la notion d'intention du locuteur. Ces théories ont posé les fondements de nombreuses modélisations du dialogue en TALN [Traum 2000] et sont à l'origine, entre autres, des propositions de [Traum et Larsson 2003] évoquées plus loin.

Afin d'appliquer concrètement ces théories linguistiques, différents modèles pratiques ont été mis en œuvre pour définir des actes de dialogue : un modèle d'annotation multicouche (DAMSL¹) [Core et al. 1997], MRDA² [Shriberg et al. 2004] ou encore un modèle spécifiquement conçu pour l'évaluation (DATE³) [Walker et Passonneau 2001].

(1) DAMSL : *Dialog Act Markup in Several Layers*.

(2) MRDA : *Meeting Recorder Dialog Act*.

(3) DATE : *Dialogue Act Tagging Scheme for Evaluation*.

Ces systèmes d’annotation en actes de dialogue sont rigides à plusieurs titres. En particulier, ils ne permettent pas de prendre en compte plusieurs actes de dialogue enchainés. Ainsi, par exemple, la phrase suivante contient deux actes de dialogue (promettre et déclarer). La seconde phrase, très commune dans le type de dialogues que nous étudions, est encore plus complexe et fait intervenir un verbe modal (vouloir) et un acte d’engagement (confirmer) :

« Je promets de vous déclarer mari et femme. »
 « Je voudrais savoir comment confirmer ma réservation. »

Tableau 1.1: Exemples de phrases difficiles à annoter avec les paradigmes DAMSL, MRDA ou DATE.

Dans la plupart des applications actuelles de dialogue homme-machine, il est nécessaire de représenter de telles phrases en les simplifiant, et cela offre des résultats satisfaisants. On peut toutefois se demander si cette simplification incontournable des représentations n’est pas la raison pour laquelle les systèmes de dialogue actuels sont incapables de s’appliquer à des tâches plus complexes.

iii Modèle d’ancrage (*grounding model*)

Si les actes de dialogue constituent la partie dynamique de celui-ci, le modèle qui reçoit et traite ces éléments atomiques de changement est appelé modèle d’ancrage. Il rend compte du statut de l’information présente dans le dialogue par rapport aux interlocuteurs. La notion de connaissance partagée, *commun ground*, est liée à un apport d’information par l’un des deux interlocuteurs ou bien à un accord ou un désaccord sur un concept ou une proposition.

Afin d’illustrer la notion d’ancrage, l’exemple suivant (tableau 1.2) montre un dialogue permettant de partager une connaissance commune. L’information sur le lieu où se trouvent les interlocuteurs est transmise dès le deuxième tour de parole : « LOCATION = Avignon » et associée à l’état d’ancrage `from_B`, montrant que cette information provient de l’interlocuteur B. Cependant, leurs deux représentations mentales ne s’accordent que grâce au troisième tour de parole. Cet état de partage d’information commune est modélisé par un état d’ancrage prenant la valeur « `grounded` ».

	Tour de parole	Information sémantique	État d’ancrage
1	A: Dans quelle ville sommes-nous, s’il vous plaît ?	LOCATION = ?	null
2	B: À Avignon.	LOCATION = Avignon	from_B
3	A: Avignon, d’accord, merci.	LOCATION = Avignon	grounded

Tableau 1.2: Exemple de dialogue entre deux interlocuteurs (A et B) illustrant l’évolution de l’état d’ancrage d’un élément d’information.

Pour modéliser le mécanisme d'ancrage, [Traum 1994; Roque et Traum 2008] ont proposé d'utiliser une machine à états finis, les transitions entre états d'ancrage sont alors réalisées selon l'acte de dialogue associé. Ce paradigme a été utilisé dans [Traum et Larsson 2003] ainsi que dans le système HIS [Young et al. 2010; Young et al. 2007].

1.1.2.2 Approches linguistiques

Les modèles issus de la linguistique permettent de détailler fortement le contenu d'un tour de parole. Cependant, leur complexité même les rend plus adaptés à l'utilisation par des experts linguistes (ou systèmes experts par règles) que par des machines statistiques. Ils ne seront pas décrits ici et on pourra se référer à [McKevitt et al. 1992] pour une description plus détaillée.

1.1.2.3 Logique du premier ordre

Il a été proposé d'utiliser des formules de logique du premier ordre pour représenter le sens des énoncés d'un dialogue. Dans le but de s'approcher des modèles complexes issus de la linguistique et de modéliser de façon plus précise les mécanismes internes aux interlocuteurs, on a défini les notions de croyance, de désir et d'intention sous forme de prédicats logiques. Cette représentation permet de réaliser des inférences logiques à partir d'un ensemble d'axiomes définis par le concepteur du système [Allen 1995]. Certains de ces axiomes peuvent être définis de façon générique, d'autres sont propres à la tâche.

1.1.2.4 Structure sémantique par *frames*

i Formulaires

Le modèle de formulaire est le modèle de représentation sémantique le plus utilisé pour des domaines restreints. Comme observé précédemment, il implique de fortes contraintes pour l'utilisateur et limite par conséquent les possibilités du système. Il ne permet pas, par exemple, de représenter plusieurs valeurs pour un même champ. De plus, la structure du formulaire doit être connue à l'avance par le concepteur du système. Enfin, toute modification requiert une intervention humaine experte.

ii Représentation flexible

Pour dépasser certaines limitations du modèle de formulaire, il a été proposé d'utiliser une représentation plus structurée de l'information. Ainsi par exemple, le système de dialogue HIS [Young et al. 2010; Young et al. 2007] repose sur une ontologie hiérarchique dont le modèle sémantique est élaboré selon une logique de compréhension et de gestion du dialogue conjointes. Un arbre, appelé ontologie, propre au domaine (figure 1.3) est défini *a priori* et sert de modèle pour la construction de la représentation sémantique.

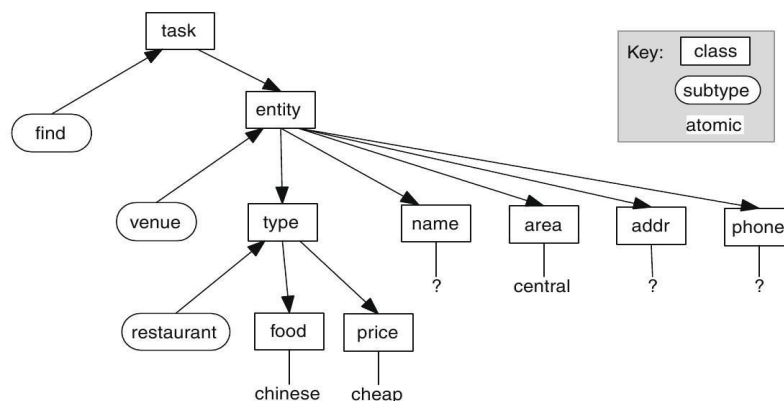


Figure 1.3: Arbre sémantique utilisé comme modèle pour définir l'ontologie du système HIS.
Extrait de [Young et al. 2010].

iii FrameNet

Une modélisation structurée peut être basée sur une représentation en *frames* sémantiques. Le projet FrameNet [Baker et al. 1998] propose d'utiliser la notion de *frame* pour avoir une représentation sémantique à la fois générique et adaptable à la tâche.

Cette notion de *frame* trouve son origine dans les travaux de [Fillmore 1985], Fillmore s'intéresse plus à ce qu'il appelle la sémantique U, ou sémantique de compréhension (Understanding) qu'à la sémantique T, ou sémantique de vérité (Truth). Pour lui, la valeur de vérité d'un énoncé n'est parfois pas l'information la plus importante. En revanche, il propose de s'intéresser plus précisément aux relations existant entre les concepts qui jouent un rôle essentiel dans l'interprétation du sens des énoncés.

Ainsi le projet FrameNet a permis de créer un dictionnaire de termes associés à un ensemble de *frames* sémantiques. Ces *frames* sémantiques permettent de représenter la sémantique des termes du dictionnaire, intégrant les relations avec les autres *frames*.

Le dictionnaire des termes fournit pour chaque terme :

- Sa définition en langage naturel.
- Une formule contenant des informations morphosyntaxiques indiquant quelle *frame* est utilisée pour représenter chaque sens du terme.
- Des exemples d'utilisation du terme défini.

La définition de l'ensemble des *frames* comporte pour chaque *frame* :

- La définition de la *frame* ainsi que des éléments la constituant.
- Les définitions des éléments de *frame* qui sont les concepts constituant la *frame*.
- Des exemples d'utilisation de cette *frame*.
- Des liens sur les *frames* correspondantes et sur les *frames* pertinentes des bases de données COMLEX [Grishman et al. 1994] ou WordNet [Miller et al. 1990; Gonçalves Oliveira et Gomes 2010] permettant de comparer ou d'utiliser conjointement ces différentes représentations.

Operating_a_system

Definition: An Operator manipulates the substructure of a System such that the System performs the function it was created for. This frame differs from the Using frame in that this frame does not profile the purpose of an agent but rather their manipulation of an entity (System/Instrument). In the case of using, the Agent need not manipulate the Instrument at all, as in: *Mrs. Adams used the room as a place to hang the family laundry to dry.*

This true story of domestic bliss was related to me by a friend of mine who for a time OPERATED a bar in San Francisco.

A California man has plead guilty to federal charges that he RAN a pyramid scam which defrauded nearly 7000 investors.

Core FEs¹:

Operator (Sentient) : The individual who manipulates the substructure of the System.

WELS Kingdom Workers OPERATES this program for students interested in doing ministry for a summer job.

System (Physical entity) : The entity that the Operator gets to perform its intended function.

Robert Faye, who has RUN the organization since its inception in 1991, said all the money the group has raised has been properly distributed.

Non-Core FEs:

Duration (Duration) : Duration denotes the length of the time interval in which the Operator runs the System.

The vendor has OPERATED this business for the last 7 years.

Manner (Manner) : Any description of the running of the System which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). In addition, it may indicate salient characteristics of an Operator that also affect the action (presumptuously, coldly, deliberately, eagerly, carefully).

The farm is owned and carefully OPERATED by John and Janice Maloney in Newberry Springs, California.

Means (State_of_affairs): This FE identifies the Means by which an Operator runs the System.

By the end of the film you'll be marveling at Buster's dexterity while he OPERATES the steamboat by climbing up and over or jumping down and around the ship.

Place (Locative relation) . This FE identifies the place where the Operator runs the System.

North Korea OPERATES a Restaurant in Vietnam.

Purpose (State_of_affairs) : This FE identifies the purpose for which an Operator runs the System.

The City OPERATES the system to ensure high quality water at competitive rates.

Time (Time) : This FE identifies the time when the Operator runs the System.

This guaranteed donation to the charity was begun by Eric Von Stroh, Mike Cioli and Andrew Seligson, the three men who RAN the business last year.

Inherits From: Using

Is Preceded by:

Is Inherited By: Operate_vehicle

Uses:

Subframe of:

Is Used By: Operational_testing

Has Subframes:

Perspective on:

Precedes:

Is perspectivized in:

Is Causative of:

See Also:

Lexical Units: manage.v, operate.v, run.v

Tableau 1.3: Définition FrameNet de la frame sémantique operating_a_system.

(1) FE : *Frame Element*, un élément de frame est un morceau d'information du type champ-valeur.

Ainsi, par exemple pour la *frame* `operating_a_system`, la base de données FrameNet fournit la définition du tableau 1.3 et le contexte de relation avec les autres *frames* est illustré par le graphe 1.4.

Notons que la proposition de FrameNet ne propose malheureusement pas de dictionnaire de *frames* dans d'autres langues que l'anglais.

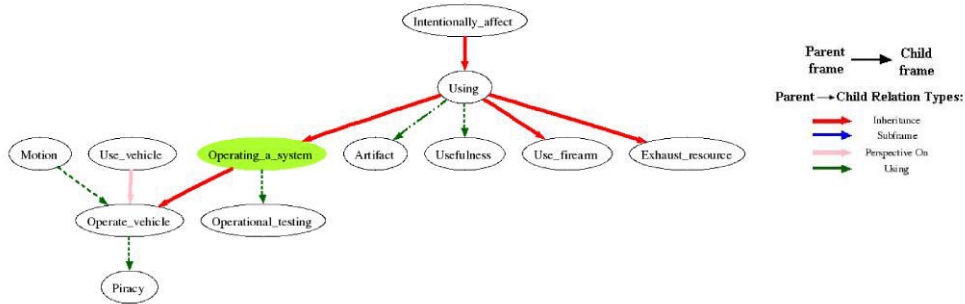


Figure 1.4: Relations de la *frame* `operating_a_system` avec les *frames* voisines dans FrameNet [Fillmore 1985].

iv Approche pratique pour le dialogue MEDIA-LUNA

Le paradigme FrameNet permet de représenter l'information sémantique de façon structurée et générique. Nous voyons ici comment les approches linguistiques de dictionnaires telles que celle de FrameNet peuvent être intégrées dans un système effectif.

Utilisant les données collectées lors du projet MEDIA [Bonneau-Maynard et al. 2005; Bonneau-Maynard et Lefèvre 2006], le projet LUNA [Meurs et al. 2009] s'appuie sur les principes de FrameNet pour leur ajouter une annotation sémantique de haut niveau. Il le généralise pour permettre des liens entre les *frames* et les spécialisent au domaine de la tâche considérée (informations touristiques). Le but est de disposer d'une représentation sémantique effective pour le dialogue tout en permettant un traitement statistique dans le cas du dialogue homme-machine. Les données du corpus MEDIA ont également été utilisées dans le projet Port-MEDIA [Lefèvre et al. 2010; Jabaian et al. 2010] afin d'étudier la portabilité des modèles sémantiques d'une langue à une autre.

MEDIA

Le projet MEDIA¹ [Bonneau-Maynard et al. 2005; Bonneau-Maynard et Lefèvre 2006] a permis la réalisation d'un corpus de 70 heures annoté sémantiquement de 1257 dialogues portant sur des informations touristiques (18831 tours de parole utilisateur) enregistrés par 250 utilisateurs différents selon la technique du magicien d'Oz. Cette dernière consiste à faire intervenir un véritable interlocuteur humain pour jouer le rôle de la machine, de sorte que l'utilisateur naïf se comporte comme s'il avait à faire à une machine. Afin de pouvoir entraîner et évaluer les systèmes d'annotation sémantique automatique, le corpus MEDIA a été annoté sémantiquement. Ces annotations offrent une première modélisation « à plat » du sens d'un énoncé par une séquence de triplets *mode-concept-valeur*. Les couples concept-valeur contiennent des informations factuelles telles que [nom-ville= Nice] ou [nombre-

(1) MEDIA : Méthodologie d'Evaluation de la compréhension du Dialogue hors et en contexte.

nuits = 2]. Pour intégrer des concepts plus généraux et essentiels à la compréhension d'un dialogue (tels que la négation ou l'affirmation d'une information), une étiquette *mode* est ajoutée à chaque concept, le *mode* d'un concept est positif (+), négatif (-), interrogatif (?) ou optionnel (~). Le dictionnaire sémantique comporte 83 concepts. Le taux d'erreur de reconnaissance sur le corpus MEDIA est de 33,5% (vocabulaire ASR : 2153 mots). Le taux d'erreur de compréhension en concept est de 21,3% sur les transcriptions exactes et de 43,3% sur les transcriptions automatiques.

Transcription	<i>Je voudrais réserver euh un hôtel plutôt euh vers Nice euh oui un hôtel à Nice pour deux nuits.</i>		
Concepts	Nom du concept	Mode	Valeur
	Réservation	+	
	Hôtel	+	
	Nom-ville	+	Nice
	Acceptation	+	
	Hôtel	+	
	Nom-ville	+	Nice
	Nombre-nuit	+	2

<i>Frames</i>	Nom de la frame	Champs	Valeurs
	F1 : VOULOIR	objet	F2
	F2 : RÉSERVATION	établissement période	F3 F4
	F3: HÔTEL	ville nom	Nice
	F4 : PÉRIODE	date-début date-fin durée	Pas de valeur Pas de valeur 2 jours

Tableau 1.4 : Exemple d'annotation en frames du corpus MEDIA-LUNA. La frame VOULOIR pointe sur la frame RÉSERVATION qui pointe sur les frames HÔTEL et PÉRIODE.

LUNA

Afin de pouvoir modéliser également l'information structurelle établissant les liens entre les concepts des énoncés, le projet LUNA a développé des techniques d'annotation sémantiques de haut niveau sur le corpus MEDIA [Meurs 2009]. La modélisation de l'état du dialogue par une structure de graphe de *frames* sémantiques vise à apporter la souplesse qui manque aux systèmes par formulaires. Le but est de favoriser des dialogues plus fortement dirigés par l'utilisateur et non plus uniquement par la machine. Il s'agit en fin de compte

d'obtenir une structure sémantique dirigée par les données, par le dialogue, par les interlocuteurs, par opposition à une structure sémantique définie par le concepteur du système.

Inspirée du paradigme de FrameNet (voir partie 1.1.2.4iii et [Baker et al. 1998]), l'annotation en *frames* ajoute aux concepts des rôles sémantiques constituant les éléments des *frames*. L'ontologie contient 21 *frames* et 86 éléments de *frames*. De plus, des liens entre les *frames* peuvent être établis : un élément de frame pointant sur une autre frame comme dans l'exemple du tableau 1.4. Un tour de parole utilisateur comporte en moyenne 5,2 *frames* ($\sigma = 3,3$; $\min=0$; $\max=26$) et 6,5 éléments de *frames* ($\sigma=9,0$; $\min = 0$; $\max = 88$). On dispose donc d'un corpus muni d'une annotation sémantique structurée par des graphes de *frames* sémantiques.

1.1.2.5 Flexibilité du modèle sémantique

Cette volonté de développer des modèles permettant de représenter l'information sémantique de façon structurée répond à un reproche qui apparaît de façon récurrente dans les commentaires des utilisateurs de systèmes de dialogue oral homme-machine : le manque de flexibilité du système. L'interaction est difficile car l'utilisateur doit s'adapter à la machine. Un des axes d'amélioration des systèmes sur ce point consiste à utiliser une représentation sémantique plus flexible.

Les modèles sémantiques généralement utilisés reposent sur le paradigme d'un formulaire pré-établi par le concepteur du système. Toute tentative de l'utilisateur de sortir de cette représentation est vouée à l'échec puisque la machine n'a pas la capacité de représenter d'autres notions que celles qui ont été prédéfinies.

Dans cette optique, les paradigmes de représentations riches en graphes de *frames* sémantiques [Meurs 2009] visent à permettre une plus grande flexibilité. Il va de soi que cette représentation dite « riche » reste encore éloignée de l'objectif d'une conversation libre et spontanée. Bien qu'elle ajoute la possibilité de modéliser des connections entre différents éléments d'informations, cette représentation n'est pas réellement tout à fait flexible et ne permet pas à l'utilisateur de s'éloigner complètement de celle que le concepteur du système a imaginé. Ainsi, il n'est pas possible à l'utilisateur d'introduire une *frame* qui n'a pas été définie dans l'ontologie du modèle, tout comme il ne peut faire référence à un concept inconnu ni utiliser un mot hors du vocabulaire du système. Cependant, nous pensons que cette limitation peut être progressivement dépassée en continuant les recherches dans cette direction en focalisant les efforts sur une difficulté majeure.

Il s'agit des difficultés de calculs induites par la complexité même de la représentation qui s'opposent à l'adoption de modèles sémantiques complexes tels que ceux reposant sur des *frames* sémantiques. En effet, pour un formulaire comportant un nombre restreint de champs (de l'ordre de 5, voire 10 dans les cas les plus complexes), la structure simple qu'il possède permet de simplifier les problèmes d'optimisation et d'apprentissage des modèles sur des données. Par exemple, il est possible de factoriser le modèle selon la liste des champs du formulaire ou d'effectuer une compression de l'ensemble des valeurs d'un champ par quantification vectorielle.. En revanche, lorsque la structure du modèle sémantique est celle d'un arbre sémantique, la taille de la représentation n'est pas bornée : un seul tour de dialogue de l'utilisateur peut comporter un nombre non limité d'informations. Les méthodes de simplification efficaces lors de l'utilisation de modèles de formulaire deviennent difficiles voire impossibles à appliquer.

De plus, lorsqu'on utilise un modèle sémantique riche par graphes de *frames* sémantiques, un autre inconvénient apparaît : les mécanismes de suivi de statuts d'ancrage ne s'appliquent plus directement. En effet, l'acte de dialogue du tour de parole n'est pas nécessairement unique : il peut y avoir plusieurs *frames* sémantiques d'actes de dialogue dans le graphe. Le mécanisme d'ancrage classique utilisé pour les formulaires doit donc être légèrement adapté pour s'appliquer à ce modèle sémantique.

1.1.3 Gestion du dialogue

Afin de situer l'intérêt de nos travaux concernant le gestionnaire de dialogue, nous décrivons dans cette partie différents paradigmes utilisés dans les systèmes de dialogue que nous avons choisi de présenter de manière historique afin de mieux rendre compte de l'évolution dans le domaine de recherche. Compte tenu de leur importance pour nos travaux, les systèmes de dialogue reposant sur des principes d'apprentissage par renforcement seront décrits dans une partie indépendante (partie 1.3).

Les modèles linguistiques déterministes continuent d'être raffinés pour représenter les phénomènes propres au dialogue tels que les confirmations, réparations et corrections d'erreurs. Cependant, ils ont une faiblesse intrinsèque face à la notion d'incertitude sur l'état du dialogue : ils ne la modélisent pas quantitativement selon une théorie mathématique décrite formellement. Cela demeure une de leur principale limitation.

Les systèmes utilisant des grammaires hors-contexte [Chomsky 1956] correspondent à une machine à états finis, ce sont les plus rigides mais aussi les plus simples à réaliser. Le développement de systèmes par plans puis à agent visait à modéliser les notions de croyance, de désir et d'intention (BDI) [Allen 1995] et même de croyance commune [Traum 1994]. On pourra consulter un état de l'art plus complet sur ce point dans [Churcher et al. 1997]. Un modèle repris par de nombreux travaux de la littérature du fait de sa généralité est celui de suivi de l'état d'information (ISU) [Traum et Larsson 2003]. Il propose de centrer la modélisation sur l'évolution de l'information disponible au système au cours du dialogue.

1.1.3.1 Approche structuraliste et grammaires

Initialement, dans les années 1950, on a commencé à chercher la structure de dialogue en identifiant les paires adjacentes (telles que par exemple une question ouverte suivie d'un apport d'information, ou bien une question fermée suivie d'une réponse positive ou négative). Puis, en cherchant à généraliser ce concept, on a commencé à utiliser des grammaires hors-contexte telles que celles de [Chomsky 1956] pour décrire les régularités des échanges lors du dialogue. Le modèle dit « genevois » [Roulet et al. 1991] est issu de cette démarche. Il peut être formulé sous la forme d'une grammaire de dialogue. Il possède deux niveaux d'analyses, hiérarchique et fonctionnelle, et a inspiré des extensions telles que le modèle dynamique de [Luzzati 1989] ou le modèle hypotético-expérimental (MHEX) de [Lehuen 1997; Rosset et Lamel. 2001]. Initialement prévu pour une analyse *a posteriori* de dialogues terminés accomplie par un linguiste expert du domaine [Bazillon et al. 2008], ce type de modèle nécessite d'être fortement simplifié pour être mis en œuvre dans un système automatique incapable d'intégrer la profondeur du raisonnement conceptuelle nécessaire.

1.1.3.2 Approches par plans

Afin de s'adapter à la nature informatique des systèmes de dialogue automatique tout en plaçant l'utilisateur au centre de la modélisation, on s'est tourné vers les méthodes issues du domaine de l'intelligence artificielle orientées plans [Perrault et Allen 1980]. Elles modélisent l'interlocuteur de la façon suivante : elles présupposent que chaque interlocuteur a un comportement rationnel et établit des plans pour atteindre ses buts. Concevoir un système de dialogue revient alors à préétablir l'ensemble des plans possibles de la tâche. Ensuite, le système utilisera une méthode d'inférence appropriée pour reconnaître le plan de l'interlocuteur parmi ces plans.

Le modèle de Croyance-Désir-Intention (BDI¹) de [Litman et Allen 1984; Allen 1995] ajoute un niveau cognitif à la modélisation du dialogue en définissant les notions de croyance, de désir et d'intention à l'aide de la logique du premier ordre. Ce paradigme modélise les interlocuteurs d'un dialogue comme des agents rationnels. Ceci permet de concevoir un système capable de créer des connaissances nouvelles à partir des informations fournies par l'utilisateur et par sa base de connaissances, en appliquant des règles logiques mimant le raisonnement humain.

Cependant, ces approches ne justifient que difficilement la présence de clarifications ou d'actes de dialogue de confirmation. Dans un système de dialogue fondé sur ce paradigme, ces mécanismes sont souvent générés ou traités de façon *ad hoc* et non générique.

Application : TRAINS – TRIPS

Un exemple de système par plans est fourni par le système TRAINS [Allen et al. 1996; Allen et al. 1994]. Dans ce système, le gestionnaire de dialogue (DM) est décomposé en plusieurs composants dont le gestionnaire de parole (*discourse manager*) et le module de raisonnement (*raisonner*). Le gestionnaire de parole maintient un modèle de l'utilisateur et du reste de l'environnement, il est chargé de résoudre les références et intervient lors de la génération de parole. Le module de raisonnement établit des plans et déduit des propositions logiques, il est implémenté dans langage un langage de programmation logique (Lisp).

Pour aller plus loin dans la modélisation, l'université de Rochester a donné un successeur à TRAINS : le système TRIPS² [Ferguson et Allen 1998] dont la conception est orientée par la tâche.

Application : ARTIMIS

L'utilisation de méthodes statistiques qui ajoutent une plus grande robustesse au système a été exploré par [Laroche 2010] à travers le système ARTIMIS³. Ce système reposant sur des formules de logique du premier ordre est mis en œuvre à l'aide d'un moteur d'inférence syntaxique [Sadek et al. 1997]. La gestion de l'incertitude lors du raisonnement logique effectué par le système repose sur une modélisation en logique modale.

(1) BDI : *Belief, Desire, Intention*.

(2) TRIPS : *The Rochester Interactive Planning System*.

(3) ARTIMIS : Agent Rationnel à base d'une Théorie de l'Interaction.

1.1.3.3 Modèle d'agenda

La réalisation d'un système par plans nécessite l'intervention d'un expert. Afin d'automatiser davantage ce processus, le modèle d'agenda [Wei et Rudnicky 1999] vise à généraliser les scripts de planifications établis par le concepteur du système en lui ajoutant plus de flexibilité. La représentation interne au système est un agenda.

Un agenda est constitué par un arbre d'actions (du système) dont chaque branche depuis la racine est une succession d'actions (du système) conduisant à la résolution de la tâche. Le parcours dans cet arbre est effectué selon une heuristique mais la généralisation selon un paradigme statistique est envisageable. La structure de l'arbre varie en fonction de l'environnement (actions de l'utilisateur, résultats des requêtes à une base de données).

Le modèle d'agenda est toutefois, en pratique, souvent simplifié en réduisant l'arbre initialement proposé par [Wei et Rudnicky 1999] en une simple liste (agenda à une seule branche).

Application : OLYMPUS – RAVENCLAW

Le système de dialogue Olympus [Bohus et al. 2007] développé à l'université Carnegie Mellon utilise le gestionnaire de dialogue Ravenclaw [Bohus et Rudnicky 2003; Bohus et Rudnicky 2009]. Il est open-source et peut donc être utilisé pour tester des variations autour du paradigme du modèle d'agenda. Plusieurs systèmes ont ainsi été développés grâce à Olympus pour des tâches diverses (aide aux personnes âgées [Raux 2003], assistance de maintenance [Bohus et Rudnicky 2002]) ou pour étudier la multimodalité dans le dialogue [Stenchikova 2007]

1.1.3.4 État d'information (ISU)

Fort des acquis des systèmes par plans, mais soucieux d'intégrer des mécanismes tels que ceux de clarification et de confirmation dans la dynamique du dialogue, [Cooper et al. 1999] propose d'utiliser le principe de l'état d'information et sa mise à jour (ISU¹). Ce principe vise à définir les principaux composants d'un système de dialogue selon un découpage très général en définissant l'état d'information, la façon de le mettre à jour et la façon de prendre une décision sur cet état.

L'état d'information s représente de façon synthétique toute l'information pertinente dont dispose le système sur l'historique du dialogue. Pour l'implémentation d'un système de dialogue, l'état d'information est défini théoriquement (*informational component*) et par une implémentation pratique (*formal representation*).

La dynamique de mise à jour de l'état d'information requiert la définition d'un ensemble d'actions a_u de l'utilisateur et d'actions a de la machine (*set of dialogue moves*), de règles de mise à jour de l'état s à partir des actions des interlocuteurs et d'un algorithme pour appliquer ces règles :

$$s' = Update(s, a_u, a) \quad (1)$$

en notant s' l'état de dialogue mis à jour d'après s , a_u et a .

(1) ISU : *Information State Update*.

Les règles de mise à jour envisagées par [Traum et Larsson 2003] sont des règles logiques et l'algorithme pour les appliquer peut utiliser un langage de programmation logique (tel que le langage Lisp). Enfin, dans le cadre de ISU, la stratégie est concrétisée par une fonction associant un score à chaque action possible selon l'état d'information courant.

Application : TRINDIKIT – DIPPER – GODIS

Le modèle de mise à jour de l'état d'information est suffisamment flexible pour avoir été utilisé dans plusieurs systèmes de dialogue. C'est le cas par exemple de l'architecture de développement Trindikit [Larsson et al. 2002] dans le cadre du projet SIRIDUS¹. Le langage de mise à jour de l'état est un langage de programmation logique. Le système utilise des modules commerciaux de Nuance pour la reconnaissance vocale et le module de TTS. Trindikit a permis de développer le système Godis [Larsson 2007].

Dans la continuité de Trindikit, et toujours fondé sur la théorie de l'état d'information, le SRI a développé Dipper [Bos et al. 2003], un ensemble d'outil de conception de systèmes de dialogue. Il repose sur l'architecture orientée agent (OAA²) présenté dans [Cheyer et Martin 2001]. Malgré certains efforts réalisés pour prendre en compte des observations incertaines, ces systèmes intègrent difficilement une théorie probabiliste qui pourrait fournir un traitement quantitatif à cette question.

Les travaux de [Lemon et al. 2006; Henderson et al. 2008; Rieser 2008] reprennent le modèle ISU en exploitant la structure dont le mécanisme de mise à jour s'adapte bien à une implémentation informatique. Cette proposition utilise en outre une optimisation de stratégie par renforcement à l'aide d'un environnement de simulation complet.

Application : HIS

Dans le cadre de ISU, la définition de l'état d'information ainsi que l'étape de mise à jour sont déterministes. Il est cependant possible de généraliser ce paradigme en utilisant un état d'information probabiliste, c'est la voie poursuivie par le système HIS³ d'information d'état caché [Young et al. 2007]. Les modèles fondés sur la théorie de l'apprentissage par renforcement seront décrits en partie 1.3.

1.1.3.5 Approche de l'industrie : VoiceXML

Toutes ces approches, issues des systèmes par plan et de leurs améliorations successives, restent très complexes et coûteuses à mettre en œuvre en pratique, c'est pourquoi l'industrie préfère utiliser un outil plus rigide mais plus simple : le langage d'annotation pour la voix VoiceXML⁴ [VoiceXML 2004]. Il s'agit d'une spécification du W3C⁵ permettant de créer des pages web avec lesquelles l'utilisateur interagit par un dialogue oral à l'aide d'un navigateur approprié. VoiceXML est l'équivalent du langage HTML pour un mode d'interaction oral.

Formellement, une application VoiceXML se définit par un graphe d'états s et deux fonctions de contrôle F et G . Une première fonction F est utilisée pour décider des transi-

(1) SIRIDUS : *Specification Interaction and Reconfiguration in Dialogue Understanding Systems*.

(2) OAA : *Open Agent Architecture*.

(3) HIS : *Hidden Information State*.

(4) VoiceXML : *Voice eXtensible Markup Language*.

(5) W3C : *World Wide Web Consortium*.

tions d'états en fonction de l'observation o reçue. La deuxième fonction G permet de prendre la décision de l'action système a en fonction de l'état s actuel.

$$s' = F(s, o) \quad (2)$$

$$a = G(s) \quad (3)$$

Un autre langage similaire, Salt¹ [Wang 2002], a été proposé mais il a été abandonné car il n'a pas réussi à s'imposer comme standard pour l'industrie.

Dans le cadre de VoiceXML comme de Salt, il est nécessaire de définir les fonctions de contrôles de façon experte. Des bibliothèques génériques ont été développées en langage VoiceXML afin de gérer la partie du dialogue qui ne dépend pas du domaine, mais toute modification ou mise à jour de la stratégie de dialogue requiert une intervention humaine. Un défi important pour l'industrie est de permettre la conception automatique et l'adaptation de la stratégie du système, qui ne rentrent pas dans le cadre du langage VoiceXML.

La spécification VoiceXML donne lieu à des implémentations variées selon les constructeurs (Nuance, Loquendo, Unisys, BeVocal, HeyAnita, Tellme, etc.). Cependant, elles restent toutes fondées sur un paradigme déterministe où le dialogue se trouve dans un état donné complètement connu par le système. Elles permettent donc difficilement d'intégrer de l'information probabiliste de façon quantitative.

1.1.3.6 Autres approches

Différentes approches s'écartant des précédentes ont été proposées, nous nous contenterons de les citer ici pour référence car cette recherche ne les utilise pas. L'approche collaborative utilise principalement les maximses de coopération et de quantité [Grice 1975] dans le but de réaliser un but conjoint; d'autres approches sont fondées sur la négociation [Dillenbourg et Baker 1996], et d'autres reposent sur un modèle de *frames* (sans rapport avec les *frames* sémantiques décrites dans cette thèse) [Kim et al. 2008] ou sont orientés par la tâche [Raux et Eskenazi 2004], le but de l'utilisateur, ou encore par la situation [Lee et al. 2006].

1.1.3.7 Limitations - Discussion

La nature purement déterministe des systèmes de dialogue classiques rend certains points difficiles à traiter. D'une part, concernant la gestion du bruit, la difficulté à gérer les erreurs de façon quantitative entraîne un problème de robustesse face aux erreurs de reconnaissance vocale et de compréhension. D'autre part, lorsque les conditions changent (changement de domaine, évolution de la tâche, évolution des demandes des utilisateurs, etc.), l'évolution et la maintenance d'un système sont assurées par un expert humain, d'où des coûts de développement élevés.

Afin d'introduire des concepts statistiques dans les systèmes de dialogue, on a cherché à généraliser les modèles précédents en remplaçant les variables déterministes par des variables probabilistes. Ces méthodes reposent sur des théories mathématiques permettant de modéliser l'incertitude due aux erreurs de reconnaissance vocale et de compréhension. De

(1) Salt : *Speech Applications Language Tabs*.

plus, elles tendent à considérer des paradigmes d'apprentissage automatique permettant une conception plus rapide. L'adaptabilité continue du système aux changements de l'environnement peut alors être effectuée sans modification majeure du système voire, idéalement, sans intervention humaine.

Un des paradigmes statistiques considéré actuellement repose sur des mécanismes d'apprentissage par renforcement, cependant ces méthodes reposent sur une modélisation de l'utilisateur et de l'environnement rigide et simplifiée. En effet, le modèle de formulaire pré-établi se factorise en une liste de couples champ-valeur connus à l'avance et les modèles mathématiques sous-jacents sont ainsi plus faciles à traiter.

1.1.3.8 Récapitulatif des implémentations existantes

Nom du système	Théorie mise en œuvre	Tâche	Probabiliste vs Déterministe	Publications associées
Trips	Orienté tâche	Recherche d'itinéraire	Déterministe	[Ferguson et Allen 1998]
Trains	Orienté plan	Recherche d'itinéraire	Déterministe	[Allen et al. 1994] [Allen et al. 1996]
Artemis	Logique modale	Assistance technique	Probabiliste	[Sadek et al. 1997]
Olympus	Agenda	Tâches multiples	Déterministe	[Bohus et al. 2007]
Trindikit	Information state	Standard téléphonique, automobile	Déterministe	[Larsson et al. 2002]
Dipper	Information state	Tâches multiples	Déterministe	[Bos et al. 2003]
GoDis	Information state	Tâches multiples	Déterministe	[Larsson 2007]
HIS	POMDP + Information state	Informations touristiques	Probabiliste	[Young et al. 2007]
BUDS	POMDP	Informations touristiques	Probabiliste	[Thomson et Young 2010]
	Logique probabiliste (LFPR ¹)	Assistance technique	Probabiliste	[Laroche 2010]

Tableau 1.5: Récapitulatif non-exhaustif des prototypes de systèmes de dialogue publiés.

Les implémentations ou les *frameworks* pour réaliser des systèmes de dialogue sont nombreux : outre la spécification VoiceXML utilisée dans l'industrie, il a été développé des systèmes orientés tâche (TRIPS) ou orientés plan (TRAINS), ainsi que plusieurs systèmes reposant sur le principe d'état d'information (ISU) proposé par Traum (Trindikit, Dipper, Godis). Les prototypes complets de systèmes de dialogue par apprentissage par renforcement sont en revanche peu nombreux et se résument essentiellement à ceux de l'équipe de

(1) LFPR : *Logical Framework for Probabilistic Reasoning*.

Cambridge (HIS et BUDS). Le tableau 1.5 propose un récapitulatif non-exhaustif des implémentations décrites dans la littérature.

Les systèmes actuellement disponibles peuvent être répartis en trois groupes :

- Ceux qui sont par nature déterministes et ne sont pas conçus pour modéliser de façon statistique l'incertitude sur l'état actuel du dialogue due à l'incertitude de reconnaissance du tour utilisateur.
- Ceux qui modélisent l'incertitude sur l'état de dialogue de façon *ad-hoc*, ou de manière très spécifique.
- Ceux qui permettent de tester des hypothèses d'apprentissage automatique par renforcement. Ils sont peu nombreux et ne sont pas librement utilisables et ne peuvent donc être mis en œuvre directement. De plus l'intégration d'un modèle sémantique plus riche implique souvent une modification profonde du système de compréhension. Ceci qui explique qu'il a été nécessaire de développer notre propre système.

1.1.4 Interface entre gestion du dialogue et compréhension

L'information sémantique transitant du SLU au DM est généralement représentée sous la forme d'un formulaire dont la structure est pré-établie. La proposition du système HIS [Young et al. 2010] consiste à utiliser une représentation par un arbre sémantique à deux niveaux qui se construit au fur et à mesure du dialogue (voir figure 1.3). Dans le cadre de l'apprentissage par renforcement pour le dialogue, aucune proposition n'a été faite pour utiliser une représentation plus flexible que celle-ci.

Pour généraliser les paradigmes statistiques utilisés avec profit dans les modules situés en amont de la boule de traitement du système (ASR [Linarès et al. 2007] et SLU [Meurs 2009]), on cherche à intégrer le maximum de compétences dans le module de SLU. C'est l'approche des projets MEDIA-LUNA dans lesquels s'intègrent ces travaux : chercher le maximum de flexibilité et d'information sémantique dans la représentation fournie par le module de compréhension.

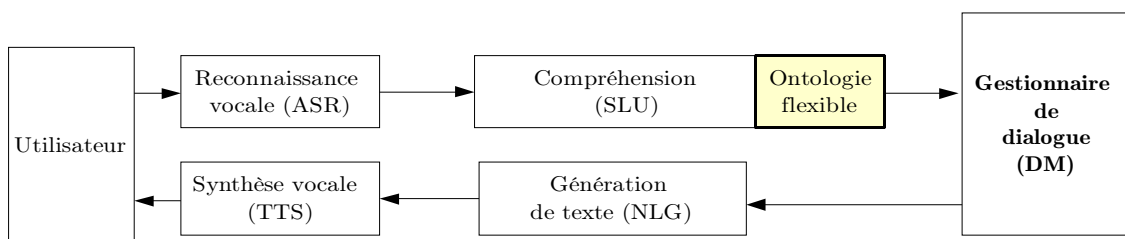


Figure 1.5: Système de dialogue avec modèle sémantique flexible.

Le but visé par la représentation en graphes de *frames* au lieu de simple formulaire est le suivant :

- obtenir une séparation claire de la partie compréhension et la partie gestionnaire de dialogue.
- effectuer le processus de compréhension sémantique dans la mesure du possible dans le module de compréhension. Une succession d'annotations s'éloigne progressive-

ment de la structure de surface de la phrase pour s'approcher d'une structure symbolique modélisant le sens des énoncés. Ces annotations successives offrent une représentation en concepts hors contexte, puis en concepts en contexte (établissant des liens avec le reste du tour de parole) et enfin en *frames* de rôles sémantiques.

Bien sûr, il est nécessaire de disposer d'un corpus avec des annotations en concepts et en *frames* de rôles sémantiques. L'utilisation de tels modèles par un gestionnaire de dialogue n'a pas encore été explorée.

Propagation de l'incertitude

L'utilisation de méthodes statistiques et du paradigme de communication du *fat pipeline* (décrit par la figure 1.2, p 20) permet de propager la notion d'incertitude d'un module à l'autre et de retarder au maximum le choix entre les différentes interprétations possibles de l'acte de dialogue de l'utilisateur. Le processus de décision final est effectué en tenant compte de l'incertitude sur toute la chaîne de traitement du signal d'entrée.

Le suivi de croyance est de la responsabilité du module de gestion de dialogue qui doit maintenir un état courant du dialogue. Le module de SLU doit pour sa part fournir au gestionnaire de dialogue une représentation sémantique flexible sans être contraint par une structure fixée au préalable. À défaut de connaître l'état exact du dialogue, le DM maintient sa meilleure estimation de l'état de dialogue d'après l'information disponible (c'est-à-dire, sachant tout l'historique du dialogue). Dans une approche probabiliste bayésienne, cette estimation est, par définition, la croyance sur l'état de dialogue.

1.2 Apprentissage par renforcement

Résumé : Les méthodes d'apprentissage par renforcement par POMDP permettent de planifier une stratégie tenant compte de l'état d'incertitude sur l'environnement. Pour des modèles simples, des stratégies optimales sont obtenues par des algorithmes de résolution exacts. Sinon, des méthodes d'approximation permettent de réduire la complexité du modèle et de trouver des stratégies sous-optimales.

La motivation pour utiliser des méthodes d'apprentissage automatique pour le dialogue est la même que celle qui a donné de bons résultats dans la plupart des domaines du Traitement Automatique du Langage Naturel : utiliser le paradigme statistique orienté par les données (*data-driven*) qui a été efficace pour d'autres domaines tel que celui de la reconnaissance automatique (ASR) ou la compréhension (SLU).

Les modèles issus de l'apprentissage par renforcement s'avèrent particulièrement intéressants pour modéliser le dialogue du fait de sa nature interactive. En effet, l'objet mathématique que constitue la chaîne de Markov permet de modéliser la progression dynamique du dialogue et le conditionnement par une action du système intègre cette notion d'évolution interactive. Ainsi, les processus de décision de Markov (MDP¹) offrent un cadre théorique pour optimiser la stratégie d'un système par une planification du cours futur du dialogue.

Afin de prendre en compte de multiples hypothèses lors du processus de décision effectué par le gestionnaire de dialogue et d'assurer ainsi une planification prenant en compte l'incertitude, les processus de décision de Markov partiellement observables (POMDP²) ajoutent un niveau de complexité en utilisant des variables cachées.

L'utilisation de modèles d'apprentissage par renforcement constituant un point important de nos propositions, nous présentons dans cette partie le domaine de l'apprentissage par renforcement à travers ses principaux modèles : MDP (en 1.2.1) et POMDP (en 1.2.2). Les notions de stratégie optimale (décrite en 1.2.3) et de fonctions valeurs V et Q (en 1.2.4) sont également essentielles pour comprendre les différences entre les familles d'algorithmes de résolution, avec et sans modèle (en 1.2.5).

Afin de rendre cette optimisation de stratégie possible, certaines méthodes permettent de réduire la complexité des calculs et de définir des stratégies approximées. On présente donc (en 1.2.6) deux méthodes consistant à effectuer des résumés des espaces considérés. La première est le résumé d'état (pour un MDP) ou résumé de croyance (pour un POMDP), la deuxième est un résumé d'action.

(1) MDP : *Markov Decision Process*.

(2) POMDP : *Partially Observable Markov Decision Process*.

1.2.1 Processus de Décision de Markov (MDP)

Une très bonne introduction aux Processus de Décision de Markov (MDP) peut être trouvée dans [Sutton et Barto 1998; Sigaud et Buffet 2008].

Les MDPs font appel à des concepts issus de la théorie des probabilités et de la théorie de la décision [Peterson 2009]. Ils mettent en place un cadre théorique permettant de définir une stratégie optimale pour un système en interaction avec son environnement dont l'évolution dans le temps n'est pas déterministe. Ils permettent en particulier une planification statistique : c'est-à-dire la prise en compte, lors de la planification, de l'incertitude sur l'effet d'une prise de décision.

Le modèle MDP se décrit formellement de la façon suivante :

- Modélisons la suite des états de l'environnement par une chaîne de Markov : s_t . On notera \mathbb{S} l'espace des états.
- Le système étant en interaction avec l'environnement, les transitions d'états seront naturellement conditionnées par les actions a_t du système. On notera \mathbb{A} l'espace des actions.
- Enfin, pour définir une stratégie optimale, on pose un critère définissant la qualité d'une action selon la transition qu'elle a générée. Ce critère est appelé fonction de récompense immédiate :

$$r_t = r(s_t, a_t, s_{t+1}) \quad (4)$$

Notons que, dans la littérature, la récompense r_t est fréquemment simplifiée en ne la faisant pas dépendre de l'état d'arrivée s_{t+1} , elle s'écrit alors simplement : $r_t = r(s_t, a_t)$.

L'hypothèse de Markov conditionnelle est la suivante :

$$P(s_{t+1}|s_t, a_{t-1}, s_{t-1}, \dots) = P(s_{t+1}|s_t, a_t) \quad (5)$$

Les hypothèses d'indépendance peuvent être représentées par le réseau bayésien dynamique [Jensen et Nielsen 2007] suivant :

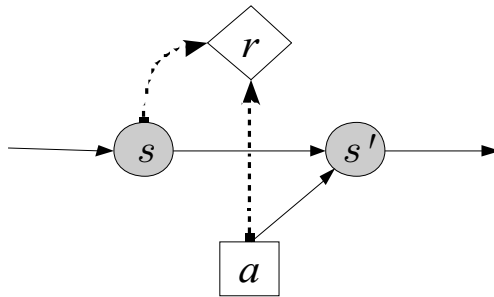


Figure 1.6: Graphe de décision d'un MDP, $s = s_t$ est l'état à la date t et $s' = s_{t+1}$ est l'état à la date $t+1$

Le système prend la décision d'effectuer une action à la date t selon sa connaissance de l'environnement représentée par l'état s_t . Cette décision est dirigée par une stratégie π suivie par le système. π donne l'action a_t à prendre en fonction de l'état courant s_t :

$$a_t = \pi(s_t) \quad (6)$$

1.2.2 Processus de Décision de Markov Partiellement Observables (POMDP)

Un POMDP¹ est défini comme un MDP, mais en remplaçant la chaîne de Markov des états par une chaîne de Markov cachée. Le modèle POMDP se décrit formellement de la façon suivante :

- La suite des états de l'environnement est cette fois une chaîne de Markov cachée : (s_t, o_t) . On notera \mathbb{S} l'espace des états et \mathbb{O} celui des observations.
- Les transitions d'états sont toujours conditionnées par les actions a_t du système. On notera \mathbb{A} l'espace des actions.
- Enfin, on pose toujours une fonction de récompense immédiate, critère définissant la qualité d'une action selon la transition qu'elle a générée.

$$r_t = r(s_t, a_t, s_{t+1}) \quad (7)$$

Comme pour les MDP, la récompense r_t est fréquemment simplifiée en : $r_t = r(s_t, a_t)$.

Les hypothèses d'indépendance s'écrivent :

$$P(s_{t+1}|s_t, a_{t-1}, s_{t-1}, \dots) = P(s_{t+1}|s_t, a_t) \quad (8)$$

$$P(o_t|s_t, a_{t-1}, s_{t-1}, \dots) = P(o_t|s_t) \quad (9)$$

et sont représentées par le graphe de décision 1.7 :

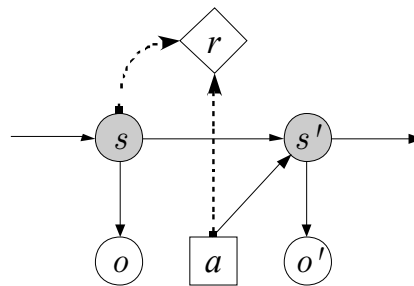


Figure 1.7: Graphe de décision d'un POMDP

Croyance

La croyance b est définie comme la probabilité conditionnelle de l'état s sachant tout l'historique observé du dialogue $H^{(o)} = (o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o_t)$.

(1) POMDP : *Partially Observable Markov Decision Process*.

$$b(s) = P(s | H^{(o)}) = P(s | o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o_t) \quad (10)$$

Elle nous intéresse particulièrement en tant que statistique suffisante pour s de l'historique observé $H^{(o)}$. En particulier, elle possède la propriété (11) selon laquelle toute variable s'exprimant comme une fonction de l'historique observé $H^{(o)}$ peut s'exprimer comme une fonction de b :

$$\begin{aligned} \text{Pour toute variable aléatoire } z \text{ s'écrivant } z = f(H^{(o)}), \text{ (avec } f \text{ mesurable)} \\ \text{il existe une fonction } g \text{ telle que } z = g(b) \end{aligned} \quad (11)$$

En particulier, toute stratégie π fonction de l'historique observé $H^{(o)}$ peut s'écrire comme une stratégie π' qui est fonction de la croyance : $\pi(H^{(o)}) = \pi'(b)$.

On notera \mathbb{B} l'espace des vecteurs de croyance.

Lorsqu'une nouvelle observation o' est effectuée, en appliquant les formules de probabilités conditionnelles on obtient la nouvelle croyance b' à partir de l'ancienne croyance b selon l'équation de mise à jour du vecteur de croyance :

$$b'(s') = k \cdot P(o'|s') \sum_{s \in \mathbb{S}} P(s'|a, s) b(s) \quad (12)$$

où k est une constante de renormalisation.

Représentation compacte d'un POMDP

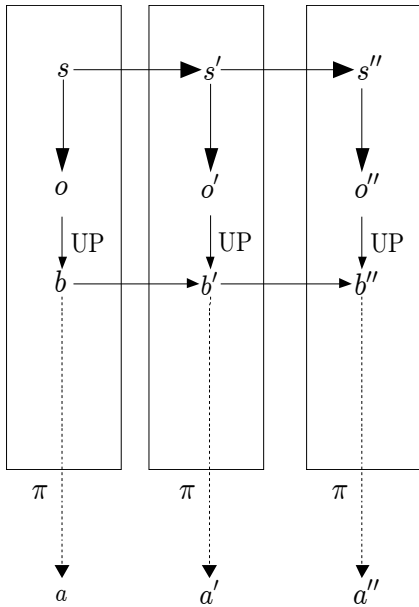


Figure 1.8: Représentation détaillée d'un POMDP

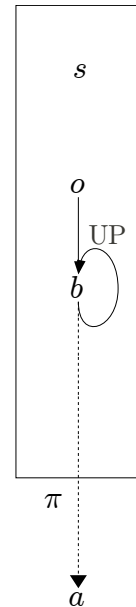


Figure 1.9: Représentation compacte d'un POMDP

Afin de pouvoir représenter plusieurs POMDPs sur un même schéma, on ne représentera qu'une tranche de temps du modèle comme sur la figure 1.9, au lieu d'une représentation détaillée comme sur la figure 1.8. Cette représentation met en valeur les deux étapes essentielles de la modélisation POMDP : la mise à jour de la croyance et la stratégie. Les dépendances en probabilité sont implicites.

Autres modèles

Il existe d'autres modèles qui généralisent les POMDP. Par exemple, les représentations de prédiction d'état (PSR¹) [Singh et al. 2004; Michael Littman et Singh 2002; Roy et al. 2000] ne maintiennent pas une distribution de croyance sur l'espace des états mais une fonction de prédiction des observations futures. Les POMDPs (à k états) eux-même sont généralisés par les PSRs à k tests. On peut citer également les modèles par opérateurs observables (OOM²) [Jaeger 2000] ou les DEC-POMDP [Corona et Charpillet 2010].

1.2.3 Stratégie optimale

On peut remarquer qu'un POMDP sur les espaces d'états, observations, actions ($\mathbb{S}, \mathbb{O}, \mathbb{A}$) est formellement identique à un MDP sur les espaces d'états et d'actions (\mathbb{B}, \mathbb{A}) dont les probabilités de transition entre croyances s'expriment selon la formule de mise à jour (12). Ce qui est décrit ici pour les MDP s'applique donc également directement aux POMDP. La résolution d'un MDP consiste à trouver une stratégie π optimale selon le critère défini ci-dessous (17).

Récompense cumulée

La récompense immédiate r_t (4) permet de définir la récompense cumulée R_t obtenue par le système depuis la date $t = 0$.

$$R_t = \sum_{k=1}^{\tau} \gamma^k r_k \quad (13)$$

où γ est le facteur d'escompte (*discount factor*) proche mais strictement inférieur à 1, qui permet d'être certain que la somme converge même dans le cas d'un processus à horizon infini.

Fonctions valeurs

La fonction V est la valeur d'un état s : l'espérance moyenne obtenue à partir de cet état en suivant une stratégie π . Elle est définie par :

$$V^\pi(s) = \mathbb{E} \left(\sum_t r_t \mid s_t = s, a = \pi(s) \right) \quad (14)$$

et peut être calculée récursivement :

(1) PSR : *Predictive State Representation*.

(2) OOM : *Observable Operators Models*.

$$V^\pi(s) = \mathbb{E}(r_t + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = \pi(s)) \quad (15)$$

On définit de façon similaire la fonction Q qui est la valeur d'un couple état-action. Elle est utilisée par les algorithmes de résolution de *Q-learning* :

$$Q^\pi(s, a) = \mathbb{E}(r_t + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1})) \mid s_t = s, a) \quad (16)$$

Optimalité

La résolution d'un MDP consiste à trouver une stratégie π_\star optimale selon V :

$$\pi_\star(s) = \operatorname{argmax}_\pi \mathbb{E}(V^\pi(s)) \quad (17)$$

Et pour un POMDP :

$$\pi_\star(b) = \operatorname{argmax}_\pi \mathbb{E}(V^\pi(b)) \quad (18)$$

π_\star est la stratégie optimale du POMDP, celle qui maximise l'espérance de la fonction valeur V .

1.2.4 Représentation de la stratégie

La stratégie π peut être représentée de multiples façons. Le choix d'implémentation de cette représentation dépend de l'algorithme utilisé ainsi que des possibilités de représentation compactes (c'est-à-dire une représentation par un petit nombre de paramètres) qui en découlent.

L'énumération complète (tabulée) d'une stratégie n'est pas toujours envisageable. Dans le domaine du dialogue oral homme-machine, que nous détaillerons par la suite, [Lefevre et al. 2009] propose de l'échantillonner et d'utiliser un algorithme de k plus proches voisins pour l'appliquer.

Les algorithmes par itération de la fonction valeur fournissent généralement une suite d'approximations à horizon fini. Ces approximations sont linéaires par morceaux et par conséquent paramétrées par un ensemble d'hyperplans, ou de façon équivalente par leurs vecteurs orthogonaux associés : les vecteurs α .

Il a également été proposé l'utilisation de contrôleurs à états finis (FSM¹). Ces FSM peuvent être déterministes [Meuleau et al. 1999] ou bien stochastiques, dans ce dernier cas les actions sont sélectionnées selon une distribution de probabilité associée à chaque nœud [Amato 2007]. Ce genre de modélisation donne cependant lieu à une explosion combinatoire sur la taille du contrôleur et des algorithmes comme l'itération sur stratégie bornée (BPI²) [Poupart et Boutilier 2003] visent à traiter ce problème. On trouvera un état de l'art plus précis sur ce domaine dans [Hansen 2008].

(1) FSM : *Finite State Machine*.

(2) BPI : *Bounded Policy Iteration*.

1.2.5 Algorithmes de résolution

L'apprentissage d'une stratégie optimale est effectué par des algorithmes d'optimisation statistiques qui nécessitent généralement une grande quantité de données. Les algorithmes de résolution se répartissent en deux familles : avec ou sans modèle.

Algorithmes avec modèle

Dans les algorithmes avec modèle, le modèle des probabilités de transitions d'état et d'observations est connu. Il peut par exemple être appris sur un corpus (réel ou simulé). La faible quantité des données pouvant être compensée par des méthodes de replis [Goddeau et Pineau 2000; Katz 1987].

Les POMDP ont longtemps été réputés impossibles à calculer mis à part pour des cas triviaux du fait de leur complexité. Cependant, des algorithmes récents tels que PBVI¹ [Pineau et al. 2006] ou Perseus [Spaan et Vlassis 2005] conduisent à revoir cet *a priori* et rendent envisageable l'utilisation de résolution avec modèle pour des cas concrets.

Il faut noter cependant un des obstacles à l'utilisation d'un modèle complet appris sur des données : dans le corpus utilisé pour apprendre le modèle de probabilité, la stratégie suivie par le système peut être très différente de la stratégie optimale. Le modèle est alors peu pertinent pour modéliser des dialogues effectués avec la stratégie optimale.

Algorithmes sans modèle

Dans les algorithmes sans modèle, de type Monte Carlo, une interaction continue avec l'environnement permet au système de constater l'effet de son choix de stratégie et de s'y adapter.

De nombreux algorithmes ont été développés pour les MDP [Sutton et Barto 1998; Sigaud et Buffet 2008]. Pour les POMDP, il a été proposé par exemple une méthode de résolution par Monte Carlo en représentant le vecteur de croyance par un échantillonnage dynamique [Thrun 2000].

1.2.6 Résolution approchée par espaces résumés

De nombreuses méthodes visent à résoudre un POMDP en l'approximant. Elles utilisent essentiellement la structure des espaces \mathbb{S} , \mathbb{A} , \mathbb{B} ou du modèle de probabilité [Poupart 2005]. Par exemple, si l'espace d'état est factorisé en champs prédéfinis (modèle de formulaire), il est possible d'utiliser un POMDP factorisé [Sim et al. 2008]. La méthode que nous détaillons ici ne requiert pas la factorisation de l'état, elle utilise un partitionnement des espaces des croyances et des actions en des croyances ou actions *résumées*.

i Résumé de croyance

Note préliminaire : le résumé de croyance décrit ci-dessous peut être appliqué sur un MDP pour réaliser un résumé de l'espace d'état. On le décrit ici sur un POMDP, avec le vecteur de croyance b , afin de garder une notation cohérente dans le reste de la thèse.

(1) PBVI : *Point Based Value Iteration*.

Pour faire un résumé de vecteur de croyance, il s'agit de définir un vecteur de croyance résumé \tilde{b} qui doit contenir toute l'information pertinente présente dans le vecteur de croyance initial. On utilise pour cela une fonction résumée (*mapping* M) permettant de projeter le vecteur de croyance de l'espace maître \mathbb{B} vers l'espace de croyance résumé $\tilde{\mathbb{B}}$:

$$\begin{aligned} M : \mathbb{B} &\rightarrow \tilde{\mathbb{B}} \\ b &\mapsto \tilde{b} = M(b) \end{aligned} \tag{19}$$

La factorisation de la stratégie π par la fonction résumée de croyance M peut être représentée graphiquement par :

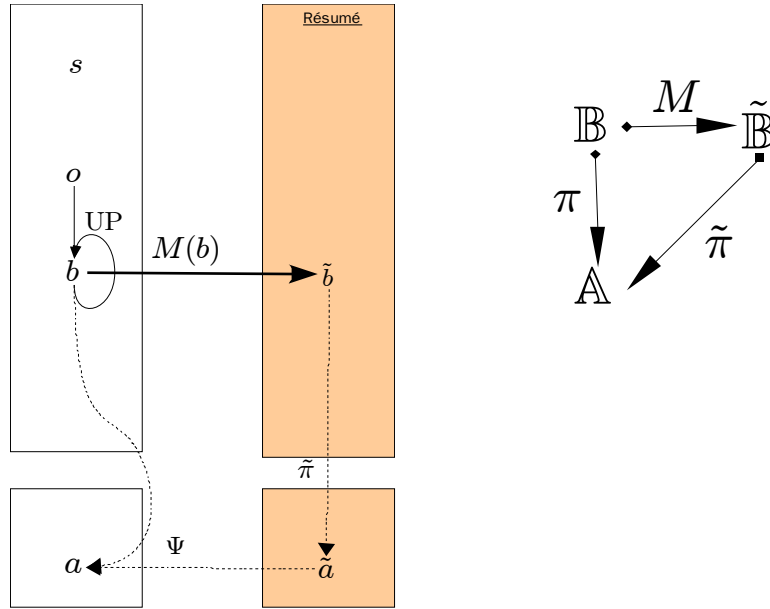


Figure 1.10: Résumé de croyance : $\pi(b) = \tilde{\pi}(M(b))$

Après avoir résumé le vecteur de croyance en une croyance résumée, l'optimisation peut être réalisée en cherchant une approximation de la stratégie optimale sur un espace de recherche réduit. Ainsi, au lieu de l'optimisation sur l'ensemble Π des stratégies possibles dépendant de b (20), on se contente d'une approximation en optimisant sur $\Pi_{\tilde{\mathbb{B}}}$: les stratégies dépendant de \tilde{b} (21).

$$\pi_{\star}(b) = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}(V^{\pi}(b)) \tag{20}$$

$$\pi_{approx}(b) = \operatorname{argmax}_{\pi \in \Pi_{\tilde{\mathbb{B}}}} \mathbb{E}(V^{\pi}(b)) \tag{21}$$

avec

$$\Pi = \{\pi : \mathbb{B} \rightarrow \mathbb{A}\} \quad (22)$$

$$\Pi_{\tilde{\mathbb{B}}} = \{\tilde{\pi} \circ M / \tilde{\pi} : \tilde{\mathbb{B}} \rightarrow \mathbb{A}\} \quad (23)$$

Limites

La stratégie obtenue est sous-optimale puisque $\Pi_{\tilde{\mathbb{B}}} \subset \Pi$. En revanche, l'explosion combinatoire due à la taille du vecteur de croyance est limitée, au prix d'une réduction drastique de l'information fournie lors de la prise de décision.

ii Résumé d'action

Il est possible d'effectuer une autre approximation en réduisant encore l'espace Π de recherche de la meilleure stratégie. Il s'agit cette fois de diminuer la taille de cet espace en partitionnant l'espace des actions \mathbb{A} . Pour cela on définit une fonction Ψ permettant de trouver une action $a \in \mathbb{A}$ à partir de la partition $\tilde{a} \in \tilde{\mathbb{A}}$: (\tilde{a} est appelée action résumée)

$$\begin{aligned} \Psi : \tilde{\mathbb{A}} \times \mathbb{B} &\rightarrow \mathbb{A} \\ (\tilde{a}, b) &\mapsto a = \Psi(\tilde{a}, b) \end{aligned} \quad (24)$$

On peut représenter la factorisation de π de la façon suivante :

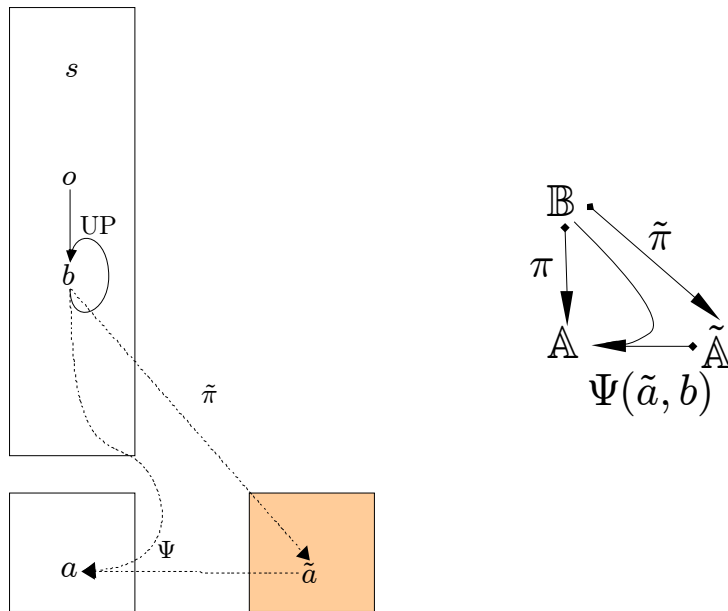


Figure 1.11: Résumé d'action : $\pi(b) = \Psi(\tilde{\pi}(b), b)$

La fonction Ψ permet de définir un ensemble de recherche restreint $\Pi^{\tilde{\mathbb{A}}}$:

$$\Pi^{\tilde{\mathbb{A}}} = \{\pi(b) = \Psi(\tilde{\pi}(b), b) / \tilde{\pi} : \mathbb{B} \rightarrow \tilde{\mathbb{A}}\} \quad (25)$$

La stratégie (sous-optimale car $\Pi^{\tilde{\mathbb{A}}} \subset \Pi$) obtenue en utilisant les actions résumées est définie par

$$\pi_{approx}(b) = \operatorname{argmax}_{\pi \in \Pi^{\tilde{\mathbb{A}}}} \mathbb{E} (V^\pi(b)) \quad (26)$$

La taille du problème est efficacement réduite par ce mécanisme. Par exemple, appliqué au cas du dialogue homme-machine, [Schatzmann et al. 2006] utilise 5 actions résumées tandis que l'espace des actions maitres peut être de taille 10 à 100 fois plus grande selon la tâche considérée.

Limites

Une première limitation est que la fonction de retour depuis l'espace résumé requiert une heuristique à définir de façon *ad hoc*. Cela nécessite une coûteuse intervention d'un expert à la fois du domaine et du système.

Une deuxième limitation est qu'une solution de repli est nécessaire lorsque l'action résumée ne peut être transformée en action maitre. Cette solution de repli fait également appel à une heuristique définie de façon *ad hoc* et entraînant la même contrainte de coût.

iii Résumés simultanés de croyance et d'action

Il s'agit simplement de réaliser simultanément les deux résumés précédents. En réalisant les deux résumés des espaces de croyance et d'action, le schéma devient :

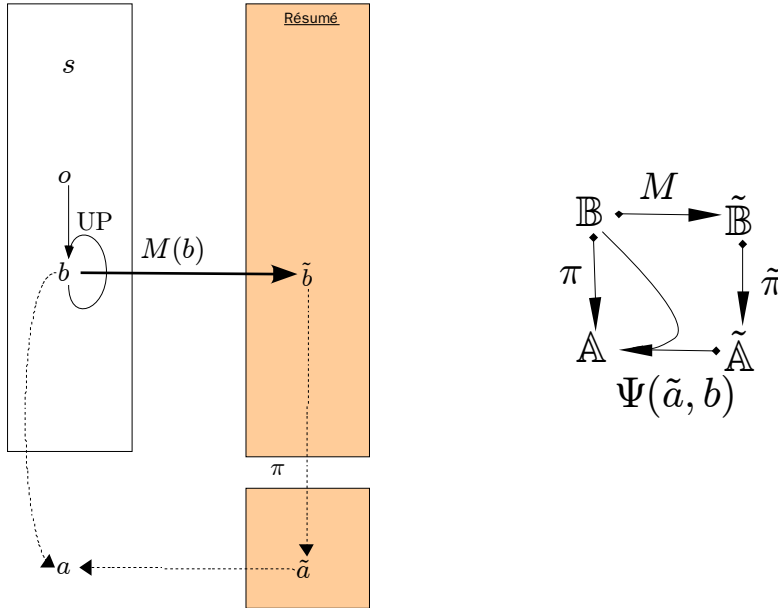


Figure 1.12 : Résumés simultanés de croyance et d'action : $\pi(b) = \Psi(\tilde{\pi}(M(b)), b)$

L'approximation de la stratégie optimale est définie comme :

$$\pi_{approx}(b) = \operatorname{argmax}_{\pi \in \Pi_{\tilde{\mathbb{B}}}^{\tilde{\mathbb{A}}}} \mathbb{E} (V^{\pi}(b)) \quad (27)$$

$$\Pi_{\tilde{\mathbb{B}}}^{\tilde{\mathbb{A}}} = \{\pi(b) = \Psi(\tilde{\pi} \circ M(b), b) / \tilde{\pi} : \tilde{\mathbb{B}} \rightarrow \tilde{\mathbb{A}}\} \quad (28)$$

1.3 Application au dialogue oral

Résumé : Les méthodes de renforcement (POMDP) peuvent être appliquées dans un système de dialogue pour effectuer une planification tenant compte de l'incertitude sur l'état de dialogue. Cependant, il est nécessaire de simplifier le modèle pour en contrôler l'explosion combinatoire se produisant lors de l'optimisation de la stratégie. Différentes simplifications ont été proposées dont celles de HIS, BUDS et du POMDP résumé.

L'utilisation de méthodes POMDP d'apprentissage par renforcement offre de multiples promesses pour les systèmes de dialogue : planification de l'évolution future du dialogue, quantification de l'incertitude, adaptabilité aux changements de l'environnement, possibilité de conception automatisée de systèmes par entraînement automatique. C'est la raison pour laquelle il nous semble intéressant de contribuer à explorer cet axe de recherche. Cette partie expose un état de l'art sur les propositions déjà faites en ce sens afin de mettre en évidence l'intérêt et les spécificités de nos travaux.

Initialement, le système déterministe de suivi de l'état d'information (ISU) [Traum et Larsson 2003] donne un cadre adapté au traitement automatique par une machine. En effet, ce paradigme, en simplifiant les différentes théories linguistiques sur lesquelles reposent les approches classiques, offre une modélisation particulièrement favorable à l'emploi de modèle d'actes de dialogue et de statuts d'ancrage, à travers les notions d'éléments élémentaires de dialogue (*dialogue moves*) et d'état d'information.

Afin d'ajouter au système la possibilité d'apprendre automatiquement une stratégie optimale, [Levin et al. 1998] a proposé l'utilisation de modèles statistiques d'apprentissage par renforcement tels que les MDP. Ces modèles permettent de plus une planification automatique quantifiée, ainsi qu'une certaine adaptabilité aux changements de conditions du dialogue. Cette piste a été longuement étudiée depuis [Walker et al. 1998; Pietquin 2004; Rieser 2008] jusqu'à conduire à la conception de système complet reposant sur un modèle MDP.

Pour obtenir des systèmes plus robustes face aux erreurs de reconnaissance vocale, [Roy et al. 2000] a proposé de tenir compte de l'incertitude sur l'état exact du dialogue. Pour cela, l'état de dialogue est modélisé par une variable cachée conduisant à faire évoluer le modèle MDP en un modèle plus complexe de POMDP. Il a été proposé différentes façons de définir le modèle POMDP [Williams et al. 2005; Zhang et al. 2001], avec des approximations différentes et des algorithmes de résolutions adaptés.

Malgré ses nombreuses qualités et possibilités, l'utilisation de POMDP pour le dialogue se heurte cependant à deux écueils majeurs : une explosion combinatoire de la taille de l'es-

pace de recherche de la stratégie optimale et un manque de contrôle sur la stratégie finale. De plus, l'utilisation d'observations sous la forme de listes *n-best* au lieu d'une observation déterministe reste difficile à traiter de façon pleinement satisfaisante car l'intégration des scores de probabilités n'entre pas strictement dans le cadre probabiliste du modèle POMDP.

Nous commencerons dans les parties 1.3.1 et 1.3.2 par exposer les principes généraux de l'usage des MDP et POMDP afin de mettre en évidence les principaux verrous technologiques en jeu (en 1.3.3). Nous aborderons ensuite plus particulièrement certaines solutions apportées aux blocages évoqués, notamment le système HIS [Young et al. 2007] qui a plus particulièrement inspiré celui que cette thèse a mis au point et testé.

1.3.1 MDP pour le dialogue

L'intégration de modèles statistiques d'apprentissage dans le gestionnaire de dialogue s'est faite progressivement. Elle n'est d'ailleurs pas terminée et malgré les recherches dans ce domaine, la question de savoir quel est le modèle le plus approprié à la gestion du dialogue n'est pas résolue.

L'intérêt initial de l'utilisation de modèles d'apprentissage par renforcement pour le dialogue est de permettre un apprentissage automatique et une planification de stratégie optimale. [Levin et al. 1998] a donc proposé d'utiliser le paradigme de processus de décision de Markov. Ce modèle a longuement été étudié depuis pour optimiser les stratégies des systèmes [Walker et al. 1998; Pietquin 2004].

Les modèles MDP ont été étudiés de manière approfondie dans d'autres domaines tels que ceux de la robotique ou des réseaux et des télécommunications. Cependant, dans notre cadre et bien que l'apport théorique de ces recherches soit considérable, l'application pratique nous éloigne fortement des modèles MDP classiques. En particulier, les dialogues homme-machine que nous considérons sont généralement très courts (horizon de l'ordre de 20) par rapport à des échanges des données entre téléphones cellulaires et antennes (horizon infini). Ainsi, la notion d'état d'équilibre de chaînes de Markov récurrentes n'est pas pertinente dans le cas qui nous préoccupe.

Parmi les points essentiels qui ont été explorés, on peut citer la question du choix de la fonction de récompense que est cruciale et qui a une très forte influence sur la stratégie optimale. Pour éviter de réaliser ce choix délicat de façon approximative, certaines méthodes [Chandramohan et al. 2010] visent à l'apprendre automatiquement en résolvant le problème de renforcement inverse (IRL¹) à partir d'une stratégie humaine. [Fard et al. 2008] s'est également intéressé à l'influence sur la stratégie finale de l'erreur commise sur le modèle de transition (dans un cadre MDP et dans un cadre POMDP, voir ci-dessous).

Un état de l'art sur l'apprentissage par renforcement est proposé dans [Kaelbling et al. 1996] ou [Sigaud et Buffet 2008]. Et [Frampton et Lemon 2009] décrit son application au dialogue.

(1) IRL : *Inverse Reinforcement Learning*.

1.3.2 Structure d'un POMDP pour le dialogue

Cette partie décrit de façon générale l'utilisation de POMDP, pour les systèmes de dialogue. On y définit les notations utilisées, les deux étapes essentielles de mise à jour de la croyance et de la prise de décision ainsi que certaines méthodes d'optimisation.

1.3.2.1 Définition de s , o et a

i État

Différentes décompositions peuvent modéliser l'état de dialogue [Roy et al. 2000; Poupart 2005]. Nous retiendrons celle de [Poupart 2005] puisqu'elle est utilisée dans les systèmes HIS et BUDS présentés ensuite. Elle vise à modéliser l'utilisateur et le dialogue en factorisant l'état de dialogue en trois composants comme illustré par la figure 1.14 :

$$s = (a_u, g, h) \tag{29}$$

a_u est l'action de utilisateur, qui peut être un fragment de formulaire associé à un acte de dialogue.

g est le but de l'utilisateur, par exemple un formulaire instancié.

h est l'historique du dialogue. Il se réduit souvent à des valeurs d'ancrage, gérées par une machine à états finis (FSM) [Traum 1994].

En utilisant la décomposition $s = (a_u, g, h)$ de l'état de la formule (29) et en posant des hypothèses d'indépendances raisonnables, le modèle du POMDP se représente selon le graphe d'influence de la figure 1.13.

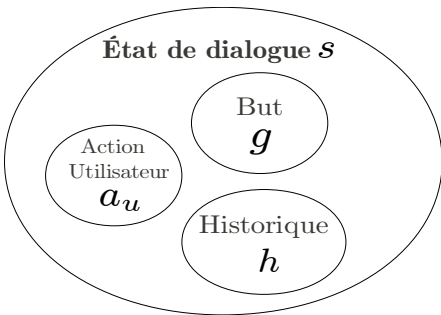


Figure 1.14: Factorisation de l'état de dialogue en trois composantes.

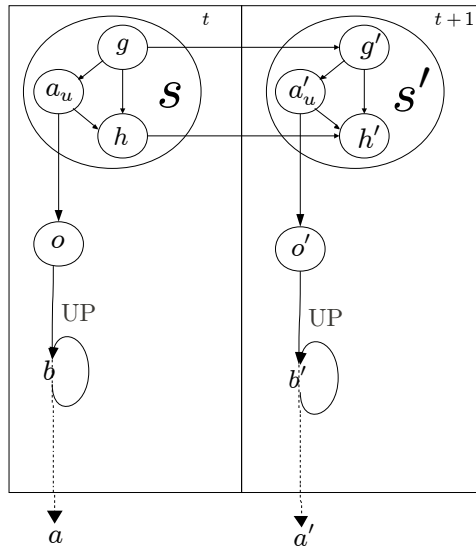


Figure 1.13: Diagramme d'influence d'un POMDP dont l'état est décomposé en trois composantes : but, action utilisateur et historique.

ii Observation

L'observation o est une liste n -best probabilisée de n actions de l'utilisateur, chacune des actions de cette liste o^i étant associée à son score de confiance p_i :

$$o = (o^{(i)}, p_i) \quad (30)$$

Comme observé précédemment, l'utilisation d'observations sous la forme de liste n -best peut conduire à des systèmes plus robustes. Cependant lorsque l'on cherche à utiliser l'information contenue dans les valeurs des scores p_i , il devient nécessaire d'obtenir des estimations fiables de ces scores de confiance. Une partie des recherches effectuées dans le domaine de la reconnaissance vocale est d'ailleurs tournée vers l'étude et la modélisation de ces scores de confiance.

iii Actions

On notera a l'action du système. a est parfaitement observée et en général non bruitée, bien que certains travaux [Pietquin 2004] conseillent de modéliser également l'erreur intervenant sur le canal de communication de sortie du système. On peut imaginer l'intérêt de cette modélisation par exemple, dans une situation de conduite automobile, lors d'un dialogue intervenant entre le conducteur et son ordinateur de bord. Lors d'une interaction orale entre un conducteur et le système embarqué, le niveau sonore lors du tour de parole du système peut permettre d'estimer le risque d'erreur de perception de l'action du système. Judicieusement intégrée dans le système, cette information peut conduire le système à répéter son action sans demande explicite de utilisateur.

iv Exemple

Dans l'exemple proposé ci-dessous, et repris par la suite, la tâche d'information touristique est simplifiée afin de pouvoir expliciter complètement les calculs de suivi de croyance. On considère que l'utilisateur fournit le nom d'une ville pour obtenir des renseignements sur cette ville. Lorsqu'il ne souhaite plus d'information, il cherche à mettre fin au dialogue. Le système cherche à identifier le nom de la ville en question, ainsi qu'à détecter la volonté de l'utilisateur de terminer le dialogue.

Cet exemple vise à illustrer l'utilisation du modèle de POMDP résumé proposé, il ne fait pas intervenir de représentation sémantique riche de l'état de dialogue. Cependant, le paradigme de POMDP résumé avec une représentation en graphes de *frames* sémantiques (présenté *infra*) s'applique de la même façon que dans l'exemple et le prototype développé intègre les mêmes mécanismes de projection résumée et double mise à jour des croyances résumée et maître.

Définissons à présent l'état et l'observation considérés dans notre exemple.

État

Dans l'exemple, l'état de dialogue se résume au but de l'utilisateur. Pour la simplicité de l'exposition, on ignorera le modèle d'ancrage représentant l'historique du dialogue. L'état

de dialogue peut alors être représenté sous la forme d'un formulaire possédant deux champs : $g = (C, F)$

- un champ ville C ayant pour valeur un nom de ville (Paris, Avignon, etc.)
- un drapeau F indiquant si l'utilisateur a exprimé son envie de terminer le dialogue (Fini ou null).

Par exemple, un état de dialogue pourra être [$\mathbf{C} = \text{« Paris »}$, $\mathbf{F} = \text{null}$], si l'utilisateur souhaite des informations sur la ville de Paris et ne souhaite pas terminer le dialogue. Si le système n'arrive pas à répondre à la demande et que l'utilisateur perd patience, l'état pourra effectuer une transition vers [$\mathbf{C} = \text{« Paris »}$, $\mathbf{F} = \text{Fini}$] : l'utilisateur souhaite toujours des informations sur la ville de Paris mais il souhaite terminer le dialogue.

On fait l'hypothèse que l'état initial du dialogue est $g = [\mathbf{C} = \text{« Paris »}$, $\mathbf{F} = \text{null}]$. On suppose de plus qu'il cherche à exprimer son but : $a_u = \text{inform}(\mathbf{C} = \text{Paris})$.

Observation

Dans cet exemple, il y a deux actes de dialogue de l'utilisateur : `inform` et `bye`. L'acte `inform(C=x)` correspond à l'expression de la valeur du champ \mathbf{C} : l'utilisateur informe le système qu'il souhaite des informations sur la ville x . L'acte `bye` correspond au désir de l'utilisateur de terminer le dialogue. Remarquons qu'on pourrait considérer l'acte `bye` comme un acte `inform(F=Fini)`. Ce choix sémantique est réalisé lors de l'implémentation du module de compréhension (SLU).

Par exemple, l'observation du tour courant peut être constituée de deux actes de dialogue, `bye` avec score de probabilité de 0,6 et `inform(C = « Avignon »)` avec score de probabilité de 0,2 :

$$o = [(\text{bye} ; 0.6), (\text{inform}(\mathbf{C} = \text{« Avignon »}) ; 0.2)] \quad (31)$$

Une telle observation signifie que l'utilisateur souhaite mettre fin au dialogue (avec probabilité 0,6) ou bien qu'il a donné la valeur d'une information pertinente pour le champ \mathbf{C} : « Avignon ». Cette valeur peut, par ailleurs, entrer en conflit avec la valeur courante du champ \mathbf{C} mais ceci est l'affaire du modèle d'ancrage qui n'est pas le sujet de cette section.

Notons que la somme des scores de la liste *n-best* n'est pas 1.0 car la liste n'est pas exhaustive mais se limite au deux actes de dialogue les plus probables (lors de l'intégration dans le vecteur de croyance, ces scores sont cependant renormalisés).

1.3.2.2 Mise à jour de la croyance

i Généralités

L'intégration d'une nouvelle observation dans l'estimation de l'état courant de dialogue constitue une étape essentielle du traitement effectué par le gestionnaire de dialogue. Dans un POMDP, cette étape est réalisée en utilisant l'observation o pour la mise à jour du vecteur de croyance b . Elle est effectuée à partir de la formule bayésienne (12) répétée ci-dessous :

$$b'(s') = k.P(o'|s') \sum_{s \in \mathbb{S}} P(s'|a, s) b(s) \quad (12)$$

Si l'on dispose d'un modèle de probabilité connu indiquant les dépendances $P(o|s)$ et $P(s'|a, s)$, cette formule peut-être utilisée directement pour la mise à jour du vecteur de croyance. Mais on se heurte à une difficulté pour apprendre le modèle complet sur les données : sa taille gigantesque requiert une quantité importante de données.

On peut cependant décomposer la formule de mise à jour en faisant des hypothèses sur la structure des états, des observations et sur les dépendances de leurs composants. C'est par exemple ce qui est réalisé dans le système BUDS (décrit *infra*) en décomposant la suite cachée des états de dialogue en un réseau bayésien dynamique. La mise à jour de la croyance peut alors être effectuée par des algorithmes propres aux réseaux bayésiens dynamiques.

ii Décomposition du modèle

En utilisant la décomposition du modèle bayésien de la figure 1.13, la formule de mise à jour de la croyance (12bis) se décompose en :

$$b'(a_u', g', h') = k.P(o'|a_u') P(a_u'|g', a) \sum_g P(g'|g, a) \sum_h P(h'|g', a_u', h, a) b(s) \quad (32)$$

Les différents modèles correspondant à chacune des probabilités conditionnelles de la formule (32) sont les modèles d'observation $P(o'|a_u')$, d'action utilisateur $P(a_u'|g', a)$, de but utilisateur $P(g'|g, a)$, et d'historique de dialogue $P(h'|g', a_u', h, a)$. Nous verrons que certaines hypothèses permettent de les définir de manière *ad hoc* ou par apprentissage statistique.

1.3.2.3 Optimisation de la stratégie

L'utilisation de variables cachées et la taille du vecteur de croyance qui en découle rend difficile le calcul de la stratégie optimale. Les différents algorithmes de résolution utilisés sont des algorithmes avec ou sans utilisation du modèle de probabilité de transition et d'observation. Selon le modèle sémantique utilisé ou les données dont on dispose, certains algorithmes sont plus adaptés que d'autres.

Ainsi, l'algorithme de critique d'acteur naturel (NAC¹) correspond bien à la représentation factorisée de l'état proposé par BUDS [Thomson et al. 2008] et peut être mis en œuvre en utilisant des fonctions d'extraction de caractéristiques (*features*) pour chaque champ du formulaire (un champ correspondant à un nœud du graphe bayésien).

Aussi, lorsqu'on dispose des données d'un corpus permettant de calculer une estimation de ce modèle, il est possible d'utiliser les algorithmes avec modèle, PBVI [Pinault et al. 2009] ou Perseus[Spaan et Vlassis 2005].

Quelque soit l'algorithme choisi, une difficulté commune provient du fait qu'il est *a priori* nécessaire de disposer de chercher une stratégie optimale dans un ensemble gigan-

(1) NAC : *Natural Actor Critic*.

tesque : l'ensemble des stratégies dépendant du vecteur de croyance complet dont la taille est celle de l'espace de croyance (et par conséquent continu). Il est possible d'éviter les problèmes dus à cette représentation exhaustive tabulée en utilisant les méthodes de représentations plus compactes de la stratégie qui sont décrites *infra*.

1.3.3 Verrous technologiques

Malgré les avancées certaines dues à l'utilisation de modèles statistiques dans différents domaines du TALN, certains points s'opposent à l'adoption de modèles statistiques POMDP dans les systèmes de dialogue actuels : une explosion combinatoire empêchant la résolution exacte, une difficulté à intégrer de façon théoriquement fondée les scores d'une observation *n-best* et un manque de contrôle de sur stratégie finale.

Explosion combinatoire

L'explosion combinatoire qui apparaît lors de la recherche de la stratégie optimale d'un POMDP a différentes causes :

- La taille de l'espace d'état est généralement très importante.
- Le nombre d'actions peut être très grand également.
- La croyance est une distribution sur l'espace d'état. Le cardinal de l'espace de croyance (après discrétisation) croît donc de façon exponentielle selon le nombre d'états.

De plus, lors de la planification, la longueur de l'historique ou la taille du vecteur de croyance rend impossible l'exploration complète de toutes les stratégies.

Intégration de scores *n-best*

Afin de profiter pleinement de l'apport des méthodes statistiques, il est nécessaire de pouvoir intégrer de façon théoriquement fondée des observations sous la forme d'une liste *n-best* associée à des scores de confiance. Cependant, le modèle POMDP classique ne modélise l'observation sous la forme d'une variable déterministe et non d'une liste de valeurs associées à des scores de probabilités. Cette intégration présente donc des difficultés tant du point de vue de sa justification théorique que de son implémentation pratique.

Incomplétude de la conception

La nature même d'un système statistique implique une notion d'optimalité statistique qui ne permet pas d'assurer un contrôle de la qualité du système obtenu. Comme le remarque [Pieraccini et Huerta 2005], la complétude de la conception de l'interface utilisateur (VUID¹) n'est pas assurée avec un POMDP : aucun expert humain n'a envisagé chaque situation afin de définir l'action optimale. Comme on ne peut pas être entièrement certain qu'un système entraîné par renforcement ne prendra jamais une action totalement aberrante, l'industrie est hésitante à adopter cette technologie dont elle ne maîtrise pas les risques.

(1) VUID : *Voice User Interface Design*.

1.3.4 Solutions proposées

Pour réduire l'explosion combinatoire, comme le suggère [Poupart 2005], l'utilisation de la structure de l'état de dialogue permet de simplifier le modèle. Les méthodes générales applicables à tout POMDP sont bien entendu utilisables dans le cas particulier des systèmes de dialogue oral homme-machine : POMDP résumé [Williams et Young 2005], POMDP hiérarchique [Barto et Mahadevan 2003; Pineau et al. 2003], POMDP factorisé [Sim et al. 2008], POMDP en logique du premier ordre [Wang et al. 2007]. Cependant, la modélisation d'un système de dialogue possède des propriétés qui lui sont spécifiques dont on peut chercher à tirer parti.

La réduction des tailles des espaces d'état, d'observation et de croyance est une piste de recherche essentielle. Ainsi [Williams et Young 2006] propose la création d'un POMDP résumé, [Lefèvre et de Mori 2007] propose d'utiliser des méthodes de clustering automatique visant également à obtenir un POMDP plus simple, [Young et al. 2007; Young et al. 2010] suggère l'utilisation d'une représentation compacte de la croyance suivie d'un résumé de croyance (HIS) et [Williams et Young 2006] une décomposition du POMDP maître en de multiples MDP (CSPBVI).

Il est également possible de faire des hypothèses sur la dynamique du POMDP, en utilisant par exemple un réseau bayésien dynamique pour les transitions d'état et d'observation (BUDS) [Thomson et Young 2010; Hoey et al. 2009].

L'intégration d'informations statistiques telles que les scores de confiance d'une liste *n-best* a été traitée par mise à jour d'un réseau bayésien (BUDS) ou bien directement dans l'étape de mise à jour de la croyance (HIS).

Concernant la complétude de conception (VUID-complétude), on a envisagé une association de la stratégie POMDP avec une deuxième stratégie plus contrôlable : à base de règles logiques [Williams 2008a] ou d'apprentissage supervisé [Henderson et al. 2008], ou en limitant l'influence de la stratégie POMDP pour réaliser une conception assistée par POMDP plutôt qu'une conception complètement automatisée [Laroche et al. 2009]. Cependant la complétude de conception reste actuellement difficile à assurer car la conception de la stratégie devenant au moins partiellement automatisée, elle échappe à un contrôle exhaustif.

Certaines de ces solutions semblent prometteuses et ont inspirés en partie nos travaux, notamment les propositions de POMDP résumé [Williams et Young 2005], de HIS [Young et al. 2010] et de BUDS [Thomson et al. 2008]. Nous les détaillons ci-dessous.

1.3.4.1 POMDP résumé de Williams

Dans [Williams et Young 2005], les auteurs ont proposé de créer un deuxième POMDP, appelé POMDP résumé. Les tailles des espaces d'état et d'observation de ce POMDP résumé sont réduites afin de minimiser l'explosion combinatoire intervenant dans le POMDP maître.

Les figures 1.15 et 1.16 représentent le fonctionnement d'un système reposant sur ce paradigme dont les mécanismes (mise à jour de la croyance dans les espaces maîtres, résumés d'espaces, mise à jour de la croyance dans les espaces résumés), sont décrits ensuite. Les limites de ce modèle sont principalement dues à l'utilisation de deux mécanismes de mise à jour différents ainsi que de fonctions de résumé qui doivent nécessairement être cohérentes entre elles.

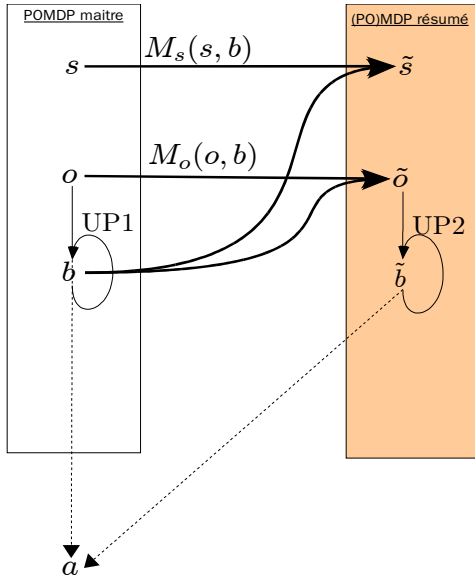


Figure 1.15: Modèle de POMDP résumé de Williams lors de l'optimisation : un résumé de l'état et de l'observation.

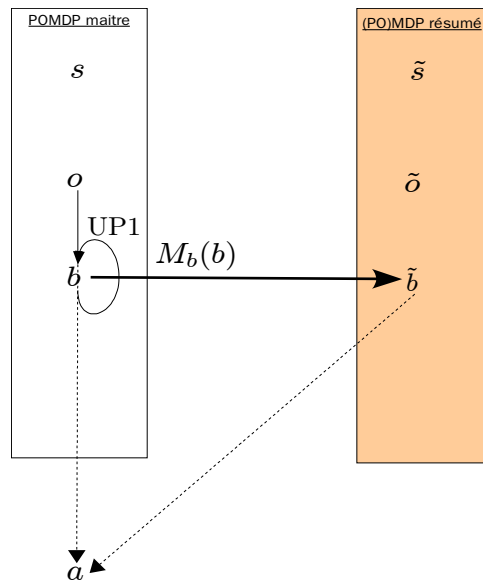


Figure 1.16 : Modèle de POMDP résumé de Williams lors de l'exécution : un résumé de la croyance.

i Première mise à jour de la croyance

Le suivi de croyance est toujours effectué dans l'espace maître du POMDP initial. Pour cela l'état est décomposé en $s = (a_u, g, h)$ selon la formule (29), certaines dépendances sont fixées de manière *ad hoc* tandis que d'autres sont apprises d'après de véritables dialogues.

ii Résumés

Les variables maîtres s , o et b sont ensuite projetées dans les espaces résumés à l'aide de fonctions résumées prédéfinies M_s , M_o et M_b . Ces fonctions sont définies de façon à extraire de s , o et b l'information pertinente pour une prise de décision. Elles s'intéressent en particulier à mettre en évidence les conflits entre s et b ou entre o et b concernant les valeurs instanciées dans les champs.

iii Seconde mise à jour de la croyance

Lors de l'optimisation, les états et les actions résumés sont définis à l'aide de fonctions *ad hoc*. Dans le modèle POMDP résumé, l'optimisation peut être réalisée de façon classique : en utilisant un algorithme de résolution POMDP sur modèle complet, ce dernier est obtenu à partir d'un corpus de dialogues ou par simulation.

En revanche lors de l'exécution, on réalise la mise à jour de la croyance dans l'espace maître et une autre fonction de résumé de croyance M_b permet de la projeter dans l'espace résumé.

iv Prise de décision

La prise de décision sur la meilleure action a à prendre est effectuée en utilisant la croyance résumée selon la stratégie résumée optimisée $\tilde{\pi}$.

v Limites

De multiples systèmes

Lors de l'optimisation et lors de l'exécution, les conditions de mise à jour sont différentes. En effet, ainsi, la croyance résumée (\tilde{b} dans le POMDP résumé) est calculée de deux façons différentes. Or, pour que l'optimisation de stratégie ait un sens, le système lors de l'apprentissage doit être suffisamment similaire au système utilisé lors de l'exécution. Par conséquent, il est nécessaire de s'assurer que ces deux façons de réaliser le suivi de croyance sont cohérentes. Cela implique que les trois fonctions résumées M_s , M_o et M_b doivent être conçues de manière à ce que les deux suivis de croyance aient des comportements similaires. Dans le cas contraire, l'optimisation (réalisée en utilisant M_s et M_o) n'est pas optimale lors de l'exécution (qui utilise M_b).

De plus, le modèle de probabilité du POMDP résumé (probabilités de transitions et d'observations) est appris d'après des données. Or, lors de la génération de ces données, un autre système a été utilisé, ce dernier ne suivant pas la stratégie optimale. Malheureusement, rien ne garantit que le modèle de probabilité résumé reste valable lorsque la stratégie est différente.

Fonctions résumées M_s , M_o et M_b

Concernant la conception de ces fonctions résumées M_s , M_o et M_b , elles sont génériques et ne nécessitent pas d'intervention du concepteur pour être adaptées à un nouveau domaine. Cependant, puisqu'elles nécessitent une modélisation du but utilisateur par un formulaire, elles ne peuvent être appliquées directement dans le cadre d'une modélisation sémantique plus riche telle que celle des graphes de *frames* sémantiques décrite en partie 1.1.2.4. On pourrait donc souhaiter utiliser une méthode plus automatisée pour concevoir ces fonctions résumées, c'est l'une de nos propositions de résumé par *clustering* automatique.

[Williams et Young 2005] montre que leur approche réduit la complexité du problème de M^N à 2^N , N étant le nombre de champs du formulaire utilisé et M le nombre de valeurs possibles de chaque champ. Cette réduction est loin d'être négligeable et peut être suffisante pour certaines tâches. Il serait souhaitable cependant de pouvoir réduire encore la taille du POMDP résumé et de pouvoir contrôler cette réduction de taille. L'utilisation de méthodes de *clustering* automatique permet également de résoudre cette limitation puisque le nombre de *clusters* peut être choisi selon la précision de résumé souhaitée.

1.3.4.2 État d'information caché (HIS)

Avec l'état d'information caché (HIS¹) [Young et al. 2007], il est proposé d'utiliser le paradigme d'état d'information (ISU) [Traum et Larsson 2003] en le considérant comme une variable cachée sur laquelle est opéré un suivi de croyance.

(1) HIS : *Hidden Information State*.

La proposition de HIS pour réduire la complexité du modèle POMDP consiste en deux points : un partitionnement variable de l'espace d'état (en i) et un résumé de croyance (en iii). Afin de suivre la chronologie du processus de traitement d'un tour de parole par le système, nous présentons, intercalé entre ces deux points (en ii), l'intégration des valeurs des scores de probabilité des *n-best*. Celle-ci est réalisée directement dans le modèle de mise à jour de la croyance maître, ce qui permet d'utiliser cette information de manière statistique.

i Partitionnement variable de l'espace d'état

Le suivi de croyance est effectué exclusivement dans l'espace maître. Afin de représenter de façon compacte cette croyance, HIS propose de réaliser une partition de \mathbb{S} l'espace d'état, guidée par une ontologie propre au domaine. Ainsi, cette solution vise à contrôler l'explosion combinatoire due à la taille du vecteur croyance en le représentant à l'aide de partitions de l'espace des états \mathbb{S} .

Rappelons que le vecteur de croyance b est une distribution sur \mathbb{S} , c'est à dire un vecteur indexé par les éléments $s \in \mathbb{S}$. Afin de représenter b de façon compacte, on utilise une partition (ou partitionnement) de \mathbb{S} qui permet de regrouper les états dans différentes classes. Nous utiliserons ici la définition classique d'une partition d'un ensemble \mathbb{S} : une partition de \mathbb{S} est un ensemble de classes (c'est à dire des sous-ensembles) non vides de \mathbb{S} , deux à deux disjointes et dont l'union recouvre \mathbb{S} . Cette utilisation du terme « partition » diffère de celle des publications originales de [Young et al. 2007; Young et al. 2010] qui considèrent que chaque classe d'état est une partition¹. Pour que la partition de l'espace d'état utilisée soit appropriée à la représentation de la croyance courante, elle est générée en accord avec les actes de dialogue des interlocuteurs (l'utilisateur et le système). Elle utilise de plus comme modèle une structure prédéfinie appelée « ontologie du domaine », représentée figure 1.3. La partition est alors adaptée à la croyance b et on constate que les états regroupés dans une même classe possèdent la même valeur de croyance. Notons que des états de même valeur de croyance ne sont pas nécessairement dans la même classe.

Par exemple, comme illustré dans la figure 1.17, au début du dialogue, la partition initiale est réduite à un seul élément regroupant tous les états possibles. Puis, l'observation d'un tour de parole de l'utilisateur conduit à raffiner cette partition selon quatre classes. Ainsi, lors de l'étape de mise à jour du vecteur de croyance, la représentation compacte de b devient moins compacte. La partition de l'espace d'états se raffine pour pouvoir représenter un nombre plus grand de groupes d'état de dialogue différents. Si le dialogue se poursuit, on est confronté à l'explosion combinatoire du nombre de classes de la partition. Par conséquent, il est nécessaire de supprimer une partie de la partition : une étape d'élagage (*pruning*) est nécessaire pour supprimer les classes d'états correspondant à des valeurs de croyance ne dépassant pas un certain seuil.

Notons que le processus de partitionnement consiste à utiliser l'ontologie de la tâche pour représenter la croyance de façon compacte. En principe, cette représentation de la croyance est sans perte, même si un élagage a lieu lors de toute implémentation. Il ne s'agit pas d'extraire (automatiquement ou non, de façon *ad hoc* ou générique) l'information perti-

(1) Cet emploi du terme « partition » pour parler d'une classe n'est cependant pas réellement abusif puisque chacune des classes d'une partition définit également une partition binaire.

nente pour la prise de décision de la machine mais de simplifier la représentation informa-

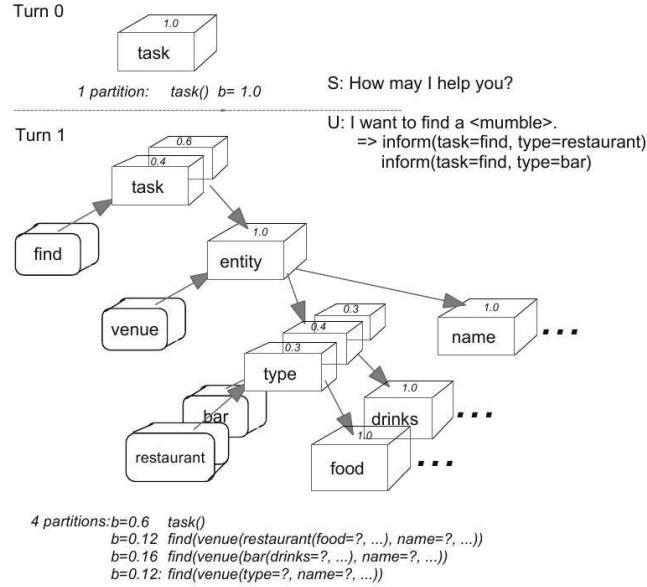


Figure 1.17: Raffinement de la partition utilisée pour représenter la croyance dans le système HIS (extrait de [Young et al. 2010]).

tique d'un vecteur de grande taille.

Cependant, malgré l'élagage déjà effectué, la taille du vecteur de croyance b reste trop grande pour effectuer une véritable optimisation POMDP (c'est-à-dire une optimisation MDP en considérant le vecteur de croyance comme état). Une nouvelle simplification s'impose, par exemple un résumé de croyance (voir iii).

ii Mise à jour de la croyance et intégration de n-best

Sur l'espace d'état maître partitionné en classes d'état, la valeur de croyance est, par construction de la partition, la même pour tous les états d'une même classe ; la mise à jour de la croyance maître est alors effectuée efficacement en regroupant les calculs identique d'états d'un même classe. Lors de cette étape, un modèle statistique qui permet d'intégrer les scores de confiance de la liste d'observation n -best.

Dans ce modèle, la modélisation de l'action utilisateur $P(a_u' | g', a)$ intègre la notion de paire adjacente d'actes de dialogue correspondant à un modèle de bigramme. Les deux autres modèles peuvent être définis de façon *ad hoc* mais indépendante du domaine. Par exemple, le modèle du but utilisateur est déterministe¹ et tel qu'une action déterminée (findalt : chercher une solution alternative) déclenche un changement de but. Le modèle d'observation $P(o' | a_u')$ permet d'intégrer l'information quantitative que constituent les scores de confiance de la liste n -best d'observation. Une proposition en ce sens a été faite dans HIS [Young et al. 2010].

(1) Modèle déterministe : c'est-à-dire que les probabilités du modèle valent 0 ou 1.

L'intégration des *n-best* peut être effectuée lors de la mise à jour du vecteur de croyance en utilisant les scores de confiance pour estimer le modèle d'observation $P(o'|a_u')$ moyennant deux approximations.

Scores de confiance fiables

Rappelons que l'observation o est constituée d'une liste de *n-best* associés à leurs scores, chaque acte de dialogue $o^{(i)}$ est associé à son score de confiance p_i :

$$o = (o^{(i)}, p_i) \quad (33)$$

L'événement « $o^{(i)}$ est la véritable action de l'utilisateur » peut se noter $\{o^{(i)} = a_u\}$.

On peut supposer que le score de confiance p_i du i -ième *n-best* fournit une bonne estimation de la probabilité pour que ce *n-best* soit la véritable action utilisateur a_u sachant o :

$$P(\text{« } o^{(i)} \text{ est la véritable action de l'utilisateur »} | o) = P(o^{(i)} = a_u | o) = p_i \quad (34)$$

Notons qu'on a sur les p_i l'inégalité suivante :

$$\sum_{i=1}^n p_i \leq 1 \quad (35)$$

avec égalité si la liste *n-best* était exhaustive ($n = \infty$)¹.

Approximation de la formule Bayésienne

En utilisant la formule de Bayes,

$$P(o | o^{(i)} = a_u) = \frac{P(o)}{P(o^{(i)} = a_u)} \cdot P(o^{(i)} = a_u | o) \quad (36)$$

$$= \frac{P(o)}{P(o^{(i)} = a_u)} \cdot p_i \quad (37)$$

$$\approx k^0 p_i \quad (38)$$

Le modèle d'observation est ainsi approximé en utilisant les scores *n-best* selon la formule suivante :

$$P(o | a_u) \approx k^0 p_i \quad (39)$$

Notons que la constante k^0 se simplifiant dans le calcul de la mise à jour de la croyance, elle peut être ignorée.

Finalement, les hypothèses faites dans le système HIS permettent d'intégrer les scores *n-best* dans la formule de mise à jour (32) reprise ci-dessous :

(1) Parce que $\sum_i p_i = \sum_{\alpha \in \{o^{(1)}, \dots, o^{(n)}\}} P(a_u = \alpha | o) \leq \sum_{\alpha \in \mathbb{A}} P(a_u = \alpha | o) = 1$.

$$b'(a_u', g', h') = k \cdot P(o'|a_u') P(a_u'|g', a) \sum_g P(g'|g, a) \sum_h P(h'|g', a_u', h, a) b(s) \quad (32\text{bis})$$

$$b'(a_u', g', h') = k \cdot k^0 \cdot p_i P(a_u'|g', a) \sum_g P(g'|g, a) \sum_h P(h'|g', a_u', h, a) b(s) \quad (40)$$

Notons que le modèle d'action de l'utilisateur peut être appris sur un corpus annoté et assurer ainsi un réordonnement de la liste n -best.

Analyse

Au delà des formules mathématiques de mise à jour, le point essentiel est ici d'explorer l'utilisation des scores n -best lors du calcul de la croyance du système. L'amélioration des systèmes de dialogue passe nécessairement par une intégration quantitative et mathématiquement fondée de cette information sur l'incertitude de l'action de l'utilisateur.

iii Résumé de croyance

Considérant le POMDP maître comme un MDP sur la croyance maître b , un résumé de croyance est effectué par une fonction résumée (notée M_o). Lors de l'optimisation, on cherche une stratégie optimale $\tilde{\pi}$ dans le MDP résumé dont l'état est $\tilde{s} = M_o(b)$. Cependant, les transitions d'état du MDP résumé sont réalisées par une mise à jour de la croyance dans l'espace maître. En ce qui concerne le suivi de croyance, on reste très proche du véritable paradigme de POMDP, car ce mécanisme n'est en rien modifié par la création du modèle résumé.

En revanche, lors de l'optimisation de la stratégie, l'espace de recherche est réduit aux stratégies résumées et l'optimisation est réalisée par des algorithmes propres aux MDP. La stratégie résumée optimale dépend de l'état \tilde{s} du POMDP résumé et non d'un véritable vecteur de croyance.

Le système intègre également le classique résumé d'action présenté en partie 1.2.6.

iv Discussion

Bien que le système HIS utilise les techniques de renforcement à la pointe de l'état de l'art actuel et propose une implémentation efficace, la recherche dans ce domaine est loin d'être terminée. Certains des choix réalisés sur la façon de procéder à la mise à jour de la croyance ou à l'optimisation de stratégie ne sont pas uniques et ces mécanismes mériteraient d'être étudiés selon des angles de vue différents.

Remarquons que l'explosion combinatoire n'est pas totalement enrayée par l'utilisation de partitions de l'espace d'état. Elle a tout de même lieu puisqu'un élagage et un résumé de croyance sont nécessaires ensuite.

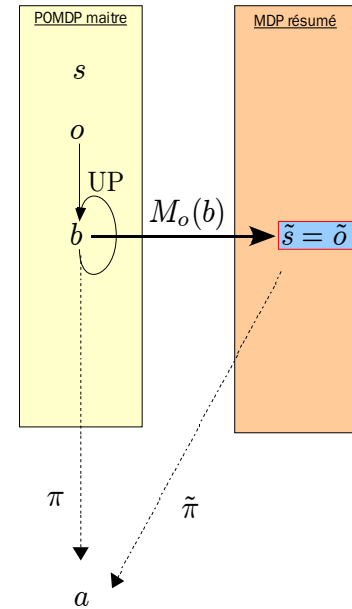


Figure 1.18: Résumé de croyance avec le système HIS

D'autre part, l'optimisation est finalement réalisée sur un MDP. Aucun suivi de croyance n'est réalisé au niveau résumé puisqu'il a déjà eu lieu dans l'espace maître. Ainsi, lors de l'optimisation, le modèle implicitement perçu par le système statistique est celui d'un MDP sur un état particulier constitué par le vecteur de croyance résumé.

Notons que, contrairement aux autres systèmes qui reposent sur une représentation du but utilisateur ou de l'état de dialogue sous la forme d'un ensemble prédéfini de champ-valeur, le paradigme HIS pourrait être étendu à des représentations plus flexibles dont la structure serait élaborée au cours du dialogue.

La proposition HIS peut être utilisée comme une référence en ce qui concerne l'utilisation de méthodes par renforcement POMDP pour le dialogue.

1.3.4.3 Mise à jour par réseau bayésien

Dans la même optique, le système BUDS¹ [Thomson et Young 2010] propose d'utiliser une modélisation de l'état de dialogue par un réseau bayésien. Il est particulièrement intéressant à considérer car il fournit une généralisation du paradigme à HIS dans un cadre théorique très comparable.

Dans cette approche, l'explosion combinatoire est traitée différemment : au lieu de réduire l'espace de recherche des stratégies en résumant le vecteur de croyance, certaines hypothèses supplémentaires sont faites sur la structure de la croyance ainsi que sur la dynamique de la mise à jour de la croyance pour que l'étape de mise à jour de la croyance soit réalisée de façon plus efficace.

L'utilisation de réseaux bayésiens dynamiques (DBN²) a été proposée par [Hoey et al. 2007; Hoey et al. 2009] qui ont développé un système d'aide aux personnes âgées utilisant un modèle POMDP. Dans cette première proposition, le modèle repose sur un DBN comportant 9 nœuds d'états et 3 nœuds d'observations.

i Mise à jour de la croyance

Le DBN du système BUDS est défini de la façon suivante :

- On fait l'hypothèse, essentielle, que l'état peut être représenté par un ensemble de champs. C'est par exemple le cas avec un modèle du formulaire qui comporte des couples champ-valeur représentant le but de l'utilisateur. Un DBN est un réseau bayésien (BN) dynamique : il est constitué d'une suite de BN indexée par t , correspondant à la suite des états s_t .
- Les dépendances entre les nœuds du réseau sont définies manuellement de façon *ad-hoc*, par exemple selon une ontologie du domaine.

La distribution de probabilités de chaque tranche de date t du réseau bayésien est le vecteur de croyance à la date t . Ensuite, la mise à jour de la croyance est effectuée selon des algorithmes classiques sur les réseaux bayésiens [Jensen et Nielsen 2007]. Notons que, par exemple en utilisant l'algorithme (non-exact) de *loopy belief propagation*, cette étape de mise à jour donne lieu à des calculs dont la complexité calculatoire peut poser problème. Il est possible de réduire cette complexité, premièrement en limitant la taille de l'historique

(1) BUDS : *Bayesian Update of Dialogue State*.

(2) DBN : *Dynamic Bayesian Network*.

considéré pour réduire la taille du DBN considéré. Deuxièmement, la propagation des messages lors d'une itération de *loopy belief propagation* requiert une sommation sur l'espace des états. En utilisant une idée similaire à celle des partitions de HIS, ce calcul peut être factorisé selon une partition et réalisé par une sommation sur des groupes d'états, ce qui réduit significativement la complexité de l'algorithme.

ii Optimisation de la stratégie

Après que l'étape de mise à jour de la croyance a été réalisée comme décrit ci-dessus, la question de l'optimisation de la stratégie peut être résolue par des algorithmes propres aux MDP, tels que SARSA, TD ou NAC. L'algorithme de NAC¹ [Peters et al. 2005; Peters et Schaal 2008] est particulièrement approprié du fait de la nature factorisée de l'état de dialogue et possède de bonnes propriétés de convergence.

Dans [Thomson et al. 2008], l'espace de recherche des stratégies possibles est indexé par un paramètre θ grâce à un ensemble de fonctions de bases $(\phi_a)_{a \in \mathbb{A}}$ en utilisant une fonction *softmax*. (équation (41)). La valeur du θ optimale est alors trouvée en utilisant une méthode de descente de gradient appropriée.

$$\pi(a|b, \theta) = \frac{\exp^{\theta \cdot \phi_a(b)}}{\sum_{a'} \exp^{\theta \cdot \phi_{a'}(b)}} \quad (41)$$

La conception des fonction ϕ_a est réalisée par un expert du domaine et est définie de manière *ad hoc*.

iii Discussion

L'approche BUDS offre un cadre théorique pour effectuer un suivi de la croyance dans l'espace maître. Elle fournit une solution effective à ce problème bien que les relations de dépendance entre les nœuds du réseau bayésien doivent être définies de manière *ad hoc*.

L'optimisation de stratégie peut ensuite être effectuée comme sur un MDP en utilisant la croyance en tant que variable d'état. De même que pour le système HIS, l'optimisation est réalisée dans un espace de recherche réduit, dans le cadre de BUDS, il s'agit des fonctions s'exprimant selon (41). Les résultats positifs présentés dans [Thomson 2009] tendent à montrer que de bonnes stratégies peuvent être trouvées dans cet espace de recherche.

Enfin, cette solution repose, elle encore, sur une hypothèse de représentation de l'état de dialogue factorisé en un ensemble prédéfini de champ-valeur (*slots*), cette représentation étant généralement associée à un modèle de formulaire.

1.3.4.4 Décomposition en plusieurs MDP (CSPBVI)

La proposition de résumé composite CSPBVI² [Williams et Young 2006] consiste à décomposer le POMDP maître en un ensemble de MDP sur des résumés de croyance et d'action similaires à ceux décrits dans la partie 1.2.6. Pour cela, un MDP résumé est défini pour chaque champ de l'état de dialogue qui est représenté par un formulaire. Chaque MDP

(1) NAC : *Natural Actor Critic*.

(2) CSPBVI : *Composite Summary Point Based Value Iteration*.

propose alors une action résumée qui est transformée en action maître. Il y a donc autant d'actions maîtres proposées par les MDP résumés que de champs dans l'état de dialogue. Une heuristique permet de choisir la plus appropriée.

Cette proposition permet de réduire drastiquement la taille des modèles à résoudre, cependant elle repose en dernier lieu sur une heuristique difficile à mettre en œuvre et à maintenir.

1.3.4.5 Couplage d'un POMDP avec un système par règles

Certains travaux ont été entrepris afin de coupler un système reposant sur des règles et un système appris par renforcement. On pourra en particulier se référer à [Williams 2008a] qui propose un couplage en deux niveaux. D'abord, un premier système utilise des règles logiques déterministes pour générer une liste d'actions possibles. Ensuite, un système POMDP considère cette liste comme observation pour choisir la meilleure action. La stratégie POMDP est donc fondée sur la liste des actions possibles et le système par renforcement intervient après l'application des règles logiques.

Pour associer deux stratégies de dialogue, une autre solution [Henderson et al. 2008] consiste à utiliser dans la mesure du possible une stratégie apprise par un apprentissage supervisé et à n'utiliser la stratégie optimisée par renforcement que lorsque les données d'apprentissage de la première stratégie sont insuffisantes. Cette approche permet de palier un certain manque de données mais ne résout pas la question de la complétude de conception de l'interface (VUID-complétude [Pieraccini et Huerta 2005]).

1.3.4.6 Autres approches

D'autres travaux se rapprochent de notre proposition ou des systèmes exposés ci-dessus. Ils sont trop nombreux pour être tous détaillés ici. Mentionnons [Cuayáhuitl et al. 2005] [Cuayáhuitl 2009] qui utilise un utilisateur simulé pour optimiser la stratégie du système. Le modèle sous-jacent généralise le modèle de chaîne de Markov cachée (HMM) et l'apprentissage est effectué par renforcement hiérarchique.

Notons également [Henderson et Lemon 2008] qui propose d'utiliser un ensemble de MDP pour approximer le POMDP trop difficile à calculer et [Laroche 2010] qui propose d'utiliser un processus de décision non-Markovien.

1.4 Simulation

Résumé : L'entraînement de systèmes statistiques requiert une quantité de données importante. De plus la nature interactive du dialogue pose le problème du mode d'obtention des données. L'utilisation d'environnements de simulation permet de générer des dialogues avec un système pour permettre l'entraînement automatique du gestionnaire de dialogue.

Toutes les méthodes statistiques ont besoin de données. Idéalement ces données sont issues de situations réelles et sont éventuellement enrichies d'annotations réalisées par des experts, de nature à permettre un apprentissage statistique supervisé.

Ceci est effectué lorsque l'on utilise un corpus réel d'interaction avec un prototype ou que l'on génère des corpus à l'aide de techniques telles que celle du magicien d'Oz. Cependant, la stratégie du système utilisée dans le corpus est alors différente de la stratégie optimisée. Cela conduit à penser que le modèle appris sur le corpus reflète imparfaitement la dynamique des dialogues avec un système optimisé.

La nature interactive du processus de dialogue pousse à considérer l'utilisation d'une autre source de données plus artificielle, impliquant un utilisateur simulé et plus généralement un environnement de simulation. La simulation d'utilisateurs permet d'une part l'évaluation d'un système : valider le fonctionnement d'un système (*proof-of-concept*) et de l'évaluer face à d'autres systèmes. D'autre part, elle joue un rôle de premier plan lors de la conception de systèmes : il s'agit de permettre l'entraînement sur un grand nombre de dialogues non disponibles dans un véritable corpus de dialogues homme-machine.

La contrepartie pour un tel contrôle des données d'apprentissage pour nos systèmes est cependant élevé : ce sont des données artificielles et par nature différentes de celles obtenues avec de véritables utilisateurs en situation réelle. C'est pourquoi de nombreux moyens ont été mis en place pour s'assurer de la modélisation la plus réaliste de tous les mécanismes présents dans un dialogue réel.

Après quelques considérations générales sur ce qui constitue le niveau de détail d'un simulateur (1.4.1), nous présentons certaines méthodes pour modéliser le comportement de l'utilisateur (1.4.2), ainsi que le bruit dû aux erreurs de reconnaissance et de compréhension (1.4.3) et les autres éléments de l'environnement (1.4.4).

1.4.1 Niveau de granularité de la simulation

La simulation d'utilisateur peut s'effectuer à différents niveaux de granularité : résumé, intentionnel, acoustique.

Il est possible de s'approcher très finement de la réalité en modélisant jusqu'aux mots prononcés par un utilisateur, voire jusqu'au signal acoustique. Certains travaux ont déjà commencé à explorer cette direction comme [Möller et Ward 2008]. C'est une solution qui

permet d'ajouter un bruit quasi-réel. Ce serait le niveau le plus proche de la réalité, à condition que les mécanismes de génération du texte (NLG) et de synthèse vocale (TTS) soient effectivement réalistes. De grands progrès ont été réalisés concernant la qualité finale du signal obtenu, en particulier pour le TTS, et le résultat est parfois assez proche pour tromper un humain. Cependant, les outils de génération ont du mal à modéliser la variabilité des locuteurs humains et un système de dialogue statistique risque de découvrir rapidement les invariants statistiques inhérents au système de génération et de se ramener *de facto* au niveau intentionnel.

On peut également se situer au niveau intentionnel (c'est celui de la représentation en *frames* sémantiques présentée *supra*). Se placer à ce niveau est un bon compromis entre le contrôle sur les expériences et le réalisme des dialogues simulés. On peut y ajouter sans difficulté majeure un modèle de bruit, adapté de façon *ad hoc* ou appris sur un corpus [Thomson et Young 2010]. Même si le paradigme de représentation sémantique est identique à celui du système, on peut s'en écarter suffisamment pour que les tests soient une véritable validation et qu'il y ait effectivement quelque chose à apprendre lors de l'apprentissage.

On peut enfin se placer dans l'univers du POMDP, c'est-à-dire qu'on simule directement les états et les observations du POMDP. C'est le niveau le moins fin, qui est le plus éloigné de la réalité, mais c'est aussi celui où l'on maîtrise le plus les différents paramètres. La validité des évaluations réalisées à ce niveau permet plutôt une vérification de l'auto-cohérence du système que sa validation effective. Ce niveau de POMDP résumé a permis de réaliser l'étude préliminaire détaillée dans la partie 3.2.

1.4.2 Simulation de l'utilisateur

Plusieurs modèles d'utilisateurs simulés ont été proposés dans la littérature, utilisant des techniques variées :

- Modèles de n -gramme et de Levin [Levin et al. 1998; Eckert et al. 1997].
- Modèle de graphe [Scheffler et Young 2000; Scheffler et Young 2001].
- Modèle d'agenda : reposant sur les travaux de [Wei et Rudnicky 1999].
- Modèle de réseau bayésien [Pietquin 2004; Pietquin 2005; Pietquin 2006].

On trouvera un état de l'art récent dans [Schatzmann 2008; Pietquin et Hastie 2011]. Le modèle de l'environnement est généralement appris sur un corpus. Par exemple, [Rieser et Lemon 2008] estime un modèle à l'aide d'un corpus de petite taille, les données en faible quantité permettent le *bootstrap* de la stratégie du système.

Nous décrivons ici ces modèles, illustrant la tendance à concevoir des utilisateurs simulés de plus en plus complexes et permettant une modélisation de l'utilisateur de plus en plus précise.

Bigramme (Eckert)

Un des premiers modèles d'utilisateur proposés est celui de [Eckert et al. 1997] qui repose sur une modélisation de bigramme (en réalité un n -gramme simplifié en un bigramme du fait de la faible quantité de données). L'action de l'utilisateur simulé a_u dépend uniquement de l'action précédente du système de façon probabiliste.

$$P(a_u|H) \approx P(a_u|a) \quad (42)$$

avec H l'historique complet du dialogue (actions de l'utilisateur et actions du système) et a la dernière action du système.

Ce modèle peut être appris sur les données d'un corpus. Cette première approche est simple à mettre en œuvre et ne dépend ni de la tâche, ni du domaine. En revanche, la cohérence à long terme sur l'ensemble du dialogue n'est pas assurée par ce modèle de bigramme en raison de l'absence de mémoire interne. L'utilisateur simulé peut tenir des propos en contradiction avec ce qu'il a dit précédemment.

Bigramme contraint (Levin)

Le modèle de bigramme a été ensuite étendu par [Levin et al. 1998; Levin et al. 2000] afin d'y ajouter des contraintes structurelles propres au dialogue. Le modèle de dépendance de l'action utilisateur selon la dernière action du système est contraint en fixant certaines probabilités à zéro. Certaines actions de l'utilisateur sont ainsi interdites et l'action de l'utilisateur est sélectionnée parmi une liste restreinte.

Graphe (Scheffler)

Le principal défaut des modèles de bigramme (contraints ou non) est le manque d'un but utilisateur qui reste relativement stable au cours du dialogue. [Scheffler et Young 2000; Scheffler et Young 2001] ont proposé un modèle reposant sur un graphe *ad-hoc* visant à palier ce problème.

Agenda

Les travaux de [Wei et Rudnicky 1999] proposent un modèle d'agenda pour la gestion du dialogue. Le but visé est d'assurer une possibilité d'initiative mixte et de modéliser des mécanismes d'interaction dépassant la portée d'un seul tour de dialogue. Remarquant que le modèle d'agenda permet de représenter de façon compacte l'état interne d'un utilisateur, [Schatzmann et al. 2007a; Schatzmann 2008] reprennent ce modèle pour la simulation d'utilisateur, de manière statistique.

Certaines des limitations de la simulation d'utilisateur par un modèle d'agenda proviennent de l'utilisation sous-jacente d'un modèle de formulaire. Ainsi il n'est pas possible de prendre en compte de multiples propositions simultanément et la phase de négociation est réduite au strict minimum.

Réseau bayésien (Pietquin)

L'utilisation de réseaux bayésiens pour modéliser l'utilisateur a été avancée par [Pietquin 2004; Pietquin 2006] en unifiant les deux approches précédentes (de Levin et de Scheffler). Afin de maintenir un état interne dans l'utilisateur simulé, l'idée proposée dans [Pietquin 2004] consiste à modéliser l'utilisateur par un réseau bayésien dont la structure et les paramètres ont été fixés par une connaissance *a priori*, apportée par le concepteur.

Le réseau bayésien de la figure 1.19 modélise l'utilisateur au cours du dialogue en incluant un mécanisme d'évaluation de la satisfaction de utilisateur. L'état interne de l'utilisateur est modélisé à travers les nœuds K (*Knowledge*) et G (but ou *Goal*). Les nœuds U , V et $Close$ sont les nœuds de sortie modélisant respectivement l'attribut, la valeur d'un champ et la volonté de terminer le dialogue. Les nœuds I (Expertise), R (Résultats), M

(Métrique) et U_s (satisfaction de l'utilisateur) modélisent le processus d'évaluation du dialogue par l'utilisateur simulé.

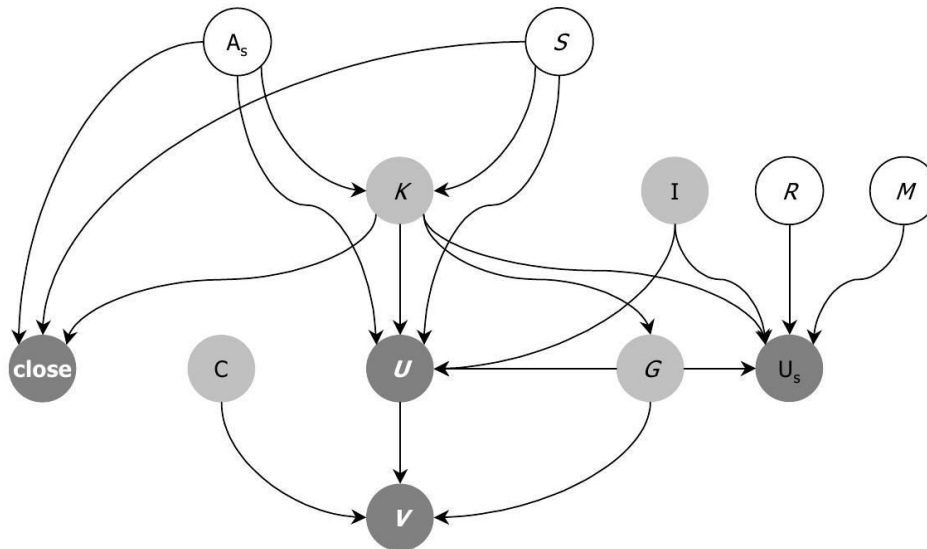


Figure 1.19: Réseau bayésien du modèle de Pietquin de simulation de l'utilisateur (extrait de [Pietquin 2004]).

Notons que le modèle utilise la factorisation de l'état en une liste de champs (formulaire par exemple). Les nœuds S, K, U, V et G du réseau bayésien de la figure 1.19 sont parallélisés autant de fois qu'il y a de champs dans le formulaire. Les dépendances entre les nœuds correspondant à chaque champ sont définies en fonction de l'ontologie de la tâche et du domaine.

Autres modèles

On peut noter aussi l'utilisation de champ de Markov conditionnels (CRF¹) [Jung et al. 2009], ainsi que l'exploration de modèles généralisant les HMMs (Input-HMM, Input-Output-HMM) par [Cuayáhuitl et al. 2005]. Malgré la variété de toutes ces propositions, le modèle de n -gramme proposé initialement n'est pas complètement abandonné : [Georgila et al. 2005] considère son utilisation associée au principe de mise à jour de l'état d'information (ISU) de [Traum et Larsson 2003].

1.4.3 Simulation des erreurs de compréhension

1-best

Le modèle de bruit utilisé est un point important qui influence la robustesse des systèmes, beaucoup de travaux ont été effectués à ce sujet. Les modèles de bruit simples tels que [Watanabe et al. 1998; Hone et Baber 1995] font l'hypothèse d'un taux d'erreur fixe. L'utilisation de méthodes statistiques s'appuie sur des simulations d'erreurs sur les mots

(1) CRF : *Conditional Random Field*.

reposant sur les probabilités de confusions entre phonèmes. Les matrices de confusion entre phonèmes peuvent être définies de façon *ad hoc* par [Pietquin 2004] ou à l'aide de transducteurs (WFST¹) [Stuttle et al. 2004].

n-best

Il y a en revanche peu de travaux réalisés pour modéliser les scores de confiance d'une sortie *n-best*. En ce qui concerne nos travaux, faute de pouvoir consacrer le temps nécessaire à un réel apprentissage sur corpus ou à une innovation sur ce point, nous nous sommes contenté d'un modèle de bruit à taux d'erreur fixe. Mais nous l'avons utilisé pour générer des listes *n-best*.

1.4.4 Simulation du reste de l'environnement

Bruit sur l'action du système

L'erreur pouvant arriver sur l'action du système est très peu souvent prise en compte, bien que [Pietquin 2004] note déjà la nécessité de modéliser aussi cette source d'erreur. En pratique, les systèmes sont conçus de façon à transmettre un message à l'utilisateur de façon non ambiguë et l'articulation des voix synthétiques (TTS) est presque parfaite.

Évaluation

L'évaluation qui définit la fonction objective à maximiser, la fonction de récompense dans le cadre POMDP, peut être définie par une fonction heuristique. Mais elle peut également être conçue d'après un modèle statistique et apprise sur des données. C'est une des utilisations possibles du paradigme PARADISE détaillé dans la partie suivante.

(1) WFST : *Weighed Finite State Transducer*.

1.5 Évaluation

Résumé : Pour juger de la pertinence des différents paradigmes de système de dialogue, l'évaluation des performances d'un système de dialogue peut être réalisée par des tests avec des utilisateurs réels (par un questionnaire) ou par des utilisateurs simulés. Par régression linéaire, il est aussi possible d'estimer les réponses des utilisateurs réels face à un questionnaire (PARADISE).

Afin de comparer les diverses propositions de conceptions de systèmes de dialogue oral homme-machine, il est nécessaire de disposer de méthodes d'évaluation quantitatives. Les critères d'évaluation des performances d'un système sont de deux types :

- Critères objectifs : ces critères sont l'expression d'un juge observant le dialogue, considérant le système comme parfaitement observable. Ils peuvent faire intervenir toutes les données enregistrées lors du dialogue (les fichiers de traçabilité ou *logs*).
- Critères subjectifs : ce sont les évaluations du point de vue de l'utilisateur, par exemple sa satisfaction, son envie de réutiliser le système ou sa perception de réussite ou d'échec de la tâche. Ils font intervenir les intentions réelles de l'utilisateur qui ne sont pas accessibles directement à l'expérimentateur.

La question de l'évaluation d'un système de dialogue requiert un grand nombre de dialogues pour être statistiquement significatif. Et cela est d'autant plus vrai dans le cas où, dans le cadre de l'apprentissage par renforcement, la fonction objective que l'on souhaite maximiser est le critère même de l'évaluation. Cette fonction objective doit alors pouvoir être calculée rapidement et sur de grandes quantités de données. Or, le coût résultant d'une évaluation humaine est très élevé. C'est pourquoi beaucoup de travaux utilisent un utilisateur simulé.

Différents modèles d'utilisateurs simulés ont été proposés dans la littérature. Ils posent cependant le problème de savoir si on évalue la performance d'un système ou bien simplement l'adéquation entre le système et l'utilisateur simulé.

L'évaluation avec des utilisateurs humains reste cependant possible dans des conditions réelles si l'on dispose d'un système déployé (Orange [Damnati et al. 2007], ATT [Phillips 2010]) ou en recrutant des sujets pour évaluer des systèmes en phase de développement. Outre les contraintes de temps qui en découlent, il convient de tenir compte de l'adaptabilité des utilisateurs au système et du cadre nécessairement contraint de l'expérience.

Le paradigme d'évaluation PARADISE a été proposé par [Walker et al. 1997] pour évaluer rapidement une grande quantité de dialogues, selon des critères proches de ceux d'utilisateurs humains. Les évaluations réalisées par des humains sur un corpus donné fournissent des paramètres subjectifs sur lesquels PARADISE réalise une régression linéaire, pour les corrélérer à des critères objectifs [Walker et al. 1997]. Il est ensuite possible d'utiliser un estimateur pour les paramètres subjectifs afin d'évaluer d'autres dialogues [Suendermann et al. 2009].

1.5.1 Simulateurs

Premier outil d'évaluation des performances d'un système de dialogue, les simulations d'utilisateurs existent sous des formes variées. Nous avons déjà présenté, dans la partie 1.4, l'état de l'art sur cette question en soulignant la diversité des propositions avancées dans la littérature. Nous avons pu montrer que ces simulateurs posent le problème de savoir si on évalue les performances d'un système ou bien simplement l'adéquation entre le système et l'utilisateur simulé. Une étude croisée des performances de plusieurs systèmes avec plusieurs utilisateurs simulés est proposée dans [Schatzmann et al. 2005; Schatzmann 2008]. Nous ne nous étendrons pas davantage sur ce point qui n'est pas l'objet principal de nos travaux et qui est inhérent à toute méthode reposant sur des simulations.

1.5.2 Utilisateurs humains

1.5.2.1 Sur un système déployé

On dispose de très peu d'études de systèmes réellement utilisés dans l'industrie, pour de multiples raisons.

Tout d'abord, parce que l'industrie a besoin d'une technique dont le risque est contrôlé. Les modèles sémantiques explorés en recherche ne sont généralement pas simples à mettre en œuvre dans un cadre industriel du fait du besoin en corpus annotés. De plus, soucieux de se préserver de la concurrence, les industriels ne divulguent pas aisément les corpus qu'ils obtiennent.

D'autre part, les modèles de gestion de dialogue à base de renforcement n'offrent pas la certitude d'un comportement correct : le contrôle complet de la conception de l'interface (VUID) [Pieraccini et Huerta 2005] n'est pas assuré. C'est pourquoi les systèmes déployés se limitent généralement à des systèmes de reconnaissance de mots-clés, déclencheurs d'une action déterministe ou au paradigme VoiceXML qui fournit un contrôle plus strict de l'exécution du dialogue.

Enfin, les utilisateurs, en situation réelle, ne prennent pas le temps de remplir un formulaire d'évaluation des performances du système lors du dialogue, mais raccrochent dès qu'ils ont terminé. L'évaluation repose donc sur des mesures objectives obtenues à partir des enregistrements des données de session (fichiers *logs*).

1.5.2.2 En laboratoire

Quelques campagnes d'évaluation ont été organisées : *Spoken Dialogue Challenge* [Black et Eskenazi 2009] ou *Evalita* en italien [Evanini et al. 2008]. Cependant, ces campagnes n'ont pas encore la maturité ni l'ampleur de celles de NIST¹ pour la reconnaissance de la parole par exemple.

Le tableau 1.6 donne un aperçu de différentes évaluations de systèmes de dialogue publiées jusqu'à présent. On constate que les évaluations réalisées en laboratoire sont conduites sur un nombre restreint de sujets, avec des sujets plus ou moins naïfs, dans des conditions souvent différentes et sur des tâches variées. Bien qu'il soit nécessaire d'explorer

(1) NIST : *National Institute of Standards and Technology*.

différentes tâches selon les principes que l'on cherche à tester, cela rend malheureusement difficile les comparaisons entre les différentes approches.

Système évalué	Nombres d'utilisateurs, de tâches et de dialogues.			Domaine	Référence
Campagne Evalita	4	3	40	Recherche d'information de clients, produits et ventes.	[Evanini et al. 2008]
<i>Spoken Dialog Challenge</i> ¹				Horaire de bus.	[Black et Eskenazi 2009]
Évaluation de Itspoke ²	30 juges	3	180	Enseignement (<i>tutoring</i>).	[Ai et Litman 2008]
CSLI DM	40	2	40	Sélection de musique sur lecteur MP3. Choix de restaurant.	[Vargès et al. 2006]
HIS	40	1	160	Informations touristiques.	[Schatzmann et al. 2007a]
Adamach		5	423	Gestion des examens.	[Vargès et al. 2008]
	20	8	160	Stratégie de clarification.	[Vargès et al. 2010]
Itspoke	80	4	80	Enseignement (<i>tutoring</i>).	[Forbes-Riley et Litman 2011]

Tableau 1.6: Publication d'évaluations de systèmes de dialogue.

Les conditions de laboratoire ne sont pas les conditions réelles et on peut se demander jusqu'à quel point ces utilisateurs « réels » se comportent vraiment naturellement ou jouent simplement le rôle qui leur est attribué à travers les consignes des expérimentateurs. Il a d'ailleurs été proposé par [Hoey et al. 2007] de faire jouer le rôle d'utilisateurs par des acteurs.

Outre le fait de contrôler l'expérience de plus près, un des intérêts des tests en laboratoire et que l'on peut accéder à un critère essentiel qui est la satisfaction de l'utilisateur (par exemple en lui demandant de remplir un questionnaire). On utilise généralement un questionnaire permettant d'évaluer chacun des critères définissant un système de qualité : [Walker 2000], SASSI [Hone et Graham 2000] ou pour le projet SIRIDUS [Larsson et al. 2002]. A titre d'exemple, le questionnaire de [Walker et al. 2000] est résumé dans le tableau 1.7.

Ce questionnaire évalue de multiples critères et offre une évaluation des différents composants du systèmes de dialogue (ASR, TTS, compréhension et gestionnaire de dialogue). La qualité de l'implémentation est également estimée à travers le critère de *rapi-*

(1) Le Spoken Dialogue Challenge a été réalisé avec de véritables utilisateurs en sélectionnant d'abord plusieurs groupes d'utilisateurs potentiels selon leur niveau d'expertise, puis en mettant en ligne les systèmes à évaluer pendant une période d'un mois.

(2) Il y a deux jugements par dialogues, c'est une évaluation de la qualité de l'évaluation.

dité du système. Et le critère d'*utilisation future* fournit un indicateur global lié au ROI¹ utilisé dans l'industrie.

Critère évalué	Question
Performance TTS	<i>Was the system easy to understand ?</i>
Performance ASR	<i>Did the system understand what you said?</i>
Facilité de la tâche	<i>Was it easy to find the message/flight/train you wanted?</i>
Rythme d'interaction	<i>Was the pace of interaction with the system appropriate?</i>
Expertise de l'utilisateur	<i>Did you know what you could say at each point?</i>
Vitesse du système	<i>How often was the system sluggish and slow to reply to you?</i>
Comportement prévisible	<i>Did the system work the way you expected it to?</i>
Utilisation future	<i>Do you think you would use the system in the future?</i>

Tableau 1.7: Questionnaire de satisfaction utilisateur utilisé dans les tests de [Walker et al. 2000].

1.5.3 Évaluation PARADISE

Les évaluations par des utilisateurs humains sont longues et coûteuses à mettre en œuvre. Afin de l'automatiser partiellement, le paradigme d'évaluation PARADISE [Walker et al. 1997] consiste à utiliser les évaluations réalisées par des utilisateurs humains sur un ensemble réduit de dialogues pour les étendre de façon automatique à un ensemble plus grand. Pour cela, il suffit d'effectuer une régression linéaire sur des évaluations réalisées par des utilisateurs (critères subjectifs), pour les corrélérer à des critères objectifs.

La formule (Erreur : source de la référence non trouvée) exprime le modèle de régression linéaire des critères d'évaluation subjectifs x_i à partir des critères objectifs y_i . Le modèle linéaire appris sur un corpus est ensuite appliqué à d'autres dialogues afin de les évaluer automatiquement.

La régression linéaire de PARADISE fournit une mesure d'évaluation automatique qui peut être utilisée directement en tant que critère à optimiser dans un processus d'apprentissage par renforcement [Walker et al. 1998; Walker 2000].

Bien que le paradigme PARADISE soit élégant et présente l'avantage d'être relativement aisé à mettre en place, il reste que l'évaluation ainsi réalisée pose deux problèmes :

- Beaucoup de données sont nécessaires pour obtenir un modèle de régression linéaire correctement appris.
- Le modèle linéaire n'est pas assez précis ni réaliste [Ai et Litman 2008].

(1) ROI : *Return On Investissement*.

On constate que PARADISE réalise en fait une modélisation minimale de l'utilisateur lors du processus de notation (remplissage du questionnaire). [Pietquin 2006] propose de le modéliser par un réseau bayésien interne à un utilisateur simulé. Affinant cette idée, des travaux récents cherchent à obtenir une modélisation plus fine de l'utilisateur [Möller et Engelbrecht 2008; Möller et Ward 2008]. Il s'agit d'utiliser un modèle plus proche de la réalité, afin de modéliser un comportement complet d'utilisateur lors du processus d'évaluation. Le modèle proposé comporte trois modules visant à modéliser différentes facettes de l'utilisateur : un module de comportement, un module de perception et de jugement et un module de valeur.

Chapitre 2 : Contributions

Sommaire

2.1 Module de compréhension et état de dialogue non contraints.....	78
2.1.1 Modèle sémantique : graphes mémoire de frames.....	79
2.1.2 Agrégation de graphes de frames sémantiques.....	80
2.1.3 Accès non contraint à la base de données.....	81
2.2 Résumé d'action système.....	83
2.2.1 Création d'actions maitres.....	83
2.2.2 Sélection de l'action maitre.....	86
2.2.3 Ordre de la génération et de la sélection.....	87
2.2.4 Complétude de la conception (VUID-completeness).....	87
2.3 Paradigme de POMDP résumé : Réduction de l'espace d'état et de croyance...88	88
2.3.1 Motivation - Principe.....	89
2.3.2 Espaces maitres.....	90
2.3.3 Espaces résumés.....	92
2.3.4 Prise de décision.....	93
2.3.5 Exemple.....	93
2.4 Conception de fonctions résumées.....	97
2.4.1 Résumés ad-hoc.....	97
2.4.2 Résumés automatiques par clustering.....	99
2.5 Optimisation de stratégie POMDP.....	108
2.5.1 Apprentissage par renforcement sur modèle complet.....	108
2.5.2 Simulations d'utilisateurs avec modèle de frames sémantiques.....	110
2.5.3 Principe de simulations itératives.....	112

L'état de l'art du chapitre précédent conduit à constater qu'un verrou technologique important handicape les systèmes de dialogue homme-machine : les systèmes actuels contraignent trop l'utilisateur à se comporter selon le modèle fixé par le concepteur. Un des objectifs de cette thèse est de dépasser ce verrou en utilisant un paradigme de représentation sémantique plus flexible.

Dans cette optique, tous les composants du système doivent permettre la flexibilité d'expression de l'utilisateur. Ainsi, le module de compréhension doit pouvoir modéliser des tours de parole complexes (c'est-à-dire plus complexes qu'un formulaire prédéfini). Le but de l'utilisateur et l'état de dialogue doivent pouvoir être représentés dans leur structure complexe avec un minimum de perte d'information. De même, l'accès à une base de données est généralement fortement contraint par un langage de type SQL. Dans la mesure du possible, cet accès doit être également souple, générique et le moins contraint possible, respectant ainsi un principe de flexibilité.

Cette thèse propose également un paradigme permettant d'utiliser les méthodes de renforcement fondées sur un modèle POMDP. Les systèmes de dialogue ont en effet besoin des qualités de robustesse, d'adaptabilité et d'auto-apprentissage offertes par les méthodes statistiques et le modèle à état caché POMDP est approprié à la nature dynamique du

dialogue. Cependant, l'utilisation de représentations sémantiques flexibles entraîne une complexité accrue du modèle sémantique, tandis que les systèmes POMDP actuels utilisent au contraire un modèle sémantique particulièrement simple pour limiter l'explosion combinatoire (modèle pré-défini de formulaire sous forme d'une liste de couples champ-valeur).

Pour tenter de dépasser le problème de la complexité du modèle, nous proposons une méthode de POMDP résumé différente de celles utilisées dans la littérature. En particulier, nous utilisons, lors de l'étape d'optimisation de stratégie, un algorithme d'optimisation sur un véritable modèle de POMDP. Sur ce point, notre proposition diffère en particulier de [Young et al. 2007] qui réalise une optimisation sur un MDP résumé dont l'état est un résumé de la croyance.

Le point de départ de nos travaux est l'utilisation d'un modèle sémantique flexible, il est détaillé en partie 2.1 (et le début de la partie 2.2 pour ce qui concerne l'accès à la base de données).

Afin de réduire la complexité du problème, nous appliquons le principe de résumé d'action. Il est intégré en utilisant un couplage des scores des actions résumées et des actions maîtres permettant un meilleur contrôle de la stratégie finale. Ceci est l'objet de la partie 2.2.

Pour contrôler l'explosion combinatoire propre au modèle POMDP, nous présentons le paradigme de « POMDP résumé à double suivi de croyance », en partie 2.3.

Ce dernier repose sur une conception appropriée de fonctions résumées. En partie 2.4, on décrit deux méthodes de conception de ces fonctions : manuelle par règles expertes et automatique par *clustering*.

Enfin, en partie 2.5, on décrit l'algorithme d'optimisation de la stratégie du POMDP. S'agissant d'un algorithme sur modèle complet, la question se pose de l'adéquation entre la stratégie optimisée et la stratégie utilisée lors de la génération du corpus qui sert à estimer le modèle. Pour y répondre, on propose d'effectuer l'optimisation du POMDP sur modèle complet, une méthode itérative permettant d'assurer que le modèle de probabilité est obtenu avec un corpus correspondant à la stratégie optimale.

2.1 Module de compréhension et état de dialogue non contraints

Résumé : Pour représenter des tours de parole de l'utilisateur plus complexes, nous proposons d'utiliser, dans le module de compréhension, un modèle sémantique plus flexible que celui d'un formulaire prédéfini. Dans le gestionnaire de dialogue, cette représentation en graphe de frames sémantiques permet d'inclure également des informations pragmatiques sur le contexte : historique du dialogue et résultats de requêtes à une base de données.

Afin de dépasser les limites de la modélisation par un formulaire préétabli et toute en bénéficiant des avantages des méthodes de renforcement, nous avons choisi de concentrer nos efforts sur l'utilisation conjointes de modèle POMDP et d'une représentation sémantique riche. Nous pensons en effet qu'un modèle sémantique riche est nécessaire afin de :

- permettre une plus grande flexibilité dans l'expression de l'utilisateur
- permettre au module de compréhension d'assumer une partie auparavant attribuée au gestionnaire de dialogue dans le processus de dialogue.

Pour ce faire, nous utilisons les acquis des projets MEDIA et LUNA (présentés en 1.1.1) qui ont proposé l'utilisation d'un modèle sémantique répondant à ce besoin de représentation riche, ainsi que des méthodes d'annotation automatiques statistiques (et les outils correspondants) permettant de propager des hypothèses *n-best* lors du traitement par le module de compréhension. La figure 2.1 représente la décomposition d'un système de dialogue qui intègre une représentation sémantique reposant sur une ontologie de *frames*.

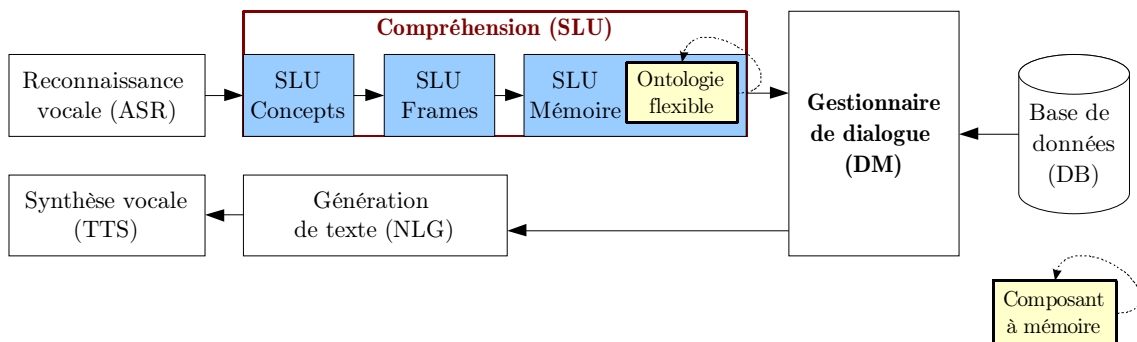


Figure 2.1: Une représentation sémantique à ontologie flexible.

Pour ne pas contraindre l'utilisateur à s'adapter à la logique de conception du système, l'accès à la base de données doit également être le moins contraint possible. Ce qui implique un mécanisme approprié de création de requêtes.

2.1.1 Modèle sémantique : graphes mémoire de *frames*

Comme illustré en partie 1.1.1, les tours de parole (de l'utilisateur comme du système) sont d'abord annotés en concepts (une première annotation dite « hors contexte » qui modélise des dépendances très localisées de type n -gramme, puis une deuxième annotation qui peut se référer à l'ensemble du tour de dialogue complet). Puis, pour chaque tour de parole, une annotation par un graphe de *frames* sémantiques se superpose aux annotations en concepts. Chaque *frame* est constituée d'une liste de rôles sémantiques (un rôle est un élément de la *frame* sémantique). Un rôle peut-être vide, affecté à un concept ou à une *frame*. On trouvera un exemple d'annotation sémantique dans le tableau 1.4.

On différencie deux types de *frames* dans l'ontologie :

- Les *frames* du domaine telles que *Hotel*, *Lodging*, *Facility*.
Dans le cas d'un changement du domaine d'application du système de dialogue, ces *frames*, propres au domaine, doivent être adaptées. Cependant, la difficulté d'adaptation est minimisée par l'utilisation du paradigme FrameNet. De plus, les outils d'annotation statistiques sont génériques et peuvent être employés directement si l'on dispose d'un corpus annoté. La collecte de tels corpus n'est cependant pas une chose aisée, comme dans toute application de méthodes statistiques, la qualité et la quantité de données est un point crucial.
- Les *frames* d'actes de dialogue telles que *Request*, *Want*, *Politeness*.
Ces *frames* ont été conçues de façon générique et ne dépendent pas du domaine ni de la tâche. Elles peuvent être réutilisées sans modification majeure.

Quelles différences par rapport à un modèle de formulaire ?

Une fois simplifié, le modèle de graphe de *frames* apparaît proche d'un modèle de formulaire. Cependant, les champs sont structurés en un graphe de sous-formulaires et la structure est élaborée au cours du dialogue par les deux interlocuteurs. Les avantages de cette représentation sémantique sont les suivants :

Représentation générique des actes de dialogue et du domaine.

La représentation unifiée des *frames* d'actes de dialogue et des *frames* du domaine permet de représenter des tours de paroles comportant plusieurs actes de parole imbriqués tels que le suivant donc le verbe *vouloir* constitue un acte de dialogue portant sur l'acte d'effectuer la réservation et sur l'acte de répéter une information :

« *Je ne veux pas réserver, mais seulement que vous répétiez le nom de l'hôtel* ».

Représentation simple des alternatives.

Une alternative entre deux fragment de représentation sémantique se représente naturellement par deux branches différentes, présentes simultanément lors du dialogue.

« *Je vous propose l'hôtel Ibis des Calanques, l'hôtel Formule Un du Beau Séjour ou le Grand Ritz.*

- *Pouvez-vous me donner le prix de l'hôtel Ibis et du Formule Un ?* ».

- « Veuillez donner les dates de votre réservation.
– D’abord du 7 au 9 octobre, puis du 14 au 16. »

Représentation multiple d’un même concept

Un même concept peut être représenté plusieurs fois avec différentes valeurs entrant dans une relation disjonctive (OU logique) ou une relation conjonctive (ET logique).

- « Dans quelle ville souhaitez-vous voyager ?
– À New York, ou bien Tokyo. »

Notons que dans le modèle sémantique de *frames* que nous avons utilisé, la distinction entre les relations disjonctive et conjonctive n’est pas modélisée. Ce point est à mettre en relation avec [Fillmore 1985] qui fait l’hypothèse que la U-sémantique (compréhension) prime sur la T-sémantique. Ainsi, le calcul des valeurs de vérité de propositions logiques est secondaire devant la modélisation des relations sémantiques des concepts contenus dans l’énoncé.

Selon la tâche considérée, il peut cependant être essentiel de modéliser de telles relations logiques. Une telle modélisation reste évidemment possible en définissant des *frames* de disjonction ou de conjonction appropriées.

Souplesse de modélisation

La structure du graphe n’est pas définie à l’avance. Seuls les éléments pouvant apparaître dans le graphe sont définis dans l’ontologie de *frames*. Ainsi l’exemple suivant est celui d’une requête de l’utilisateur qui n’a pas été envisagée lors de la définition du modèle sémantique mais qui peut parfaitement être représentée dans ce modèle :

- « Je veux que vous me demandiez de confirmer ma réservation »

La définition des objets sémantiques élémentaires du modèle est guidée par les données d’apprentissage. Ces dernières sont fournies par des experts linguistes qui ont annoté le corpus en *frames*. Elles reposent donc sur un fondement théorique linguistique mais ne requièrent pas *a priori* d’expertise dans le domaine de l’intelligence artificielle ou des systèmes de dialogue. Le processus de modélisation sémantique est ainsi partiellement automatisé.

2.1.2 Agrégation de graphes de *frames* sémantiques

L’état du dialogue est modélisé par un graphe de *frames*. Ce graphe est élaboré progressivement grâce aux *frames* des tours de parole de l’utilisateur (correspondant à l’action exacte de l’utilisateur a_u ou à un élément de la liste *n-best* d’observation o) et aux *frames* du système (correspondant à l’action du système a).

Le modèle sémantique des projets MEDIA/LUNA offre une représentation sémantique riche. Cependant, ces projets n’intègrent pas d’information pragmatique concernant le dialogue en entier mais s’arrêtent à la représentation sémantique dans le contexte d’un tour de parole. L’agrégation des informations contenue dans les différents tours de dialogue en une seule structure est évidemment essentielle dans le cadre d’un système de dialogue homme-machine. Cette réunion de graphes de *frames* de chaque tour pour former un seul graphe peut être réalisée en utilisant les mêmes méthodes statistiques (DBN, CRF) que celles présentées

dans les projets MEDIA/LUNA. Nous avons choisi de développer des règles génériques pour réaliser cette étape car on ne dispose pas pour l'instant de corpus annotés par des représentations sémantiques se référant au contexte global du dialogue. Cette méthode peut être rapprochée de celles de [Meurs 2009] par réseaux bayésiens dynamiques ou de [Duvert et al. 2008] avec une approche par règles logiques.

Nous décrivons ici les principes guidant la façon dont les nouvelles informations observées sont intégrées dans la mémoire du système. L'information nouvelle, représentée sous la forme d'un graphe de *frames* sémantiques, est décomposée en branches. Puis, l'intégration est effectuée pour chaque branche en partant de l'information la plus haute sémantiquement (une des *frames* sans prédécesseur) de manière récursive selon l'algorithme 2.1.

Algorithme 2.1 : Intégration d'un nouveau graphe sémantique dans un graphe mémoire

Entrées : G_1 graphe mémoire initial.

G_2 nouvel arbre à intégrer dans G_1 .

Sortie : G_1 graphe modifié.

début:

trouver l'information la plus haute sémantiquement dans G_2 : notée f_2

trouver, par une heuristique, les endroits d'intégration de f_2 dans G_1 :

si il y en a un

 sélectionner le meilleur endroit dans G_1 pour intégrer f_2 : noté f_1

 intégrer f_2 dans f_1 en résolvant les conflits entre des valeurs différentes (ancrage).

 intégrer les enfants de f_2 (**appel récursif**)

sinon

 utiliser une ontologie pour créer un point de jointure f_2' dans G_2 au dessus de f_2

 intégrer f_2' dans G_1

 recommencer au *début*.

2.1.3 Accès non contraint à la base de données

Afin de profiter de la richesse de représentation de l'état de dialogue lors de l'accès à la base de données, il est nécessaire de prolonger le paradigme de représentation flexible motivant l'emploi de structures de graphes de *frames*. Nous décrivons ici un mécanisme générique de génération de requêtes à la base de données, et d'intégration des informations obtenues à l'état de dialogue à partir duquel l'action est choisie.

Génération des requêtes

Étant donné un état de dialogue modélisé sous la forme d'un graphe de *frames*, toutes les requêtes possibles sont générées en faisant correspondre tous les éléments de *frames* sémantiquement compatibles avec un champ du formulaire de requête.

Par exemple, à partir d'une information sur l'état de dialogue présentée selon la figure 2.2, il est possible de générer les requêtes du tableau 2.2. Les requêtes incohérentes, absurdes ou non autorisées sont supprimées, ce qui permet d'assurer une meilleure qualité

de la conception système (VUID). Les requêtes autorisées sont ensuite soumises à la base de données qui fournit une liste de résultats (tableau 2.4, p 85) qui pourront être utilisés lors de la génération d’actions maîtres.

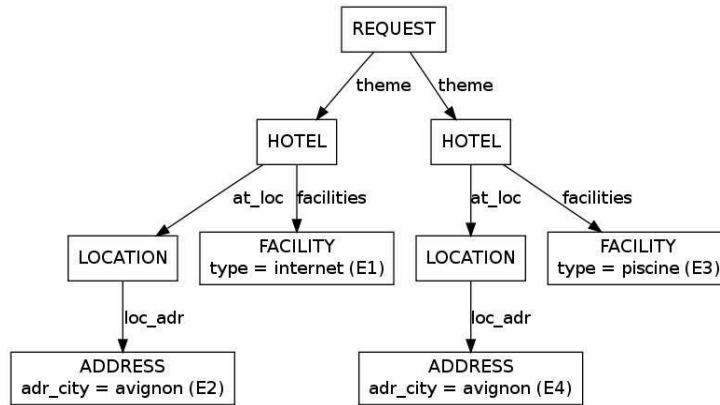


Figure 2.2 : État de dialogue permettant de générer plusieurs requêtes.

Intégration des résultats de la base de données

Toutes les requêtes sont ensuite envoyées à la base de données et les réponses sont ajoutées au graphe mémoire de *frames*. Chaque résultat de la base de données reste associé aux éléments de *frames*, morceaux d’information, qui ont généré la requête.

	Ville	Équipement / Gamme de prix / Marque	Validation de la requête	Nombres de résultats
req1	Avignon	avec piscine et internet	Autorisée	0
req2	Avignon	avec piscine	Autorisée	1
req3	Avignon	internet	Autorisée	1
req5		avec internet	Non autorisée car le nom de la ville est requis (par décision du concepteur).	n/a
req6		avec piscine		
req7		avec piscine et internet		

Tableau 2.2 : Listes des requêtes à la base de données (générées à partir de l’état courant de la figure 2.2).

Utilisation des résultats de la base de données pour prendre une décision

On définit un état de dialogue enrichi par les informations de l’historique de dialogue ainsi que par des informations pragmatiques issues de la base de données. Dans un système non bruité, la décision de l’action à prendre serait réalisée sur cet état enrichi. Dans un POMDP complet, si l’explosion combinatoire et le manque de données n’empêchent pas la résolution, elle utiliserait un vecteur de croyance sur cet état. Dans un POMDP résumé, la variable permettant le choix est un résumé de ce vecteur de croyance.

2.2 Résumé d'action système

Résumé: La projection dans un espace d'action résumé permet de réduire la taille du modèle. Un mécanisme de retour dans l'espace maître est ensuite nécessaire. Il peut être implémenté par un couplage avec un système non résumé.

Afin de contrôler la complexité du POMDP utilisé dans le système de dialogue, et de réduire la taille des stratégies à optimiser, on propose d'utiliser le mécanisme d'action résumée décrit dans l'état de l'art (partie 1.2.6). À partir d'un système résumé fournissant un score sur des actions résumées, il est nécessaire de revenir à l'espace des actions maîtres pour prendre une décision réelle. Ensuite, il faut choisir entre des actions maîtres ayant la même action résumée, et, par conséquent, le même score POMDP.

Le retour à l'espace des actions maîtres est réalisé en générant la liste de toutes les actions possibles dans l'état de croyance courant (en 2.2.1).

Dans un second temps, on sélectionne l'action du système en utilisant simultanément deux mécanismes de décision (en 2.2.2) :

- Un système de règles non stochastiques définies de façon experte en fonction du domaine et de la tâche.
- Un système POMDP, entraîné par renforcement, fournissant les scores des actions résumées à travers la fonction $Q(\tilde{b}, a)$.

Il est techniquement équivalent de considérer, comme dans [Gasic et al. 2009], que le choix de l'action s'effectue dans l'ordre inverse (sélection de l'action par le POMDP, retour à une action maître, et repli sur une autre en cas d'impossibilité). Ceci est détaillé dans la partie 2.2.3.

2.2.1 Création d'actions maîtres

Afin de minimiser la quantité de données nécessaires pour apprendre les modèles de probabilité du POMDP et surtout de réduire le temps d'optimisation à une limite raisonnable, c'est dans l'espace d'actions résumées que la stratégie du POMDP est optimisée. Le modèle statistique fournit ainsi une estimation de la valeur de chaque action résumée selon la croyance résumée courante : c'est la fonction $Q(\tilde{b}, a)$. Il faut alors réutiliser les informations contenues dans l'état du dialogue (ou une estimation de cet état basée sur la croyance) pour revenir dans l'espace des actions maîtres.

Pour cela, on génère tout d'abord pour chaque action résumée \tilde{a} une liste d'actions maîtres potentielles ou possibles $L_{\tilde{a}}$. Et on concatène ces listes en une seule liste L .

$$L = \bigcup_{\tilde{a}} L_{\tilde{a}} \quad (43)$$

L'action du système est ensuite sélectionnée parmi les actions possibles L . Nous prendrons comme exemple les actions résumées `Offer` (proposer un résultat issu de la base de données) ou `ConstQ` (poser une question contrainte).

La génération d'une liste d'actions maîtres $L_{\tilde{a}}$ à partir d'une action résumée \tilde{a} est effectuée en deux étapes :

- i - Requêtes à la base de données.
- ii - Génération d'actions en utilisant les résultats de la base de données ainsi que les informations sémantiques associées à la requête.
- iii - Génération de texte (NLG) par modèles (*templates*)

En pratique, dans notre système, puisque l'essentiel de nos travaux portaient sur le gestionnaire de dialogue et la représentation sémantique, l'étape (iii) de génération de texte à partir de la représentation symbolique d'une action a été implémentée de façon particulièrement simple par l'utilisation de modèles (*templates*) de tours de parole. Par conséquent, et pour simplifier l'exposé, nous présenterons ensemble les deux étapes (ii) et (iii) de génération d'actions et NLG.

i Requêtes à la base de données

Les résultats de ces requêtes sont notés res_1, \dots, res_k . On ne retient parmi ces résultats que ceux qui sont les plus appropriés pour l'action \tilde{a} . Par exemple, pour l'action `Offer`, ou `Confirm`, seules les requêtes ayant fourni des résultats d'hôtels disponibles sont pertinentes. De même, pour l'action `Inform`, toutes les requêtes ayant fourni des résultats d'hôtels disponibles sont pertinentes. Pour les actions `ConstQ`, `Open` ou `Close` en revanche, aucune requête n'est utilisée. De plus chaque résultat est associé aux éléments d'information de l'état qui sont à l'origine de la requête.

	Nom de l'hôtel	Ville	Équipement	Éléments d'information sémantique associés
hôt1	Hôtel du Chat Noir	Avignon	piscine, jacuzzi	Éléments E3 et E4.
hôt2	Hôtel du Pont	Avignon	internet, salle de sport	Éléments E1 et E2.

Tableau 2.3: Résultats de la bases de données, obtenus à partir des requêtes autorisées du tableau 2.2.

ii Génération d'action et NLG par modèles (*templates*)

A chaque action résumée \tilde{a} sont associés des modèles d'actions maîtres. Ils sont instanciés en utilisant les résultats et les informations associées provenant de l'état.

Exemple 1 : Action résumée `Offer`

Dans cet exemple, toutes les requêtes à la base de données sont pertinentes pour l'action résumée `Offer`, les résultats de ces requêtes sont donnés dans le tableau 2.3.

Les modèles de génération de texte associés à Offer peuvent être :

Offer 1 : « À \$ville\$, je vous propose l'hôtel \$nom-hotel\$. »

Offer 2 : « À \$ville\$, je vous propose l'hôtel \$nom-hotel[1]\$ ou l'hôtel \$nom-hotel[2]\$. »

Offer 3 : « À \$ville\$, je vous propose l'hôtel \$nom-hotel[1]\$, l'hôtel \$nom-hotel[2]\$, ou l'hôtel \$nom-hotel[3]\$. »

Les modèles comportent des champs qui doivent être instanciés en utilisant les éléments d'information associés au résultat de la requête. Ces champs à instancier pour transformer le modèle en véritable action maître sont noté à l'aide du signe dollar (\$). L'instanciation de ces champs est requise, c'est-à-dire qu'il est nécessaire de disposer des informations correspondantes pour générer l'action maître. Par exemple, le modèle d'action « Offer 3 » requiert l'utilisation de trois noms d'hôtels issus de la base de données, elle ne donne lieu à aucune action maître car on ne dispose que de deux résultats.

Les actions maîtres générées pour l'action résumée Offre (proposition d'une solution) sont les suivantes :

A1	« À Avignon, je vous propose l'hôtel du Chat Noir »
A2	« À Avignon, je vous propose l'hôtel Du Pont »
A3	« À Avignon, je vous propose l'hôtel Du Pont ou l'hôtel du Chat Noir »

Tableau 2.4: Actions maîtres correspondant à l'action résumée Offre.

Exemple 2 : Action résumée ConstQ

Les modèles de génération de texte associés à l'action résumée ConstQ (poser une question contrainte) peuvent être :

ConstQ 1 : « Dans quelle ville voulez-vous réserver ? »

ConstQ 2 : « À quelles dates voulez-vous réserver ? »

ConstQ 3 : « De quels équipement de confort voulez-vous bénéficier ? »

On constate qu'aucun résultat de requêtes à la base de données n'est pertinent pour l'action résumée ConstQ puisque les modèles ne comportent pas d'élément à instancier à partir d'information sémantique de l'état de dialogue.

A4	« Dans quelle ville voulez-vous réserver ? »
A5	« À quelles dates voulez-vous réserver ? »
A5	« De quels équipement de confort voulez-vous bénéficier ? »

Tableau 2.5: Actions maîtres correspondant à l'action résumée ConstQ.

Le choix des modèles d'actions maîtres est un point très important dans la conception de systèmes de dialogue. Cependant, le sujet de cette thèse portant sur les modules de compré-

hension (SLU) et de gestion du dialogue (DM) situés en amont de la chaîne de traitement, nous n'avons pas cherché à tester de nouvelles hypothèses sur ce point, ni à réaliser un module de génération de texte (NLG) optimal. Des méthodes de renforcement ont été également appliquées pour optimiser le module NLG conjointement à la stratégie de dialogue [Lemon 2011].

2.2.2 Sélection de l'action maître

Ambiguïté de l'action résumée

En raison de l'utilisation d'un mécanisme de résumé d'action (décrit en 1.2.6), la meilleure action sélectionnée par le système POMDP est donc une action résumée. Le processus de génération d'actions maîtres détaillé ci-dessus permet d'obtenir plusieurs actions maîtres correspondant à la meilleure action résumée. Il est donc nécessaire de choisir la meilleure action maître parmi celles qui ont été générées.

On propose pour cela d'utiliser un couplage entre le POMDP résumé et un système expert *ad hoc* particulièrement simple qui affecte des scores différents à chaque action maître.

Fonction Q

L'algorithme de résolution du POMDP fournit une estimation de la fonction V , valeur d'un état de croyance b . Elle donne aussi une estimation de la fonction Q , valeur d'une action a dans un état de croyance b . Ces estimations sont fournies à travers une liste de vecteurs α associés à des actions. Les fonctions V et Q ont été définies en partie 1.2.

Ainsi chaque action maître a correspond à une action résumée à laquelle le modèle POMDP associe un score :

$$\text{score pomdp}(\tilde{b}, a) = Q^\pi(\tilde{b}, a) \quad (44)$$

Hiérarchie d'actions maîtres

On définit une fonction Q^0 experte visant à assurer un ordre complet sur l'ensemble des actions maîtresses. Q^0 est définie de façon *ad hoc* selon la tâche considérée.

Couplage de systèmes

Afin de coupler les scores fournis par les deux systèmes POMDP résumé et de la hiérarchie d'actions, nous proposons de réaliser une combinaison linéaire de leurs fonctions Q :

$$\text{score couplé}(\tilde{b}, b, a) = \lambda Q^\pi(\tilde{b}, a) + (1 - \lambda)Q^0(b, a) \quad (45)$$

Le coefficient λ peut être choisi de façon à normaliser les scores POMDP, en utilisant les valeurs extrêmes des deux scores sur un corpus simulé :

$$\lambda = \frac{\max(Q^0) - \min(Q^0)}{\max(Q^\pi) - \min(Q^\pi)} \quad (46)$$

Grâce à ce couplage, les actions maîtres correspondant à une même action résumée mais à des modèles différents obtiennent des *scores couplés* différents. Lorsque plusieurs actions

maitres sont générées par le même modèle, l'action est choisie de façon aléatoire parmi les actions de meilleur score.

2.2.3 Ordre de la génération et de la sélection

Le retour à l'espace maitre depuis une action résumée est présenté ici différemment de celui trouvé dans la littérature. Les tableaux 2.6 et 2.7 compare notre proposition avec celle de [Young et al. 2007]. Les deux paradigmes reposent sur une heuristique pour le retour à l'espace maitre. De plus, ils font tous deux intervenir un système de règles logiques au niveau des espaces maitres afin de contrôler le comportement du système POMDP résumé. Notre proposition permet de réaliser l'étape de sélection d'action en une seule fois en prenant en compte simultanément les scores fournis par les deux systèmes POMDP (résumé) et par règles (maitre). Ce couplage étant réalisé de façon quantitative, une optimisation automatique du mécanisme de couplage pourrait conduire à une amélioration des performances du système.

sélection de la meilleure action résumée par le système résumé (POMDP)
généralisation de l'action résumée : retour à l'espace maître (heuristique).
élimination de l'action si elle est impossible.
sélection éventuelle d'une autre action si l'action a été éliminée par un système maitre (règles logiques)

Tableau 2.6: Retour à l'espace maitre proposé dans [Young et al. 2007].

généralisation des actions résumées : retour à l'espace maître (heuristique).
élimination des actions impossibles.
sélection de la meilleure action maitre par un système résumé (POMDP) et par un système maitre (règles logiques)

Tableau 2.7: Couplage de systèmes résumé et par règles proposé dans cette thèse.

2.2.4 Complétude de la conception (VUID-completeness)

Ces mécanismes de création d'actions maitresses selon des règles permettent éventuellement d'ajouter un contrôle sur le POMDP. En effet, certaines actions peuvent être tout simplement interdites dans certains cas. Par exemple, dans une tâche impliquant une transaction bancaire, l'accès à une ressource non autorisée telle que le solde d'un compte bancaire peut être refusé si le champ « code d'accès » correspondant n'est pas correctement rempli. Cela autorise un meilleur contrôle de l'interaction avec le système.

2.3 Paradigme de POMDP résumé : Réduction de l'espace d'état et de croyance

Résumé : Dans la méthode de POMDP résumé exposée, la complexité du modèle est réduite par une projection dans des espaces d'état et d'observation résumés. On effectue un double suivi de la croyance, dans les espaces maîtres et dans les espaces résumés. La conception des fonctions de projections résumées peut être réalisée de façon experte ou par clustering automatique.

Comme évoqué dans le chapitre 1, une des limitations fortes des méthodes de renforcement à base de POMDP est la taille de l'espace de recherche de la stratégie optimale. En effet, l'utilisation de méthodes de renforcement par POMDP au lieu de MDP est caractérisée par l'introduction de la variable de croyance b pour remplacer un état s complètement observé. L'explosion combinatoire qui s'ensuit limite fortement l'adoption de cette technologie.

Afin de s'attaquer à ce problème certaines méthodes de réduction des tailles des espaces utilisés ont été proposées dans la littérature. Une approche prometteuse, proposée à travers le système HIS [Young et al. 2010], consiste à résumer le vecteur de croyance après avoir réalisé sa mise à jour dans l'espace maître. L'optimisation de stratégie est ensuite réalisée en considérant le MDP dont l'état est un résumé du vecteur de croyance maître. Ceci est décrit dans le chapitre 1.

On propose ici de résumer le POMDP maître (impossible à résoudre) en un POMDP résumé. Après avoir réalisé un premier suivi de croyance dans les espaces maîtres, il est possible d'effectuer une véritable optimisation POMDP, en résumant l'état d'abord, puis en réalisant le suivi du vecteur de croyance sur cet état résumé \tilde{s} . L'optimisation est réalisée ensuite sur un véritable POMDP. Cette proposition se rapproche de la méthode de résumé de Williams [Williams et Young 2005].

Un deuxième point important dans l'utilisation de POMDP est l'intégration de n -best d'observations. Ce point demande une justification théorique et peut conduire potentiellement à de grandes améliorations des systèmes. L'approche présentée permet également d'intégrer l'information apportée par les scores de confiance d'une observation constituée d'une liste n -best.

Par cette proposition, nous nous interrogeons sur la meilleure façon d'effectuer les deux étapes essentielles de suivi de la croyance et de la réduction de la quantité d'information utilisée. Faut-il absolument établir un suivi de croyance dans l'espace maître ? A quel niveau est-il intéressant d'intégrer des connaissances issues d'un corpus ?

La partie 2.3.1 présente le principe général de notre approche. La première mise à jour de la croyance dans l'espace maître ainsi que l'intégration de n -best est décrite dans la

partie 2.3.2. La deuxième mise à jour effectuée dans les espaces résumés est détaillée dans la partie 2.3.3. La prise de décision finale en 2.3.4 renvoie à la partie 2.2.2 : « Sélection de l'action maître » (ci-dessus). Enfin, un exemple détaillé est présenté en 2.3.5.

2.3.1 Motivation - Principe

Nous avons vu que les propositions de la littérature ne proposent pas toujours une véritable optimisation POMDP [Young et al. 2010]. D'autres utilisent une projection résumée différente lors de l'apprentissage et de l'exécution [Williams et Young 2005]. Le modèle POMDP résumé que nous présentons ici ne présente pas ces limitations.

En effet, il est possible de résumer le modèle POMDP maître en un véritable POMDP en définissant des états et des observations résumés. Le suivi de croyance peut alors être effectué dans l'espace maître et dans l'espace résumé successivement.

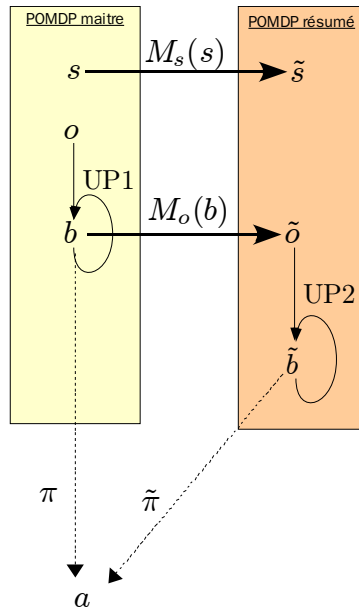


Figure 2.3: POMDP sur résumé de croyance (double suivi de croyance).

Ce POMDP résumé effectue deux résumés. Il s'agit d'abord d'utiliser une fonction résumée pour obtenir l'état résumé à partir de l'état maître : M_s . Ensuite, on calcule l'observation résumée à partir de la croyance maître grâce à une autre fonction résumée : M_o . Enfin, on peut mettre en place un POMDP dont l'état et l'observation sont ces résumés. L'optimisation de ce POMDP résumé n'est pas contrainte et la stratégie optimale y est optimisée sur l'ensemble des stratégies résumées possibles.

2.3.2 Espaces maitres

Décomposition du modèle

On utilise une décomposition du modèle de l'état de dialogue plus simple que celle de $s = (a_u, g, h)$ (présentée en partie 1.3.2 : « Structure d'un POMDP pour le dialogue »). En effet, on ne sépare pas le couple de variables g et h et on le note σ . La décomposition de l'état utilisée est la suivante :

$$s = (a_u, \sigma) \quad (47)$$

La croyance $b(s)$ est définie comme la probabilité conditionnelle de l'état s sachant tout l'historique observé du dialogue $H^{(o)} = (o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o)$.

$$b(s) = P(s | H^{(o)}) = P(s | o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o) \quad (48)$$

Les croyances $b(\sigma)$ et $b(a_u)$ sont définies de même :

$$b(\sigma) = P(\sigma | H^{(o)}) = P(\sigma | o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o) \quad (49)$$

$$b(a_u) = P(a_u | H^{(o)}) = P(a_u | o_1, a_1, o_2, a_2, \dots, o_{t-1}, a_{t-1}, o) \quad (50)$$

Notons que la propriété de somme des probabilités conjointes permet d'écrire :

$$b(\sigma) = \sum_{a_u} b(a_u, \sigma) \quad (51)$$

Mise à jour de la croyance

La formule générale de mise à jour de la croyance dans un POMDP est :

$$b'(s') = k \cdot P(o'|s') \sum_s P(s'|s, a) b(s) \quad (52)$$

En utilisant la décomposition (47) avec une hypothèse raisonnable d'indépendance¹, la formule de mise à jour s'écrit alors :

$$b'(a'_u, \sigma') = k \cdot P(o'|a'_u) \sum_{\sigma} \sum_{a_u} P(a'_u, \sigma' | a_u, \sigma, a) b(a_u, \sigma) \quad (53)$$

Modèle de l'utilisateur

Le modèle de utilisateur $P(a'_u, \sigma' | \sigma, a)$ peut être calculé de façon déterministe, c'est-à-dire que la dépendance entre a'_u et σ' est $P(a'_u, \sigma' | \sigma, a) = comp(a'_u, \sigma', \sigma, a)$. La fonction de compatibilité $comp$ est définie par une loi déterministe :

(1) L'hypothèse d'indépendance est la suivante : $P(o'|a'_u, \sigma) = P(o'|a'_u)$.

$comp(a_u', \sigma', \sigma, a) = 1$	Les variables a_u', σ', σ et a sont compatibles si et seulement si en prenant les actions a et a_u' , la transition (déterministe) s'effectue de σ à σ' .
$comp(a_u', \sigma', \sigma, a) = 0$	Sinon.

Tableau 2.8: Fonction de compatibilité des actions avec les but et historique courants.

La formule classique de mise à jour de la croyance s'écrit alors, en utilisant la fonction de booléenne $comp$ définie dans le tableau 2.8 :

$$b'(a_u', \sigma') = k.P(\sigma'|a_u') \sum_{comp(a_u', \sigma', \sigma, a)} \sum_{a_u} b(a_u, \sigma) \quad (54)$$

Soit, en utilisant (51) :

$$b'(a_u', \sigma') = k.P(\sigma'|a_u') \sum_{comp(a_u', \sigma', \sigma, a)} b(\sigma) \quad (55)$$

2.3.2.1 Intégration de *n-best*

La présentation de la partie 1.3.4.2 : « État d'information caché (HIS) » est reprise ici pour faciliter la lecture.

Scores de confiance fiables

Rappelons que l'observation o est constituée d'une liste de *n-best* associés à leurs scores de probabilités, chaque acte de dialogue o^i est associé à son score de confiance p_i :

$$o = (o^{(i)}, p_i) \quad (56)$$

L'événement « $o^{(i)}$ est la véritable action de l'utilisateur » peut se noter $\{o^{(i)} = a_u\}$.

On peut supposer que le score de confiance p_i du i -ième *n-best* fournit une bonne estimation de la probabilité pour que ce *n-best* soit la véritable action utilisateur a_u sachant o :

$$P(\text{« } o^{(i)} \text{ est la véritable action de l'utilisateur »} | o) = P(o^{(i)} = a_u | o) = p_i \quad (57)$$

Deuxième approximation

En utilisant la formule de Bayes,

$$P(o | o^{(i)} = a_u) = \frac{P(o)}{P(o^{(i)} = a_u)} \cdot P(o^{(i)} = a_u | o) \quad (58)$$

$$= \frac{P(o)}{P(o^{(i)} = a_u)} \cdot p_i \quad (59)$$

$$\approx k^0 p_i \quad (60)$$

Le modèle d'observation est ainsi approximé en utilisant les scores n -best selon la formule suivante :

$$P(o | a_u) \approx k^0 p_i \quad (61)$$

La constante k se simplifiant dans le calcul de la mise à jour de la croyance, elle peut être ignorée.

Intégration des scores de confiance

Soit en sommant sur a'_u pour calculer $b'(\sigma')$:

$$\begin{aligned} b'(\sigma') &= \sum_{a'_u} b'(a'_u, \sigma') \\ &= \sum_{a'_u} k.P(o'|a'_u) \sum_{comp(a'_u, \sigma', \sigma, a)} b(\sigma) \\ &= \sum_{\alpha \in \mathbb{A}} k.P(o'|a'_u = \alpha) \sum_{comp(\alpha, \sigma', \sigma, a)} b(\sigma) \end{aligned}$$

On suppose que la liste n -best de σ' est quasi-exhaustive ou, à tout le moins, que les probabilités $P(\sigma'|a'_u = \alpha)$ sont très faibles pour les actions α qui ne sont pas dans la liste. La somme sur \mathbb{A} peut alors être approximée par la somme sur les actions de la liste :

$$\begin{aligned} b'(\sigma') &\approx k \sum_i P(\sigma'|a'_u = o^{(i)}) \sum_{comp(o^{(i)}, \sigma', \sigma, a)} b(\sigma) \\ &\approx k \sum_i \sum_{comp(o^{(i)}, \sigma', \sigma, a)} p_i b(\sigma) \end{aligned} \quad (62)$$

C'est cette dernière formule (62) que nous utiliserons pour effectuer la mise à jour de la croyance dans l'espace maître. Il s'agit de réaliser le produit cartésien des listes n -best d'observation et de croyance et de sommer les couples (σ', σ) qui conduisent à un même état σ' .

2.3.3 Espaces résumés

2.3.3.1 Projections résumées

Les variables d'état et d'observation résumées sont définies à partir de l'état maître et de la croyance maître selon les formules de projections suivantes :

$$\tilde{s} = M_s(s) \quad (63)$$

$$\tilde{o} = M_o(b) \tag{64}$$

Les fonctions de projection peuvent être définies de façon *ad hoc* par un examen attentif des données. Une méthode plus générique reposant sur un *clustering* automatique est cependant souhaitable. Ces solutions sont détaillées *infra*.

2.3.3.2 Deuxième mise à jour de la croyance

Le vecteur de croyance du POMDP résumé est mis à jour en utilisant les probabilités de transitions et d'observations d'un modèle. La formule de mise à jour de la croyance est la formule (12) répétée ci-dessous pour faciliter la lecture :

$$b'(s') = kP(o'|s') \sum_{s \in \mathbb{S}} P(s'|a, s) b(s)$$

Cela suppose que l'on dispose d'un modèle, par exemple appris sur un corpus. C'est par cet apprentissage sur corpus que le POMDP permet de choisir une action plus appropriée comme illustré dans l'exemple qui suit (en 2.3.5).

2.3.4 Prise de décision

Pour rappel, la décision de l'action à prendre est effectuée par un couplage entre un système déterministe utilisant des informations de l'espace maître et le système POMDP résumé. Le couplage entre ces deux systèmes à travers leurs fonctions Q a été décrit en partie 2.2.2 : Sélection de l'action maître.

2.3.5 Exemple

Espace maître

Rappelons que dans les exemples illustrant l'utilisation d'un modèle de POMDP résumé à double suivi de croyance, nous utiliserons le système simplifié décrit en partie 1.3.2 (*p* 50) comportant deux champs, (**C** et **F**).

L'état (maître) de dialogue initial est $s = [(\mathbf{C} = \text{« Paris »}, \mathbf{F} = \text{fini})]$.

L'observation o est celle définie en (31) :

$o = [(\text{bye} \quad ; 0.6), (\text{inform}(\mathbf{C} = \text{« Avignon »}) ; 0.2)]$.

On supposera que la croyance initiale est

$b = [(\mathbf{C} = \text{« Paris »}, \mathbf{F} = \text{null} ; 1,0)]$

Comme exposé ci-dessous, l'étape de mise à jour est réalisée en utilisant la formule (62) pour intégrer les scores de confiance de l'observation et en faisant le produit croisé avec les éléments de probabilités non-nulle de la croyance.

Après observation, le vecteur de croyance devient :

$$b = [(\mathbf{C} = \text{« Paris »}, \mathbf{F} = \text{Fini} ; 0,6), \\ (\mathbf{C} = \text{« Avignon »}, \mathbf{F} = \text{null} ; 0,4)]$$

Ainsi, après avoir observé une liste *n-best* donnant plus de confiance à l'acte utilisateur `bye`, on se trouve dans un état de croyance qui reflète cet acte : c'est-à-dire que le champ \mathbf{F} prend la valeur « `Fini` ». Pour le système, l'action la plus pertinente d'après cette croyance sera sans doute de clore le dialogue en répondant, par exemple, également par l'acte de dialogue système `bye`.

État résumé

État maître s	État résumé \tilde{s}
État maître permettant de générer une requête à la base de données	<i>CFull</i>
État maître ne permettant pas de générer une requête à la base de données	<i>CEmpty</i>
L'utilisateur a exprimé son souhait de terminer le dialogue	<i>End</i>

Tableau 2.9: Extrait de la définition de la projection d'état résumé M_s .

Comme illustré dans le tableau 2.9, on suppose que la fonction de projection M_s associe les états maîtres permettant de générer une requête à la base de données à l'état résumé *CFull*. Et on suppose également que les états maîtres où l'utilisateur a exprimé son souhait de terminer le dialogue sont projetés sur l'état résumé *End*. On trouvera d'autres exemples d'états résumé définis par une fonction de projection *ad hoc* M_s en partie 2.4 : « Conception de fonctions résumées ».

Dans notre exemple, on fait l'hypothèse que l'état initial du dialogue se traduit dans les espaces résumés par :

$$\text{L'état résumé est : } \tilde{s} = \text{CFull.} \quad (65)$$

Observation résumée

Observation maître o	Observation résumée \tilde{o}
La croyance reflète le fait que l'utilisateur semble avoir exprimé son souhait de terminer le dialogue.	<i>over</i>
La croyance reflète le fait que l'utilisateur souhaite obtenir des renseignements sur une ville qu'il a explicité.	<i>requestX</i>
La croyance reflète le fait que l'utilisateur souhaite obtenir des renseignements mais n'a pas spécifié de nom de ville.	<i>request</i>

Tableau 2.10 : Extrait de la définition de la projection d'observation résumée M_o .

On supposera que la fonction de projection M_o associe les vecteurs de croyance maître où l'utilisateur semble avoir exprimé son souhait de terminer le dialogue à l'observation

résumée *over*. On trouvera d'autres exemples d'observations résumées définies par une fonction de projection *ad hoc* M_o en partie 2.4 : « Conception de fonctions résumées ». Contrairement à cet exemple illustratif qui ne comporte que 3 observations résumées, le nombre de valeurs prises par les fonctions résumées utilisées dans notre prototype sont comprises entre 6 et 19.

Dans notre exemple, comme la croyance du POMDP résumé est, après observation

$$b = [(\mathbf{C} = \text{« Paris »} , \mathbf{F} = \text{Fini} ; 0,6) , \\ (\mathbf{C} = \text{« Avignon »} , \mathbf{F} = \text{null} ; 0,4)]$$

Cette croyance maître reflète le souhait de l'utilisateur de terminer le dialogue. Par conséquent, en utilisant la première ligne du tableau 2.10 :

$$\text{L'observation résumée est : } \tilde{o} = \textit{over}. \quad (66)$$

Croyance résumée

On supposera que la croyance résumée est : $b_{\tilde{s}} = [(\mathit{CFull} ; 1,0)]$, ce qui est cohérent avec l'hypothèse faite sur l'état résumé $\tilde{s} = \mathit{CFull}$.

Modèle d'observation

Le modèle d'observation est appris sur corpus. On supposera que les probabilités d'observations sont celles du tableau 2.11.

$P(\tilde{o} \tilde{s})$	$\tilde{s} = \mathit{CFull}$	$\tilde{s} = \mathit{End}$	$\tilde{s} = \mathit{CEmpty}$
$\tilde{o} = \textit{over}$	0,4	0,9	...
$\tilde{o} = \textit{requestX}$
$\tilde{o} = \textit{request}$

Tableau 2.11: Modèle d'observation : $P(\tilde{o} | \tilde{s})$.

Le POMDP a appris sur le corpus que l'observation *over* correspondait parfois à un état *Cfull* (probabilité de 0,4) et que l'état *End* n'était pas toujours bien observé (seulement 0,9 au lieu de 1,0 pour une observation parfaite).

Modèle de transition

Le modèle de transition est appris sur corpus. On supposera de plus que les probabilités de transition apprises d'après le corpus sont indépendantes de l'action du système \tilde{a} et ont pour valeurs celles du tableau 2.12.

$P(\tilde{s}' \tilde{s}, \tilde{a})$	$\tilde{s} = \mathit{CFull}$	$\tilde{s} = \mathit{End}$	$\tilde{s} = \mathit{CEmpty}$
$\tilde{s}' = \mathit{CFull}$	0,8
$\tilde{s}' = \mathit{End}$	0,2
$\tilde{s}' = \mathit{CEmpty}$

Tableau 2.12: Modèle de transition d'état : $P(\tilde{s}' | \tilde{s}, \tilde{a})$.

Le modèle a appris que la transition de *CFull* à *End* n'est pas très probable. C'est, associé au principe de croyance du modèle POMDP, ce qui lui permet de rester en grande partie dans l'état *CFull*. On le constate en poussant les calculs jusqu'au bout lors de la mise à jour de la croyance.

Mise à jour de la croyance

La formule (12) de mise à jour de la croyance est reprise ci-dessous :

$$\tilde{b}'(\tilde{s}') = kP(\tilde{o}'|\tilde{s}') \sum_{\tilde{s} \in \tilde{\mathcal{S}}} P(\tilde{s}'|\tilde{a}, \tilde{s}) \tilde{b}(\tilde{s})$$

$$\begin{aligned} \tilde{b}'(\tilde{s}' = End) &= kP(\tilde{o}' = over|\tilde{s}' = End).P(\tilde{s}' = End|\tilde{s} = CFull, \tilde{a}).b(\tilde{s} = CFull) \\ &= k \times 0,9 \times 0,2 \times 1,0 = k \times 0,18 \end{aligned}$$

$$\begin{aligned} \tilde{b}'(\tilde{s}' = Cfull) &= kP(\tilde{o}' = over|\tilde{s}' = CFull).P(\tilde{s}' = CFull|\tilde{s} = CFull, \tilde{a}).\tilde{b}(\tilde{s} = CFull) \\ &= k \times 0,4 \times 0,8 \times 1,0 = k \times 0,32 \end{aligned}$$

Après renormalisation (par $k = 2$), la nouvelle croyance \tilde{b}' du POMDP résumé est :

$$\tilde{b}' = [(CFull; 0.64) , (End; 0.36)]$$

Décision

Dans un tel état de croyance, il est vraisemblable que l'optimisation de la stratégie conduira à choisir une action différente de *bye*, puisque le dialogue ne semble pas terminé du point de vue du système.

2.4 Conception de fonctions résumées

Résumé : Le paradigme proposé dans la partie précédente peut être appliqué de deux façons différentes. La première utilise une conception experte. La seconde met en œuvre une méthode d'apprentissage automatique par clustering utilisant une distance entre graphes sémantiques et une mesure de similarité entre listes n -best de graphes.

2.4.1 Résumés *ad-hoc*

Les fonctions de projections dans les espaces résumés M_s et M_o peuvent être élaborées de façon experte en observant les données et en extrayant des graphes de *frames* sémantiques les informations jugées pertinentes à la prise de décision.

L'information complexe que l'on cherche à projeter dans un espace résumé est constituée de l'historique du dialogue. En utilisant l'hypothèse de Markov propre au HMM, l'historique peut être remplacé par la croyance maitre. Cette variable maitre à résumer (l'historique ou la croyance) se présente sous la forme d'une liste de graphes de *frames* sémantiques. Ces graphes sont des structures riches en information sémantique de haut niveau, ils ont été décrits dans le chapitre 1.

On propose d'extraire de la variable maitre des caractéristiques (*features*) primaires, sur lesquelles seront construites des caractéristiques secondaires plus élaborées. Ces mécanismes ont été décrits dans [Pinault et al. 2009] et [Pinault et Lefèvre 2010].

2.4.1.1 Extraction de caractéristiques (*features*) primaires

Caractéristique	Définition	Type
$F_{\text{Query},u}$	Nombre de <i>frames</i> <code>Query</code> apparaissant dans le dernier tour de parole de utilisateur.	Entier
$F_{\text{Query},s}$	Nombre de <i>frames</i> <code>Query</code> apparaissant dans le dernier tour de parole de système .	Entier
$F_{\text{Hotel},u}^{\neq \emptyset}$	Nombre de <i>frames</i> <code>Hotel</code> , non vides , apparaissant dans le dernier tour de parole de utilisateur.	Entier

Tableau 2.13: Exemples de caractéristiques primaires extraites de l'historique de structures de *frames*.

Comme illustré dans le tableau 2.13, les caractéristiques primaires que nous proposons d'utiliser, et celles que nous avons mises en œuvre dans les expériences présentées dans le chapitre 3, sont définies par le comptage de certaines *frames* sémantiques (par exemple la

frame nommée *Query*). On considère le nombres de *frames* sur un tour de parole, ou sur tout l'historique du dialogue, prononcées par l'utilisateur ou par la machine. On établit aussi une différence entre les *frames* vides (qui ne contiennent pas de valeur dans leurs éléments de *frames*) et des *frames* non-vides (qui sont plus informatives). Les caractéristiques primaires sont toutes de type entier.

2.4.1.2 Élaboration de caractéristiques secondaires

Les caractéristiques secondaires sont élaborées à partir des caractéristiques primaires et d'autres caractéristiques secondaires selon différentes méthodes :

- Combinaisons linéaires dont les coefficients ont été définis de façon *ad-hoc* (par exemple F_{QUERY}).
- Comparaison directe avec des valeurs de seuil, cela permet obtenir des caractéristiques booléennes (par exemple F_{BYE})
- Formules logiques (par exemple $F_{\text{USERASKED}}$).
- Utilisation de fonctions simples telles que le calcul du maximum ou du minimum (par exemple dans F_{SUMMARY}).
- Fonction tabulée (par exemple F_{SUMMARY}).

Caractéristique (<i>feature</i>)	Définition	Type
F_{QUERY}	$\theta_1 F_{\text{Query},u} + \theta_2 F_{\text{Query},s} + \theta_3 F_{\text{Closing},u} + \theta_4 F_{\text{Informm},u}$ avec $\theta_1 = +5$, $\theta_2 = +2$, $\theta_3 = -10$ et $\theta_4 = -0,5$	Réel
F_{BYE}	$F_{\text{Closing},u} > 0$	Booléen
$F_{\text{USERASKED}}$	$(F_{\text{Query},u} > 0) \text{ OR } (F_{\text{Request},u} > 0)$	Booléen
F_{SYSASKED}	$(F_{\text{Query},s} > 0) \text{ OR } (F_{\text{Request},s} > 0)$	Booléen
F_{SUMMARY} (\tilde{s} ou \tilde{o})	$F_{\text{QUERY}} > \max(F_{\text{REQUEST}}, F_{\text{INFORM}}) \Rightarrow \tilde{s} = 1$ $F_{\text{SYSASKED}} = \text{Vrai} \Rightarrow \tilde{s} = 2$ $F_{\text{USERASKED}} = \text{Vrai} \Rightarrow \tilde{s} = 3$ Sinon $\Rightarrow \tilde{s} = 4$	Entier

Tableau 2.14: Exemples de caractéristiques secondaires élaborées à partir des caractéristiques primaires.

État résumé : $\tilde{s} = M_s(s)$

La caractéristique de résumé terminale F_{SUMMARY} est la projection résumée permet de calculer la variable d'état résumé \tilde{s} .

Observation résumée : $\tilde{o} = M_o(b)$

La croyance b est une distribution de probabilité sur l'espace d'état, on peut calculer son entropie et s'intéresser à l'état dont la valeur de croyance est maximale. En appliquant le calcul des caractéristiques primaires et secondaires sur cet élément de poids maximal et en y ajoutant la valeur d'entropie discrétisée, la caractéristique de résumé terminale F_{SUMMARY} permet de calculer la projection résumée $\tilde{o} = M_o(b)$.

2.4.2 Résumés automatiques par *clustering*

Afin de limiter au maximum l'intervention humaine lors de la conception d'un système de dialogue, on peut chercher à utiliser des méthodes de *clustering* automatique pour concevoir les fonctions de projection dans les espaces résumés M_s et M_o .

L'utilisation de telles méthodes pour simplifier un POMDP a été étudiée par [Lefèvre et de Mori 2007] dans le cadre d'un modèle sémantique simple (liste de couples champ-valeur). Dans notre cas en revanche, les états que nous allons traiter sont des graphes de *frames* sémantiques. Cette partie décrit les méthodes permettant de réaliser du *clustering* de graphes. On détaille la façon dont est utilisée la distance d'édition entre graphes, pour effectuer du *clustering* de graphes de *frames*.

Les vecteurs de croyance que nous allons traiter pour obtenir les observations résumées sont des listes graphes de *frames* sémantiques. On propose de généraliser la distance entre graphes à la distance entre listes *n-best* de graphes. Et on montre une façon de réaliser un *clustering* de k plus proches voisins en utilisant cette distance.

2.4.2.1 Distance d'édition entre graphes

La distance d'édition entre deux graphes (étiquetés) généralise la notion de distance d'édition de Levenstein. Elle peut être calculée de façon exacte par programmation linéaire. Cette distance permettra de réaliser le *clustering* de graphes sémantiques.

Il a été proposé différentes façons de mesurer la similarité entre deux graphes, leur principale application étant le domaine de la reconnaissance d'image. Outre les métriques fondées sur le plus grand commun sous-graphe (MCS¹) [Messmer et Bunke 1998; Wang et al. 1997; Read et Corneil 1977], une métrique très utilisée est la distance d'édition entre deux graphes. On pourra se référer à un état de l'art sur ce sujet dans [Gao et al. 2010] ou [Sakr et Al-Naymat 2010].

(1) MCS : *Maximal Common Subgraph*.

i Définition

La distance d'édition entre graphes (GED¹) est une généralisation aux graphes de la distance d'édition entre deux chaînes [Navarro 2001] (par exemple la distance de Levenstein). De plus, en réglant correctement les paramètres de coût de la GED (coûts d'une substitution, insertion ou suppression de nœud ou d'arête), résoudre la GED permet de trouver aussi le MCS [Bunke 1997].

Pour résoudre la GED, les algorithmes de résolution dépendent du type de graphes considérés : étiquetés ou non (c'est-à-dire avec ou sans attributs sur les sommets). Nous nous intéresseront aux graphes étiquetés car les graphes de *frames* sémantiques entrent dans cette catégorie.

Afin de fixer les notations, rappelons quelques définitions de théorie des graphes :

Un **graphe** $G = (V, E)$ est défini par ses nœuds V et ses arêtes E .

Un **graphe étiqueté**² $G = (V, E, \alpha, \beta)$ possède de plus des étiquettes (*attributs*) sur ses nœuds et ses arêtes définies respectivement par les fonctions α et β .

Une **édition** est une transformation de graphe telle que l'effacement, l'insertion ou la substitution de nœud ou d'arête. Une édition est **valide** si elle peut être appliquée à un graphe.

Une **suite d'éditions** (ou chemin d'éditions) est une suite indexée par un entier dont les éléments sont des éditions. Un chemin d'éditions est **valide** si ses éléments peuvent être appliqués successivement à un graphe.

Le **problème de la distance d'édition entre graphes** que l'on cherche à résoudre est le suivant :

Soient deux graphes étiquetés

$$G_1 = (V_1, E_1, \alpha_1, \beta_1) \quad (67)$$

$$G_2 = (V_2, E_2, \alpha_2, \beta_2) \quad (68)$$

Soient $c_{nd}, c_{ni}, c_{ns}, c_{ed}, c_{ei}, c_{es}$ les coûts d'édition : respectivement d'effacement, insertion et substitution de nœud et d'arête.

Le coût total d'édition d'un chemin d'édition entre G_1 et G_2 est la somme des coûts d'édition du chemin. Et la distance d'édition entre G_1 et G_2 est le coût d'édition minimal d'un chemin valide entre G_1 et G_2 . Un exemple est proposé *infra* (tableau 2.15 p 103).

ii Utilisation des graphes en TAL

L'utilisation de graphes dans le domaine du traitement automatique de la langue (orale ou écrite) n'est pas nouvelle, un groupe de travail spécifique à cette question contribue activement à l'introduction de méthodes issues de la théorie des graphes dans le domaine du TAL : TextGraph [Mihalcea et Radev 2006].

(1) GED : *Graph Edit Distance*.

(2) Graphe étiqueté : *Attributed graph*.

Le terme de regroupement de graphes (*graph clustering*) a deux acceptions :

- Soit il s’agit de représenter sous la forme d’un graphe les relations entre les points de données à regrouper. Les n points des données sont les sommets d’un unique graphe et il s’agit alors de trouver des groupes (*clusters*) à l’intérieur de ce graphe.
- Soit les données elles-même sont des graphes : on dispose alors de n graphes. Après avoir défini une distance entre graphes, les méthodes de *clustering* classiques peuvent être appliquées (notons tout de même que certaines d’entre elles devront être adaptées, la moyenne entre graphes n’étant pas définie trivialement).

L’utilisation de méthodes de regroupement de graphes en TAL concerne principalement le premier point [Chen et Ji 2010]. Certains travaux ont pourtant proposé d’utiliser la mesure de similarité entre graphes [Molla 2006] pour le problème des questions-réponses, afin trouver des règles logiques automatiquement, les questions et les réponses sont représentées sous la forme de graphes.

iii Résolution

Les familles d’algorithmes de la littérature sont variées. Nous utiliserons dans cette thèse un algorithme de programmation linéaire binaire proposé par [Justice et Hero 2006; Almohamad et Duffuaa 1993]. Il est particulièrement adapté aux graphes de petite taille (< 20 nœuds) et a été utilisé dans [Justice et Hero 2006] pour la classification de molécules de taille réduite (< 20 atomes).

Dans certains cas (pour des valeurs de coûts d’édition uniformes), on peut calculer une distance d’édition à partir de la distance δ , définie comme la différence de taille entre le plus grand sous-graphe commun $G_1 \vee G_2$ et le plus petit super-graphe commun $G_1 \wedge G_2$.

$$\delta(G_1, G_2) = |G_1 \wedge G_2| - |G_1 \vee G_2| \quad (69)$$

L’utilisation de noyaux a également été étudiée [Gärtner et al. 2003] : l’idée des algorithmes de résolution par noyaux est de projeter les graphes dans un espace de grande dimension. Dans ce nouvel espace, les opérations d’éditions peuvent être représentées par des noyaux. Cette méthode ressemble aux machines à vecteur support (SVM) où les données sont également projetées dans un espace plus grand afin d’utiliser des classifieurs linéaires.

D’autres algorithmes plus ou moins complexes reposent sur des réseaux de neurones non-supervisés (SOM¹ [Neuhaus et Bunke 2002]) ou d’autres méthodes probabilistes [Myers et al. 2000].

(1) SOM : *Self Organising Map*.

2.4.2.2 Clustering de graphes de *frames* sémantiques

Les graphes que l'on cherche à regrouper sont obtenus à partir de graphes de *frames* munis d'état d'ancrage, d'étiquettes sur les liens entre *frames* et de valeurs éventuelles sur les éléments d'information (éléments de *frames*). On décrit ici comment les graphes de *frames* sont transformés en graphes de *frames* réduits pour assurer que la distance entre graphes ait les propriétés souhaitées. Puis on détaille comment se calcule de la distance d'édition entre les deux graphes réduits obtenus. Cette distance permet de réaliser le *clustering* de graphes de *frames*.

Graphes de *frames* réduits

Considérons par exemple le calcul de la distance d'édition entre les deux graphes de *frames* sémantiques des figures 2.4 et 2.5. On leur associe les graphes réduits des figures 2.7 et 2.6. Un exemple de chemin d'édition valide est proposé dans le tableau 2.15.

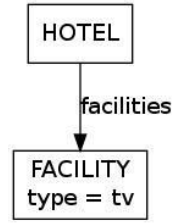


Figure 2.4: Graphe de frames sémantiques 1.

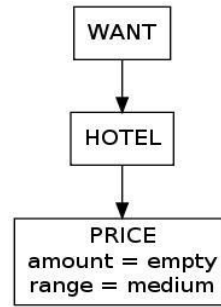


Figure 2.5: Graphe de frames sémantiques 2.

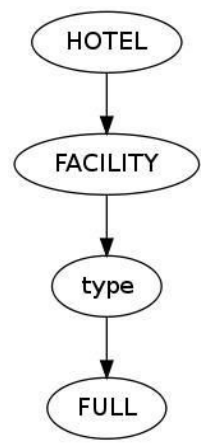


Figure 2.6: Graphe réduit 1.

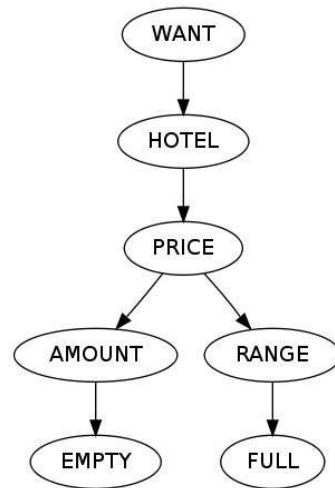


Figure 2.7: Graphe réduit 2.

Édition	Coût
<i>remplacer</i> (FACILITY,PRICE)	c_{ns}
<i>remplacer</i> (range,type)	c_{ns}
<i>insérer</i> (amount)	$c_{ni} + c_{ei}$
<i>insérer</i> (EMPTY)	$c_{ni} + c_{ei}$
<i>insérer</i> (WANT)	$c_{ni} + c_{ei}$

Tableau 2.15: Exemple de chemin d'édition entre les graphes réduits 1 et 2.

Distance entre graphes

La matrice d'adjacence jointe ces deux graphes est la suivante :

		Graphe 1				Graphe 2						
		HOTEL	FACILITY	type	FULL	WANT	HOTEL	PRICE	amount	EMPTY	range	FULL
Graphe 1	HOTEL	1	0	0	0	0	1	0	0	0	0	0
	FACILITY	0	1	0	0	0	0	0	0	0	0	0
	type	0	0	1	0	0	0	0	0	0	0	0
	FULL	0	0	0	1	0	0	0	0	0	0	1
Graphe 2	WANT	0	0	0	0	1	0	0	0	0	0	0
	HOTEL	1	0	0	0	0	1	0	0	0	0	0
	PRICE	0	0	0	0	0	0	1	0	0	0	0
	amount	0	0	0	0	0	0	0	1	0	0	0
	EMPTY	0	0	0	0	0	0	0	0	1	0	0
	range	0	0	0	0	0	0	0	0	0	1	0
	FULL	0	0	0	1	0	0	0	0	0	0	1

Tableau 2.16: Matrice d'adjacence conjointe des deux graphes.

Le calcul de la distance d'édition entre deux graphes peut être effectué par la résolution d'un système de programmation linéaire [Justice et Hero 2006]. Si l'on considère des coûts d'édition uniformes ($c_{nd} = c_{ni} = c_{ns} = c_{ed} = c_{ei} = c_{es}$), le chemin d'édition proposé en exemple dans le tableau 2.15 est optimal.

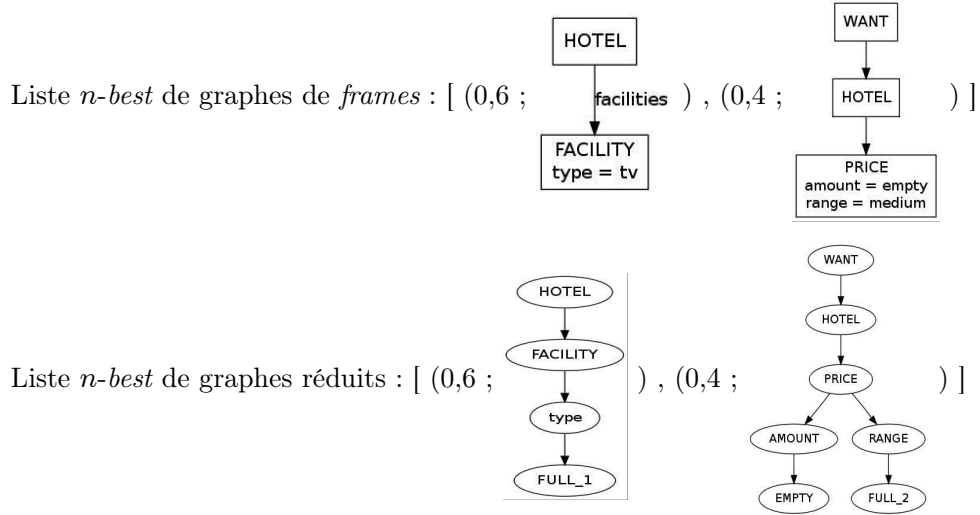
2.4.2.3 Clustering de listes de représentations en graphes de frames

Pour le *clustering* sur des listes n -best de graphes de *frames*, on transforme chaque graphe en graphe réduit, puis on définit une mesure de similarité entre des distributions sur l'espace des graphes.

Pour cela, une première tentative montre que l'utilisation de la distance euclidienne est vouée à l'échec pour ce qui nous préoccupe, voir en iii. On propose ensuite, en iv, une distance inspirée de la distance de Skorokhod qui possède de bonnes propriétés pour notre problème. Enfin, en v, cette distance est simplifiée en une mesure de similarité linéaire afin d'en favoriser l'implémentation.

i Graphes réduits

On réduit chaque graphe de la même façon que pour un graphe simple en différenciant les valeurs des champs selon le rang dans la liste *n-best*. Par exemple, la liste de graphes de *frames* sémantiques ci-dessous se transforme en liste de graphes réduits de la façon suivante :



C'est sur cette liste de graphes réduits que les calculs de distance seront effectués. Nous présentons dans la suite de cette partie, les éléments théoriques permettant de définir une distance entre listes *n-best*.

ii Définitions

Nous introduisons ici les espaces D_G et B sur lesquels vont être définies les distances et les mesures de similarité qui nous intéressent.

Soit D_G l'espace des distributions sur G .

On considère qu'une liste *n-best* est une distribution à support fini : une somme finie de distribution de Dirac. Soit B l'ensemble des listes *n-best* de graphes :

$$B = \left\{ b = \sum_{i=1}^n b_i \cdot \delta_{g_i} / n \in \mathbb{N}, g_i \in G, \sum_{i=1}^n b_i = 1.0 \right\} \quad (70)$$

Remarque : On notera que B est inclus dans l'ensemble des distributions de probabilités, c'est-à-dire que les éléments de B ont pour somme 1.0 :

$$B \subset \{ b \in D_G \text{ tq } \int_G b(g) dg = 1.0 \} \quad (71)$$

iii Distance euclidienne

Notre première tentative pour définir une distance sur D_G a été d'utiliser la distance euclidienne, à l'aide du produit scalaire canonique. Mais cette distance ne convient pas à l'usage que l'on veut en faire car elle n'utilise pas la distance déjà définie sur G .

L'espace des distributions est un espace euclidien, muni d'un produit scalaire :

$$\forall (b, b') \in B^2, \langle b, b' \rangle = \int_G b(g) \cdot b'(g) dg \quad (72)$$

On alors peut utiliser la distance euclidienne induite par le produit scalaire entre les distributions :

$$\forall (b, b') \in B^2, d(b, b') = \|b - b'\|_2 = \int_G (b(g) - b'(g))^2 dg \quad (73)$$

On constate que la distance ainsi définie ne tient pas compte de la similarité des graphes. Elle ne possède donc pas les propriétés souhaitées pour réaliser un *clustering* efficace dans notre cas.

Contre-exemple

Soient deux graphes x et y tels que $d(x, y)$ soit non nul mais très petit. Soit z un graphe très différent de x et de y : les distances $d(x, z)$ et $d(y, z)$ sont très grandes.

Notons b_x, b_y et b_z les listes *n-best* associées à x, y et z :

$$\begin{aligned} b_x &= \Phi(x) = [(x, 1.0)] \\ b_y &= \Phi(y) = [(y, 1.0)] \\ b_z &= \Phi(z) = [(z, 1.0)] \end{aligned}$$

On constate que les distances entre les listes b_x, b_y et b_z ne reflètent pas les distances entre les graphes x, y et z . Au contraire, elles sont toutes égales, alors qu'on souhaiterait que la distance $d(b_x, b_y)$ soit plus petite que les deux distances $d(b_x, b_z)$ et $d(b_y, b_z)$.

Ce contre-exemple montre que la distance euclidienne n'est pas celle que l'on souhaite utiliser pour quantifier la similarité et la dissimilarité entre deux listes *n-best*.

Plus généralement, on désire que la distance entre les distributions généralise la distance entre graphes. La propriété requise correspond à la propriété (74) de conservation de la distance par le morphisme Φ .

$$d(\Phi(g_1), \Phi(g_2)) = d(g_1, g_2) \quad (74)$$

où Φ est l'injection canonique définie par

$$\begin{aligned} \Phi : G &\rightarrow B \\ g &\mapsto \delta_g = [(g, 1.0)] \end{aligned} \quad (75)$$

On constate qu'un problème identique se posera si l'on choisit des distances telles que celles associées aux normes $\|\cdot\|_1$ de Manhattan, $\|\cdot\|_p$ de Minkowski, $\|\cdot\|_\infty$ de Tchebychev ou à la « distance » KL de Kullback-Leibner.

En fait, puisqu'on dispose d'une distance sur l'espace des graphes G , on peut chercher à l'utiliser pour la généraliser à D_G en une distance qui possède des propriétés appropriées.

iv Distance d_1

Les détails de la distance de Skorokhod [Skorokhod 1956; Contreras 2009] sont présentés en annexe C.1. Le point principal dont nous nous inspirons ici est l'idée initiale de définir une distance entre fonctions en utilisant de façon conjointe les distances déjà définies dans les espaces de départ et d'arrivée.

Afin d'utiliser la distance sur les graphes et d'assurer la propriété (74), on propose une formulation probabiliste inspirée par la distance de Skorokhod.

Définition

Soit deux distributions de probabilité b et b' , soient

$$\begin{aligned}\mathbb{X}_b &= \{X \text{ variable aléatoire de loi } b\} \\ \mathbb{X}_{b'} &= \{X' \text{ variable aléatoire de loi } b'\}\end{aligned}$$

On définit la distance $d_1(b, b')$ par un minimum d'espérance de la distance entre deux variables de loi b et b' selon :

$$d_1(b, b') = \inf_{(X, X') \in \mathbb{X}_b \times \mathbb{X}_{b'}} \mathbb{E}(d(X, X')) \quad (76)$$

Notons que les éléments X et X' de \mathbb{X} et $\mathbb{X}_{b'}$ prennent leurs valeurs dans G l'espace des graphes; la distance entre graphes $d(X, X')$ est donc bien définie. d_1 vérifie la propriété (74) de généralisation de la distance d . De plus, d_1 est une distance, on trouvera la démonstration de cette proposition en annexe C.2.

v Simplification (extension par linéarité)

Pour limiter la complexité des calculs, on propose d'utiliser une approximation de la distance d_1 précédemment définie. A cet effet, on définit la mesure de similarité d_2 .

Définition

Soit deux distributions de probabilité b et b' , soient

$$\begin{aligned}\mathbb{X}_b &= \{X \text{ variable aléatoire de loi } b\} \\ \mathbb{X}_{b'} &= \{X' \text{ variable aléatoire de loi } b'\}\end{aligned}$$

On définit la distance $d_2(b, b')$ par un minimum d'espérance de la distance entre deux variables de loi b et b' , en ajoutant une condition d'indépendance entre X et X' , selon :

$$d_2(b, b') = \inf_{\substack{(X, X') \in \mathbb{X}_b \times \mathbb{X}_{b'} \\ X \text{ et } X' \text{ independants}}} \mathbb{E}(d(X, X')) \quad (77)$$

Notons que cette mesure de similarité n'est pas une distance. En particulier, les propriétés ci-dessous **ne sont pas vérifiées** :

$$\forall b, d_2(b, b) = 0 \quad (78)$$

$$d_2(b, b') = 0 \Rightarrow b = b' \quad (79)$$

Forme explicite

En explicitant les deux éléments b et b' de B de la façon suivante :

$$b = \sum_{i=1}^n b_i \cdot \delta_{g_i} \text{ et } b' = \sum_{j=1}^m b'_j \cdot \delta_{g'_j} \quad (80)$$

La distance simplifiée d_2 ainsi définie s'écrit en fonction de la distance entre graphes :

$$d_2(b, b') = \sum_{\substack{1 < i \leq n \\ 1 < j \leq m}} b_i b'_j d(g_i, g'_j) \quad (81)$$

On trouvera la démonstration de cette expression en annexe C.3.

vi Lien avec une définition *ad hoc*

Notons que, dans le système HIS, la mesure de similarité utilisée fait intervenir une comparaison avec le deuxième *n-best* ainsi que la distance KL discrétisée. Comme la distance d_1 , la mesure de similarité d_2 ou la distance de Skorokhod, elle utilise ainsi la distance dans l'espace de départ¹ (comparaison avec le deuxième *n-best*) ainsi que la distance dans l'espace d'arrivée² (à travers le calcul de KL).

2.4.2.4 Clustering

On a appliqué cette méthode pour la conception de fonctions de projections résumées d'état $M_s(s)$ et de croyance $M_o(b)$. Grâce à la distance entre les graphes, on peut appliquer des méthodes de *clustering* pour définir la fonction $M_s(s)$ de résumé d'état. Pour le résumé de croyance $M_o(b)$, on a choisi la mesure de similarité étendue par linéarité d_2 pour effectuer le *clustering*. Les résultats sont détaillés dans le chapitre 3 (3.3.4 : Passage aux espaces résumés).

(1) L'espace de départ est l'espace des états.

(2) L'espace d'arrivée est l'intervalle $[0;1]$.

2.5 Optimisation de stratégie POMDP

Résumé: On montre ici que la résolution d'un POMDP par un algorithme sur modèle complet peut être réalisée avec un apprentissage du modèle sur peu de données. Les simulations d'utilisateurs permettent de générer de nouveaux corpus de données. Afin d'entraîner la stratégie du système sur un corpus de dialogues générés de façon cohérente avec cette stratégie, nous proposons une méthode itérative.

Pour appliquer les algorithmes de résolution de POMDP (tel que PBVI [Pineau 2004] ou Perseus [Spaan et Vlassis 2005]), il est nécessaire de disposer d'un modèle de transitions et d'observations complet. Le paragraphe 2.5.1 décrit la modélisation d'un tel modèle à partir d'un corpus annoté manuellement et automatiquement. L'utilisation de méthodes de repli permet d'obtenir ce modèle malgré la faible quantité de données.

Les optimisations ont été réalisées sur des corpus générés par simulation. Les paragraphes 2.5.2 et 2.5.3 décrivent l'utilisation d'un environnement de simulation pour apprendre une stratégie optimale.

2.5.1 Apprentissage par renforcement sur modèle complet

On pourra consulter [Sutton et Barto 1998] pour voir la différence entre les résolutions avec modèle complet ou sans modèle. Les algorithmes sur modèle complet font l'hypothèse que le modèle de probabilité (transitions et observations) est connu. Ce modèle peut être estimé sur un corpus de dialogues réels ou simulés, comme nous le proposons ensuite. Tandis que lors d'une résolution par un algorithme dit « sans modèle », le système fait intervenir une interaction avec l'environnement, les transitions sont réalisées par une évolution réelle de l'environnement et les observations sont en fait de véritables mesures expérimentales *a priori* moins facilement contrôlables (sauf dans un environnement de simulation).

i Environnement de simulation

[Rieser et Lemon 2008] utilise les données d'un corpus obtenu par la technique du magicien d'Oz pour entraîner un environnement de simulation sur lequel sera optimisée la stratégie du système. Lorsqu'on dispose d'un corpus de dialogue tel que le corpus MEDIA [H. Bonneau-Maynard 2004], on peut se tourner vers des méthodes de résolution avec modèle complet. Il est nécessaire de définir l'état s , l'observation o et l'action a dans le corpus. Ensuite, la génération du modèle est théoriquement une simple mesure empirique pour estimer les probabilités de transition $P(s'|s, a)$ et d'observation $P(o|s)$.

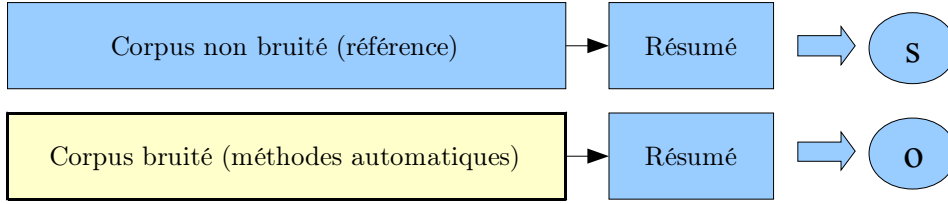


Figure 2.8: Élaboration des états et des observations du corpus résumé.

Comme l'illustre la figure 2.8, l'état exact de dialogue s est défini à partir des données de référence du corpus : celles qui ont été annotées manuellement. L'observation o est en revanche définie à partir des données bruitées : celles qui ont été générées par les outils d'annotation automatiques.

Le corpus de dialogue MEDIA est annoté de deux manières, manuellement et automatiquement, et peut être utilisé à cet effet. Nous utiliserons également d'autres corpus, générés par des simulations. Nous prendrons soin, lors de la génération de ces corpus, de générer des données exactes non bruitées ainsi que les données utilisant ou simulant les outils de transcription et d'annotation automatiques.

ii Modèle empirique

A priori il suffit de compter les occurrences dans le corpus \mathcal{C} :

$$P(o = o_o | s = s_o) = \frac{\text{card}\{(o, s) \in \mathcal{C} / o = o_o \text{ et } s = s_o\}}{\text{card}\{(o, s) \in \mathcal{C} / s = s_o\}} \quad (82)$$

De même pour les probabilités de transition :

$$P(s' = s'_o | a = a_o, s = s_o) = \frac{\text{card}\{(s, a, s') \in \mathcal{C} / s = s_o, a = a_o \text{ et } s' = s'_o\}}{\text{card}\{(s, a, s') \in \mathcal{C} / s = s_o \text{ et } a = a_o\}} \quad (83)$$

L'utilisation du modèle empirique défini par (82) et (83) au lieu du modèle exact entraîne une erreur sur le modèle utilisé. [Mannor et al. 2004] étudie l'erreur induite sur la stratégie optimale d'un MDP (représentée par sa fonction valeur V).

iii Degrés de liberté

Un rapide calcul du nombre de degrés de liberté d'un modèle de probabilité POMDP montre que la quantité de données nécessaire est rapidement inaccessible. En effet, notons n_s , n_a et n_o le nombre d'états, d'actions et d'observations du modèle POMDP. Pour le modèle d'observation $P(o|s)$, il y a $n_o \cdot (n_s - 1)$ degrés de liberté. Pour le modèle de transition $P(s'|s, a)$, il y a $n_s \cdot (n_a \cdot n_s - 1)$ degrés de liberté. Le tableau 2.17 propose une application numérique. Dans le prototype développé lors de cette thèse, on a utilisé des POMDP comportant 8 actions, et dont le nombre d'états et d'observations sont de l'ordre de quelques dizaines ($5 < n_s < 27$ et $6 < n_o < 19$).

Nom du paramètre	Symbole	Cas 1	Cas 2	Cas 3
Nombre d'états	n_s	10	20	100
Nombre d'actions	n_a	8	8	8
Nombre d'observations	n_o	10	20	100
Nom de la variable				
Nombre de degrés de liberté du modèle de transition : $P(s' s, a)$	$n_s \cdot (n_a \cdot n_s - 1)$	790	3180	79900
Nombre de degrés de liberté du modèle d'observation : $P(o s)$	$n_o \cdot (n_s - 1)$	90	380	9900

Tableau 2.17: Nombre de degrés de liberté du modèle de probabilité d'un POMDP.

iv Modèle de repli

Le repli pour l'estimation des probabilités d'observations $P(o|s)$ s'effectue lorsque le nombre d'occurrences du couple (o, s) ne dépasse pas un certain seuil (de 1 à 5 selon les expériences). Un seuil trop élevé entraîne de nombreux replis mais un seuil trop bas donne un modèle moins robuste.

De même, en ce qui concerne les observations, lors de l'estimation des probabilités de transitions $P(s'|a, s)$, le repli s'effectue lorsque le nombre d'occurrences du triplet (s, a, s') ne dépasse pas un seuil donné.

Des méthodes de *discounting* lors d'un repli permettent de renormaliser correctement les valeurs des probabilités telles que celles de Witten-Bell [Witten et Bell 1991] que nous utiliserons, de Ney [Andrés-Ferrer et Ney 2009] ou de Ristad [Ristad et al. 1995]. D'autres méthodes ont été proposées, voir par exemple [Geist et al. 2009] pour une approche par filtrage bayésien reposant sur l'utilisation de fonctions noyaux.

v Quantité de données

Concernant la quantité de données finalement nécessaire pour que l'emploi de méthodes statistiques d'apprentissage par renforcement soit fondé, [Keizer et al. 2010] utilise un corpus de 541 dialogues (et 4479 tours de parole) pour apprendre les paramètres d'un utilisateur simulé. [Schatzmann et al. 2007a] recommande d'utiliser plus de 10000 dialogues pour apprendre une stratégie MDP.

2.5.2 Simulations d'utilisateurs avec modèle de *frames* sémantiques

Les simulations ont pour but de générer des corpus de dialogues variés et réalistes. Différents modèles ont été proposés (voir 1.4), nous avons choisi de réutiliser le modèle d'utilisateur [Schatzmann et al. 2007a] reposant sur un agenda [Wei et Rudnicky 1999].

<p>Tours de parole</p>	<p>Transcription (NLG) Acte de dialogue au format Umlib . Représentation en arbre de <i>frames</i> sémantiques.</p>	
<p>USR 1</p>	<p>Je voudrais réserver un hôtel avec restaurant.</p> <p><code>inform(restaurant=yes)</code></p>	<pre> graph TD RESERVATION[RESERVATION] -- object --> HOTEL[HOTEL] HOTEL -- facilities --> FACILITY[FACILITY type = restaurant] </pre>
<p>SYS 1</p>	<p>Je vous écoute.</p> <p><code>null()</code></p>	
<p>USR 2</p>	<p>Je voudrais réserver la ville rennes avec restaurant 15 juin 26 septembre sont les dates de réservation.</p> <p><code>inform(restaurant=yes, city=rennes, start_date=15_juin, end_date=26_septembre)</code></p>	<pre> graph TD RESERVATION[RESERVATION] -- for_time --> PERIOD[PERIOD] RESERVATION -- establishment --> HOTEL[HOTEL] PERIOD -- start_date --> DATE1[DATE month = juin day = 15] PERIOD -- end_date --> DATE2[DATE month = septembre day = 26] HOTEL -- at_loc --> LOCATION[LOCATION] HOTEL -- facilities --> FACILITY[FACILITY type = restaurant] LOCATION -- loc_adr --> ADDRESS[ADDRESS adr_city = rennes] </pre>
<p>SYS 3</p>	<p>A Rennes, il y a l'hôtel des Remparts qui possède un restaurant.</p> <p><code>inform(name=remparts, restaurant=yes, city=rennes)</code></p>	<pre> graph TD PROPOSE[PROPOSE] -- theme --> HOTEL[HOTEL name = Remparts] HOTEL -- at_loc --> LOCATION[LOCATION] LOCATION -- loc_adr --> ADDRESS[ADDRESS adr_city = rennes] </pre>
<p>USR 4</p>	<p>Oui</p> <p><code>ack()</code></p>	<pre> graph TD ACCEPT[ACCEPT] </pre>
<p>SYS 5</p>	<p>L'hôtel des Remparts qui possède un restaurant, confirmez vous votre réservation à Rennes pour l'hôtel des Remparts pour les dates que vous avez fournies ?</p> <p><code>confirm(name=remparts)</code></p>	

Tableau 2.18: Extrait d'une simulation de dialogue Umlib avec les représentations en actes de dialogue et en frames sémantiques associées.

Un environnement de simulation comportant un utilisateur simulé a été implémenté à Cambridge [Schatzmann et al. 2007a] dans une bibliothèque C++ réutilisable (Umlib). Afin que la simulation se situe au niveau intentionnel des structures de *frames* sémantiques et non plus de formulaire, il est nécessaire d’adapter la librairie Umlib.

Idéalement, tous les modèles utilisés dans l’utilisateur simulé devraient être adaptés afin de profiter de la nouvelle flexibilité du modèle de *frames* : les mécanismes de génération de but utilisateur, la structure de l’agenda, le modèle d’ancrage, etc. Cependant, en première approche, et pour accélérer notre travail, on a choisi de garder des structures internes reposant sur le modèle de formulaire (*slot-value*), tant pour le but utilisateur que pour l’agenda utilisateur. Ce qui permet aussi d’avoir un cadre plus propice à une comparaison effective des résultats. En revanche, on a ajouté un module de génération de structures de *frames* sémantiques afin que l’utilisateur simulé génère des représentations sémantiques en accord avec notre paradigme de sémantique riche.

Le tableau 2.18 montre un exemple de dialogues en actes de dialogue et en graphes de *frames* sémantiques issu d’un corpus simulé.

2.5.3 Principe de simulations itératives

Les simulations sont réalisées avec un système de dialogue possédant une stratégie issue de la combinaison d’une stratégie POMDP π et d’une hiérarchie de règles π_{rules} . Ces stratégies sont combinées à travers leurs fonctions Q selon les méthodes décrites en 2.2.

L’optimisation de la stratégie POMDP π est effectuée de façon itérative selon le processus de l’algorithme 2.19 illustré par la figure 2.9. L’algorithme Perseus utilisé est un algorithme *off-line* : avant l’optimisation de la stratégie par Perseus, on a fixé la stratégie du système qui permet de générer le corpus permettant d’estimer le modèle de transition et d’observation. Cependant, du fait des itérations successives, on s’approche d’une résolution *on-line* où le modèle de transition et d’observation reflète le comportement de la stratégie en cours d’optimisation.

Évaluation des stratégies

En analysant le corpus des dialogues simulés obtenus lors d’une itération, on dispose d’une métrique permettant l’évaluation de la stratégie suivie lors de cette itération. Cependant, cette évaluation est biaisée du fait de l’utilisation d’une stratégie exploratoire (de coefficient ϵ). Par conséquent, une évaluation correcte des stratégies intermédiaires nécessite de générer de nouvelles simulations de dialogues.

Algorithme 2.19 : Principe de simulations itératives.

Initialisation :

La stratégie initiale π_0 est uniforme : π_0 affecte le même score à toutes les actions. Elle donne ainsi un avis neutre et laisse le choix de l'action à prendre à l'autre stratégie π_{rules} .

Le corpus initial est vide $C_0 = \emptyset$.

Itération :

1. On génère un corpus de N dialogues par N simulations : d_1, \dots, d_N . Le système utilisé a une stratégie POMDP π_k (combinée à une stratégie de règles comme décrit en 2.2) avec coefficient d'exploration ϵ et l'utilisateur simulé génère des *n-best* d'observation avec un niveau de bruit donné e .
 2. Le corpus s'enrichit des nouvelles simulations : $C_{k+1} = C_k \cup \{d_1, \dots, d_N\}$.
 3. Le modèle de probabilité complet (transitions et observations) est calculé d'après le corpus C_{k+1} ainsi obtenu.
 4. La résolution du POMDP obtenu est effectuée par l'algorithme Perseus [Spaan et Vlassis 2005] pour obtenir la stratégie POMDP suivante π_{k+1} .
 5. Répéter l'étape 1 avec $k + 1$.
-

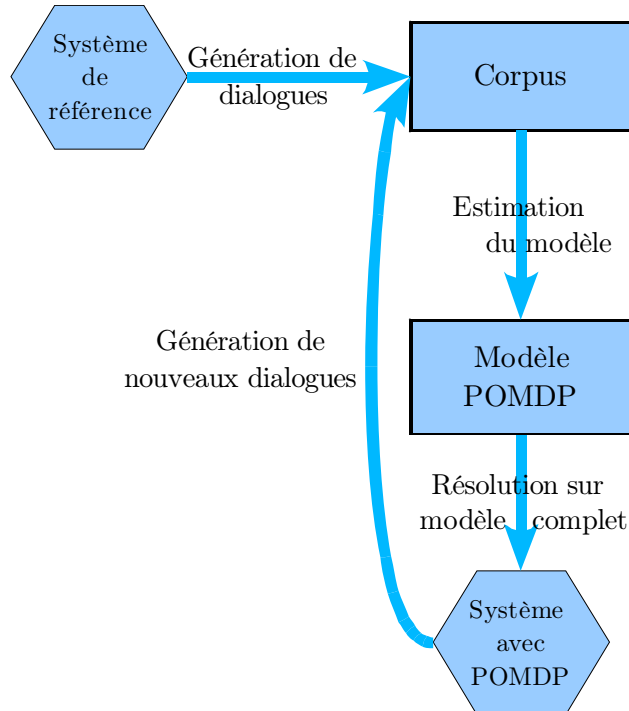


Figure 2.9: Procédé itératif d'optimisation de stratégie POMDP.

Chapitre 3 : Application et Résultats

Sommaire

3.1 Méthodologie d'évaluation.....	118
3.1.1 Tâches.....	119
3.1.2 Critères objectifs	121
3.1.3 Critères subjectifs.....	123
3.1.4 Synthèse.....	124
3.2 Étude préliminaire dans les espaces résumés.....	125
3.2.1 Système.....	125
3.2.2 Optimisation de stratégie de dialogue.....	126
3.2.3 Évaluation.....	126
3.2.4 Résultats.....	127
3.2.5 Conclusion et limites.....	128
3.3 Implémentation d'un système couplé avec un POMDP résumé.....	130
3.3.1 Architecture d'un prototype.....	130
3.3.2 Modèle sémantique : mise à jour d'un graphe sémantique.....	132
3.3.3 Modèle sémantique d'ancrage.....	133
3.3.4 Passage aux espaces résumés.....	136
3.3.5 Retour à l'espace maître.....	137
3.3.6 Couplage entre règles et POMDP.....	137
3.4 Optimisation de stratégie POMDP par simulations.....	138
3.4.1 Mise en œuvre des simulations.....	138
3.4.2 Entraînement du POMDP résumé.....	141
3.4.3 Suivi de croyance maître.....	143
3.4.4 Comparaison à un MDP.....	144
3.4.5 POMDP résumés par clustering automatique.....	147
3.4.6 Couplage de systèmes maître-résumé.....	149
3.4.7 Limites.....	150
3.5 Tests utilisateurs	152
3.5.1 Mise en œuvre.....	152
3.5.2 Évaluation objective.....	154
3.5.3 Évaluation subjective.....	156
3.5.4 Conclusion et limites	162

Le but de ce chapitre est de tester en pratique les propositions exposées dans le chapitre 2 et notamment de présenter les résultats des différentes expériences, menées dans le domaine des renseignements touristiques et de réservation de chambres d'hôtel, qui justifient l'intérêt porté aux méthodes d'apprentissage par renforcement dans les systèmes de dialogue homme-machine.

La partie 3.1 expose et justifie la méthode d'évaluation choisie, avec les critères d'évaluation utilisés.

La partie 3.2 expose les résultats de notre première piste de travail : il s'agit d'expériences préliminaires par simulation sur les espaces résumés d'état et d'observation, avec des observations en *1-best* sur des *frames* sémantiques structurées. Le principal intérêt de ces expériences simulées est de mettre en évidence l'intérêt de l'approche mais elles ne constituent pas une validation complète car une implémentation réelle est nécessaire.

Convaincu de la pertinence de notre approche par les résultats de cette première expérience, et afin d'aller plus loin dans l'application pratique du paradigme de POMDP résumé à double suivi de croyance, on présente en partie 3.3 les détails permettant la mise au point d'un prototype complet. L'emploi de graphes de *frames* sémantiques pour le dialogue nécessite d'implémenter un module de compréhension intégrant cette nouvelle représentation (en incluant un modèle d'ancrage). L'utilisation d'un POMDP résumé pour le dialogue couplé à un système déterministe est précisée ensuite : la façon d'effectuer les projections résumées, de revenir dans l'espace d'actions maitres et d'assembler les deux scores maitre et résumé. Le prototype de système de dialogue présenté permet de mettre en œuvre le couplage d'un module de décision déterministe et d'un POMDP utilisant des observations *n-best* de *frames* sémantiques.

Afin de tester l'efficacité du paradigme proposé, nous avons exploré l'influence de différents paramètres sur les performances du système : taux d'erreurs de reconnaissance vocale et de compréhension, précision des suivis de croyance dans les espaces maitre et résumé, etc. La partie 3.4 détaille l'utilisation d'environnements de simulation pour optimiser le système de dialogue et analyse les gains de performances obtenus.

Finalement la partie 3.5 décrit les tests réalisés avec de véritables utilisateurs dans des conditions réalistes. Elle s'appuie en outre sur une analyse contrastive d'un système de référence (sans renforcement) avec un système comportant un POMDP résumé optimisé par renforcement.

3.1 Méthodologie d'évaluation

Résumé : L'évaluation de la réussite ou de l'échec d'un dialogue est réalisée en testant les systèmes sur des tâches d'informations touristiques. Les critères d'évaluation sont objectifs (du point de vue d'un observateur omniscient) ou subjectifs (du point de vue de l'utilisateur).

Avant de mener des expériences dont le but est de tester les performances d'un système, il est nécessaire d'avoir une vision claire des méthodes d'évaluation qui vont permettre de juger de ces performances. Le but de cette partie est de décrire l'ensemble des critères qui vont être utilisés pour tester les différentes stratégies du prototype. Les évaluations proprement dites comprenant les performances des systèmes sont présentées dans les parties correspondantes (étude préliminaire avec modélisation au niveau résumé en partie 3.2, dialogues simulés en partie 3.4 ou avec utilisateurs réels en partie 3.5).

Comme évoqué dans la partie 1.5 : « Évaluation », les critères d'évaluation des performances d'un système sont de deux types :

- Critères objectifs : expression d'un juge observant le dialogue, ils utilisent les données enregistrées lors du dialogue.
- Critères subjectifs : points de vue de l'utilisateur, ils font intervenir les intentions réelles de l'utilisateur qui ne sont pas accessibles directement à l'expérimentateur.

Les critères d'évaluation peuvent être également classés selon leur niveau de granularité. Les notions de récompenses immédiate ou cumulée (utilisées dans un POMDP) ainsi que celle de performance globale d'un système (nécessaire pour l'évaluation) se situent à des niveaux de granularité différents selon que l'on cherche à évaluer un tour de dialogue, un dialogue complet ou un ensemble de dialogues.

Niveau de granularité	Évaluation
Tour	Récompense immédiate
Dialogue	Récompense cumulée
Ensemble de dialogues	Performance globale du système

Tableau 3.1: Trois niveaux de granularité pour l'évaluation.

A partir de ces grands principes, nous avons choisi de mettre l'accent dans notre démarche d'évaluation sur les critères objectifs et subjectifs qui nous semblent plus pertinents pour nos travaux, et l'ensemble de nos critères se situent aux niveaux de granularité du tour du dialogue.

La présentation débutera par l'exposé des tâches que le système (et l'utilisateur) doivent accomplir (partie 3.1.1), avant de développer les critères objectifs (partie 3.1.2) et les critères subjectifs (partie 3.1.3) que nous avons retenus. Pour finir, une synthèse présentera les différents critères d'évaluation disponibles selon les conditions de réalisation des dialogues (partie 3.1.4).

3.1.1 Tâches

La tâche 1 (tableau 3.2) est la tâche standard pour laquelle le système est prévu : réserver un hôtel selon certaines contraintes, à certaines dates, sous l'hypothèse qu'il est possible de trouver une réponse satisfaisante dans la base de données des hôtels.

Vous avez prévu un séjour à Limoges, vous voulez donc réserver un hôtel du 18 juin au 27 août. Vous êtes prêt à accepter n'importe quel hôtel mais vous tenez à avoir un hôtel avec la télévision.
Match 1 : name="hôtel du Chat Noir" id="134"
Match 2 : name="Bonne Nouvelle" id="52"

Tableau 3.2: Exemple de scénario pour la tâche 1 : réservation d'hôtel.

La tâche 2 (tableau 3.3) consiste à obtenir le prix d'un hôtel selon certaines contraintes. Cette tâche n'est pas celle pour laquelle a été élaboré l'ensemble de règles déterministes constituant le système de référence. Elle reste cependant simple à mettre en place dans la mesure où on s'assure également qu'il y a une réponse dans la base de données des hôtels. Cependant, l'utilisateur ne peut fournir au système toutes les données qu'il attend dans le cadre de la tâche 1 : il n'y a pas de dates de réservation. De plus, l'utilisateur ne souhaite pas effectuer une réservation, mais obtenir une information. Ainsi, l'action finale décidant de la réussite du dialogue n'est pas la même pour la tâche 1 et pour la tâche 2.

Vous prévoyez un séjour à Bayonne, vous cherchez un hôtel avec une piscine. Vous ne voulez pas faire de réservation mais seulement savoir combien coûte la chambre.
Match 1 : price="cinquante euros" name="velizy" id="173"
Match 2 : price="cent trois euros" name="valentine" id="66"

Tableau 3.3: Exemple de scénario pour la tâche 2 : recherche de prix.

La tâche 3 (tableau 3.4) est une tâche difficile : l'utilisateur ne peut obtenir ce qu'il souhaite car les contraintes qu'il fournit sont trop restrictives et par conséquent impossibles à satisfaire. Le système doit lui proposer des solutions alternatives et négocier la relaxation de certaines de ces contraintes. De plus, il devient souhaitable, voire nécessaire, de maintenir en mémoire plusieurs propositions, pour que le système se souvienne si ce qu'il a proposé a été accepté ou refusé, ou pour proposer plusieurs possibilités.

Vous avez prévu un séjour à Limoges du 10 juin au 27 septembre, vous avez absolument besoin de réserver un hôtel. Dans la mesure du possible, vous aimeriez un hôtel avec un cours de tennis, qui accepte les animaux, avec une salle de musculation et avec climatisation.	
Match :	None.
Alternative-match 1 :	name="hôtel de l'Arc de Triomphe" id="44" must_relax : animaux = false
Alternative-match 2 :	name="hôtel de la gare de Lyon" id="48" must_relax : climatisation = false

Tableau 3.4: Exemple de scénario pour la tâche 3 : réservation négociée.

En terme de complexité, si on transcrit les tâches sous la forme de but utilisateur constitué d'un formulaire, on aurait le formulaire du tableau 3.5. Soit 1 048 576 buts utilisateurs différents avec 16 champs ($32 \times 2^{13} \times 1 \times 2 \times 2$).

Champ	Nombre de valeurs	Valeurs
Ville	32	Besançon, Caen,...
13 valeurs binaires de contraintes telles que : <i>Avec jacuzzi</i>	2^{13} 2	Oui, non.
Nom de l'hôtel requis par l'utilisateur.	1	Oui (toujours).
Prix requis par l'utilisateur.	2	Oui, non.
Dates de début et de fin fournies par l'utilisateur	2	[18/06/10, 27/09/10], non.

Tableau 3.5: But utilisateur sous forme de formulaire.

Trois tâches avaient initialement été envisagées. Cependant, en raison de nos choix d’actes de dialogue adaptés aux annotations du corpus MEDIA, différents de ceux de l’utilisateur simulé de Cambridge d’une part, et des représentations sémantiques différentes d’autre part, les dialogues simulés pour la tâche 3 n’étaient pas réalistes et ne permettaient pas d’optimiser une stratégie de façon satisfaisante.

Nous sommes convaincus que ce sont des tâches mettant en œuvre une étape de négociation (du type de la tâche 3) qui sont l’objectif des systèmes de dialogue futurs. Cependant, lors des expériences avec utilisateur simulé et par conséquent avec les utilisateurs réels, nous nous sommes concentrés sur les tâches 1 et 2. L’adaptation du système et de l’utilisateur simulé pour réaliser des tâches de négociation du type de la tâche 3 sera l’objet de futurs travaux.

3.1.2 Critères objectifs

Les critères objectifs sont calculés à partir des données enregistrées durant les dialogues (les fichiers de traçage ou *logs*).

i Récompense immédiate réelle

Comme présenté dans le chapitre 1, la définition d’une récompense immédiate $r(s, a, s')$ est nécessaire pour effectuer l’entraînement d’un POMDP.

Dans le cas où l’on dispose de la connaissance des états exacts du dialogue, du but exact de l’utilisateur et de la base de données, il est possible de savoir si le système a correctement répondu à la demande de l’utilisateur. Cela permet de définir le succès ou l’échec d’un dialogue (tableau 3.6).

Succès	Une réservation correspondant au but de l’utilisateur a été effectuée.
Échec	Aucune réservation effectuée, ou
	Une réservation ne correspondant pas au but de l’utilisateur a été effectuée.

Tableau 3.6: Définition du succès et de l’échec d’un dialogue avec connaissance du but utilisateur et de la base de données.

En définissant les deux états finaux absorbants correspondant à un dialogue se terminant par un succès (END-OK) ou par un échec (END-NOK) on définit le critère de récompense immédiate du tableau 3.7.

Condition	Définition	Valeur de récompense
$s \neq s' = \text{END-NOK}$	Le dialogue vient de se terminer par un échec.	-20.0
$s \neq s' = \text{END-OK}$	Le dialogue vient de se terminer par un succès.	+10.0
$s = s' = \text{END-OK}$ ou $s = s' = \text{END-NOK}$	Les états finaux END-OK et END-NOK sont absorbants et ne génèrent plus de récompense.	0.0
Tous les autres cas.	Par défaut, chaque tour de parole génère un léger malus.	-1.0

Tableau 3.7: Récompense immédiate avec information sur le but exact de l'utilisateur et sur la base de données.

Récompense immédiate réelle (version approximée)

Lorsque la base de données n'est pas disponible, par exemple lorsque l'on utilise le corpus MEDIA, le succès ou l'échec d'un dialogue ne peuvent plus être définis d'après le but utilisateur. Ce critère utilise l'action et la connaissance des états exacts du dialogue, mais n'utilise pas de résultats de la base de données ni le but exact de l'utilisateur.

Condition	Définition	Valeur de récompense
$a = \text{CLOSING}$ et $s' \neq \text{END}$	Tentative avortée de terminer le dialogue.	-20.0
$a = \text{CLOSING}$ et $s' = \text{END}$	Tentative réussie de terminer le dialogue.	+10.0
$s = \text{END}$ et $s' = \text{END}$	L'état final END est absorbant et ne génère plus de récompense.	0.0
Tous les autres cas.	Par défaut, chaque tour de parole génère un léger malus.	-1.0

Tableau 3.8: Récompense immédiate rudimentaire.

Cette définition de la récompense immédiate est très rudimentaire et repose sur l'hypothèse que l'utilisateur refuse de terminer le dialogue tant qu'il n'est pas satisfait. Ainsi, le système peut proposer la fin du dialogue à l'utilisateur par l'action $a = \text{Closing}$ et le dialogue ne peut se terminer que sur un succès. C'est pourquoi on introduit un mécanisme d'arrêt du dialogue après 50 tours de dialogue afin de pénaliser les dialogues sans succès par une récompense négative assez grande.

ii Longueur du dialogue

Un critère de performance objectif simple à mettre en place est celui de la longueur du dialogue. Il intervient dans les deux critères précédents de récompenses immédiates réelles (exacte ou approximée) à travers le léger malus de -1.0 correspondant à chaque tour de dialogue.

iii Complétion de la tâche

Le critère de complétion de la tâche est défini pour un dialogue. C'est un critère booléen valant 0 ou 1 si le dialogue se termine respectivement par un échec ou un succès (au sens défini par la table 3.6.)

3.1.3 Critères subjectifs

Les critères subjectifs s'expriment selon le point de vue de l'utilisateur et sont générés par l'utilisateur lui-même. Il est cependant difficile de demander à un utilisateur réel de fournir une évaluation immédiate pour chaque tour de dialogue; en effet, cette contrainte, plus que rébarbative, nuirait à l'expérience elle-même en interrompant le fil du dialogue.

i Satisfaction finale de l'utilisateur

La satisfaction finale de l'utilisateur peut être évaluée en demandant à l'utilisateur de donner son avis à la fin du dialogue. Par exemple, dans le cadre de notre prototype, deux boutons permettent de terminer le dialogue, ils expriment de manière très succincte mais claire le sentiment dominant (la réussite satisfaisante de « J'ai terminé » ou l'échec exaspérant d'un « J'abandonne ») :



Ce critère est également utilisable lors des simulation puisque l'utilisateur simulé offre la possibilité de reproduire ce comportement. En effet, l'utilisateur possède un état interne de *patience*, variable entière fixée arbitrairement avant chaque dialogue et qui décroît à chaque répétition d'acte de dialogue et l'utilisateur simulé interrompt le dialogue lorsque cette patience est à bout ($patience = 0$).

ii Réponses de l'utilisateur au questionnaire

Pour obtenir des informations plus précises sur la satisfaction de l'utilisateur, nous avons préparé un court questionnaire (6 questions et une note quantitative) centré autour des critères de réussite, compréhension, initiative, facilité d'utilisation. Ce questionnaire, rempli par les utilisateurs réels, est décrit de façon détaillée dans la partie 3.5 et est reproduit en annexe B.2.

3.1.4 Synthèse

Selon les données disponibles, seuls certains critères ont un sens. Il y a trois conditions possibles : les dialogues du corpus MEDIA, les dialogues simulés (au niveau intentionnel, avec l'utilisateur simulé de Cambridge UMLIB), et les dialogues réels avec des utilisateurs humains.

Pour un dialogue du corpus MEDIA, on ne dispose pas de la base de données associée, ni de scénario donnant le but véritable de l'utilisateur et les utilisateurs n'ont pas rempli le questionnaire. On se contente donc des critères objectifs de récompense immédiate approximée et de longueur.

Pour un dialogue avec des utilisateurs réels, on ne dispose pas des actes de dialogue exacts qui ont été prononcés par l'utilisateur, par contre on a accès au scénario donnant le but véritable de l'utilisateur ainsi que les réponses au questionnaire donnant une évaluation finale.

Pour un dialogue simulé, et c'est le cas le plus favorable, on dispose des actes de dialogue exacts ainsi que du scénario et d'une évaluation booléenne de satisfaction finale de l'utilisateur simulé. On peut utiliser toutes ces informations. En revanche, on n'a pas modélisé l'utilisateur suffisamment finement pour obtenir un équivalent aux réponses au questionnaire. Une possibilité pour cela serait d'utiliser le paradigme PARADISE (voir 1.5.3).

La table 3.9 donne un récapitulatif des critères disponibles pour évaluer les dialogues obtenus dans les différentes conditions possibles.

Critère	Corpus MEDIA	Utilisateurs réels	Utilisateurs simulés
Récompense immédiate 1	Oui	Oui	Oui
Récompense immédiate 2	Non	Non	Oui
Longueur du dialogue	Oui	Oui	Oui
Complétion réelle de la tâche	Non	Oui	Oui
Satisfaction finale de l'utilisateur	Non	Oui	Oui
Réponses au questionnaire	Non	Oui	Non

Tableau 3.9: Disponibilité des critères d'évaluation des dialogues selon les conditions de réalisation.

3.2 Étude préliminaire dans les espaces résumés

Résumé : Un premier test de l'approche de POMDP résumé à double suivi de croyance peut être effectué en se plaçant dans les espaces résumés uniquement. On montre qu'un POMDP appris d'après un corpus semble robuste aux erreurs ASR.

Tout d'abord, nous avons procédé à une première expérience afin de considérer l'intérêt de l'approche probabiliste par modèle POMDP. En effet, il s'agit dans un premier temps de vérifier l'intérêt potentiel de l'utilisation de l'apprentissage par renforcement pour le dialogue avant de l'appliquer au sein d'un système réel.

Cette expérience utilise l'annotation en *frames* sémantiques du corpus MEDIA (voir partie 1.1.2.4iv). Elle consiste à mettre en œuvre un algorithme d'optimisation de POMDP sur modèle complet au niveau des espaces résumés. L'évaluation des performances est effectuée par des simulations dans les espaces résumés et la robustesse est testée en observant l'influence des erreurs ASR sur les performances du système [Pinault et al. 2009].

Après avoir décrit la mise en place du POMDP résumé (en 3.2.1) ainsi que l'optimisation de sa stratégie (en 3.2.2). Nous présentons ensuite les deux jeux de données permettant de comparer les performances du système dans des conditions de bruit différentes (en 3.2.3). Les résultats sont présentés en partie 3.2.4, ainsi que les conclusions de cette expérience préliminaire et ses limites (en 3.2.5).

3.2.1 Système

Le système POMDP considéré est un POMDP résumé comme décrit en partie 2.3. Pour définir un corpus résumé, on utilise le corpus MEDIA (1.1.2.4iv). Il comporte des données de références pour élaborer l'état résumé s et des données bruitées pour élaborer l'observation o :

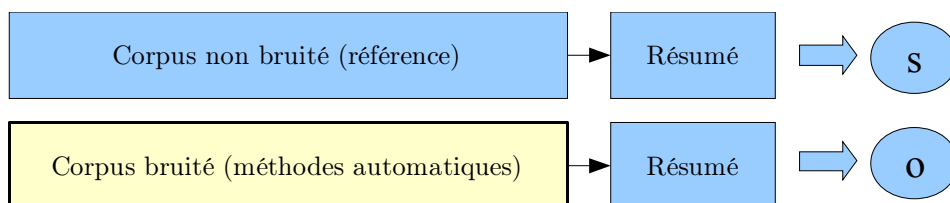


Figure 3.1: Élaboration des états et des observations du corpus résumé.

État : La projection sur l'espace d'état résumé est effectuée d'après le corpus de référence (non bruité). Elle est définie par des règles logiques sur des *frames* de référence des dialogues.

Observation : la projection sur l'espace d'observation résumé utilise en revanche les données d'un corpus bruité. Elle est définie de la même façon que la projection d'état résumé, mais elle utilise l'annotation en *frames* qui est observée : celle qui est obtenue automatiquement d'après le premier des *n-best*.

Action : La projection dans l'espace résumé des actions est définie d'après le tour de parole du magicien d'Oz qui n'est pas bruité. Elle utilise des règles *ad hoc* permettant une détection de motifs sur la transcription texte des tours de parole systèmes.

Le modèle complet de probabilité est alors appris sur ce corpus résumé en effectuant un repli lorsque la quantité de données est insuffisante pour assurer une estimation fiable des probabilités de transition et d'observation (voir partie 2.5.1). La stratégie est alors apprise sur le modèle complet.

Cette stratégies est une stratégie résumée : elle ne fournit qu'une action résumée. Dans un système de dialogue réel, il serait nécessaire de revenir dans l'espace des actions maîtres. Cependant, lors de cette étude préliminaire, on fait l'économie du retour aux espaces maîtres puisque la simulation et l'évaluation peuvent être effectuées en restant au niveau résumé.

3.2.2 Optimisation de stratégie de dialogue

Récompense

La récompense est fixée selon un critère de récompense immédiate similaire à ceux décrit en partie 3.1.2 dans le tableau 3.8.

Résolution

Concernant la résolution du POMDP pour obtenir une stratégie optimale, on a utilisé l'algorithme itératif PBVI [Pineau et al. 2003] où l'optimisation est interrompue en fixant l'horizon de planification à une valeur finie.

3.2.3 Évaluation

Critère d'évaluation

L'évaluation est effectuée en générant des simulations de dialogues au niveau résumé, en réutilisant le modèle de probabilité, et en considérant la récompense moyenne.

Robustesse aux erreurs ASR

Afin de tester la robustesse des systèmes POMDP, on se propose de comparer deux systèmes plus ou moins bruités. Pour cela, on utilise un corpus de dialogues (signal acoustique, transcriptions et annotations sémantiques) dont on dispose d'une version de référence et de deux versions bruitées, les deux systèmes comparés utilisent chacun l'une des deux versions de corpus bruités.

Les actions résumées sont définies de façon identique par recherche de motifs sur les tours de parole du système (magicien d'Oz).

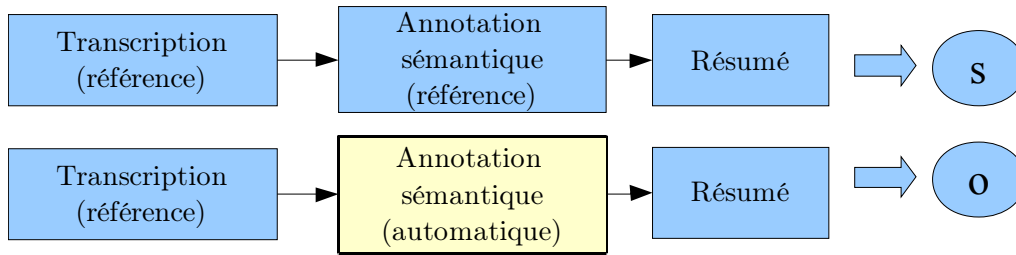


Figure 3.2 : Système « SLU », sans erreur de reconnaissance (annotation de référence pour l'ASR) et avec erreur de compréhension SLU.

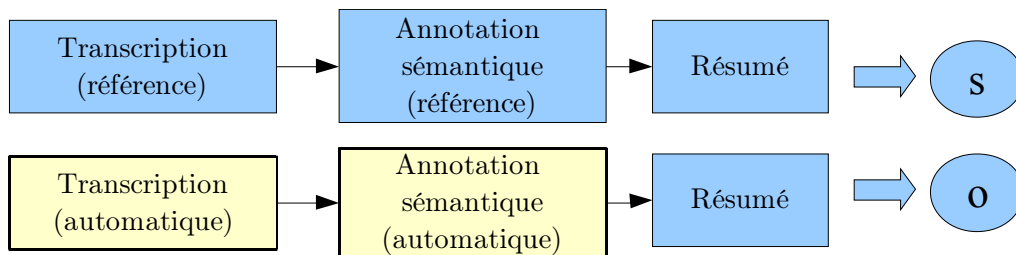


Figure 3.3 : Système « ASR+SLU », avec erreur de reconnaissance ASR et de compréhension SLU.

3.2.4 Résultats

L'évaluation montre une amélioration des récompenses moyennes obtenues quand l'horizon augmente, ce qui traduit le fait que le système apprend correctement une stratégie correspondant mieux au modèle de probabilité et de récompense qu'on lui fournit.

Sur les résultats de la figure 3.4 présentant les récompenses moyennes obtenues par les deux systèmes selon la taille de l'horizon de résolution, on constate tout d'abord une nette augmentation des performances du système lorsque l'horizon passe de 1 à 2. Ceci montre que la planification est efficace : le système trouve une meilleure stratégie en prenant en compte l'évolution du dialogue au tour suivant. En revanche, les courbes sont très rapidement stables : il semble inutile d'effectuer la planification au delà de l'horizon 2. Cet horizon de planification s'avère très faible pour effectuer une planification réellement efficace. Cependant, il est lié à la présence de phénomènes à très court terme (paires adjacentes notamment). Ces derniers sont bien pris en compte par les modèles de bigrammes [Eckert et al. 1997] qui ont fourni une première approche de la gestion du dialogue par des méthodes statistiques.

D'autre part, on constate peu de différence entre les deux systèmes ASR et ASR+SLU. Cela laisse supposer que le mécanisme de POMDP résumé est robuste face aux erreurs ASR (WER de 33%).

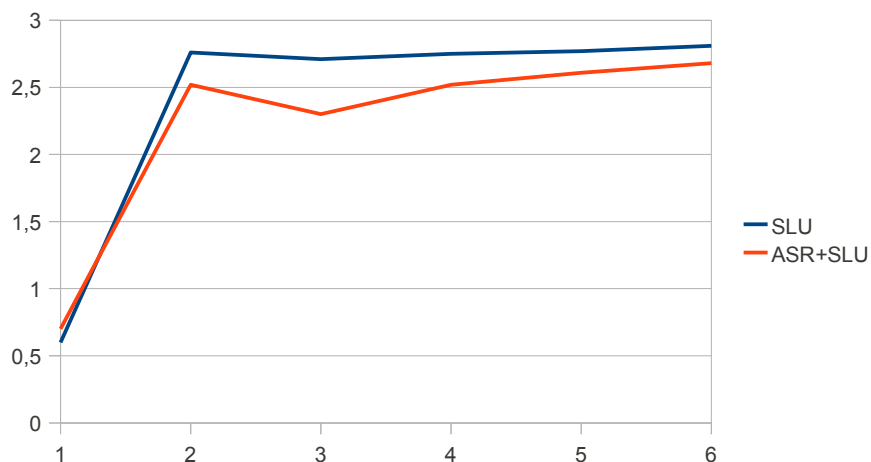


Figure 3.4: Effet du bruit ASR sur les performances du système résumé selon différents horizons de résolution avec l’algorithme PBVI.

3.2.5 Conclusion et limites

Ces expériences préliminaires ne sont pas suffisantes pour justifier complètement l’utilisation de méthodes de renforcement pour les systèmes de dialogue. Les principales limitations viennent du fait de réutiliser le même modèle pour l’optimisation et l’évaluation, le passage à l’espace résumé, et l’utilisation de données d’apprentissage très différentes des données de test.

i Réutilisation du modèle

L’évaluation est réalisée par de nouvelles simulations en réutilisant le modèle de probabilité de transition et d’observation sur lequel a été effectué l’apprentissage. Ce double emploi du même modèle ou des mêmes conditions, pour l’optimisation de la stratégie et pour son évaluation, comporte un biais certain. C’est cependant une méthode souvent utilisée pour tester les systèmes de dialogue [Schatzmann 2008]. En réalité, une évaluation plus pertinente demanderait d’utiliser d’autres conditions par exemple un autre corpus, mais on ne dispose pas de suffisamment de corpus comparables au corpus MEDIA en terme de tâches et d’annotation sémantique.

ii Projections résumées

La modélisation au niveau résumé entraîne un lissage important. C’est peut-être ce qui explique la stabilité face aux erreurs de reconnaissance vocale ASR. Il est nécessaire de réaliser des expériences dans des conditions plus proches de l’utilisation réelle d’un système de dialogue pour vérifier cette hypothèse.

iii Données d'apprentissage non représentatives

La stratégie suivie dans le corpus n'est pas celle du système réel. Les dialogues du corpus, bien que très réalistes car réalisés par un véritable utilisateur et un magicien d'Oz, ne sont donc pas représentatifs de ceux qui seraient générés en utilisant le système optimisé. Dans la mesure du possible, il serait souhaitable que le système utilisé pour générer le corpus soit proche du système optimisé final. Le mécanisme de simulation et d'optimisation itératif décrit en 2.5.3 vise à dépasser cette limitation. Il est mis en œuvre lors des expériences avec le prototype que nous avons développé.

3.3 Implémentation d'un système couplé avec un POMDP résumé

Résumé : On présente une implémentation possible du modèle sémantique par frames dans un système de dialogue effectif, ainsi que l'intégration des accès à la base de données. Le retour de l'espace d'actions résumées à l'espace d'actions maîtres est effectué selon une fonction heuristique et la stratégie POMDP est couplée avec une stratégie de règles simples pour assurer une meilleure complétude de conception (VUID). Cette implémentation sous la forme d'un prototype complet permet de tester les propositions précédentes.

Cette partie décrit l'implémentation pratique du modèle de *frames* sémantiques à travers le prototype utilisé pour réaliser les expériences dans des conditions réelles (utilisateurs humains, environnement bruité et scores *n-best* de l'ASR et du SLU). Ce prototype est présenté en partie 3.3.1.

Nous organisons le reste de notre présentation en suivant la boucle de traitement du système de manière temporelle, afin de détailler la propagation des nouvelles informations issues du tour de parole de l'utilisateur jusqu'à la génération de la réponse de la machine.

Ainsi, l'agrégation de nouveaux graphes sémantiques est réalisée par un algorithme déterministe (partie 3.3.2) associé à un modèle d'ancrage généralisé aux *frames* (partie 3.3.3). Les projections sur les espaces résumés sont décrites en partie 3.3.4.

En partie 3.3.5, on présente le retour depuis les espaces résumés dans les espaces maîtres. La génération d'une action maître à partir d'une action résumée utilise les éléments ayant constitué les requêtes à la base de données et certaines informations contenues dans l'état du dialogue courant.

Puis on décrit une façon d'implémenter un couplage entre les systèmes de dialogue par règles logiques et par POMDP (partie 3.3.6).

3.3.1 Architecture d'un prototype

Les communications entre les différents modules sont réalisées selon le principe du *fat pipeline* : les données échangées ne sont plus de nature déterministe mais sont constituées de listes *n-best* ou de treillis de mots.

Reconnaissance vocale (ASR)

Le module de reconnaissance vocale utilisé est Speeral, développé au laboratoire d'Avignon. Il utilise un modèle de *n*-grammes et procède à plusieurs passes pour effectuer le décodage [Linarès et al. 2007]. Les modèles linguistiques utilisés sont spécifiques au domaine

d'informations touristiques et de réservation de chambres d'hôtels, la taille du vocabulaire est de 2153 mots. Le modèle acoustique est un modèle adapté au signal audio de conversations téléphoniques. Le taux d'erreur de reconnaissance sur le corpus MEDIA est de 33,5%. Le module ASR fournit un treillis de mots au module de SLU.

Compréhension sémantique (SLU)

Le module de compréhension se décompose en plusieurs parties.

- La compréhension en concepts est effectuée en utilisant des modèles statistiques tels que ceux décrits dans [Lefèvre 2007; Jabaian et al. 2010]. Une liste *n-best* d'annotations en concepts est ainsi générée à partir du treillis reçu.
- Puis la compréhension en *frames* sémantiques est fondée également sur les principes statistiques : selon les versions du prototype, on utilise soit des modèles de réseaux bayésiens dynamiques (DBN¹) [Meurs 2009] ou des champs de Markov conditionnels (CRF²). La liste *n-best* d'annotations en concepts est ainsi transformée en liste *n-best* d'annotations en *frames* sémantiques.

Gestionnaire de dialogue (DM)

Le gestionnaire de dialogue reçoit une liste *n-best*, mémoire de *frames* sémantiques. À défaut de posséder une information certaine sur son environnement, il l'utilise pour maintenir une croyance sur l'état de dialogue. Cette croyance permet de maintenir plusieurs états de dialogues concurrents en parallèle.

Le gestionnaire de dialogue procède à la mise à jour de la structure des états de dialogues concurrents et des états d'ancrage. La liste *n-best* d'annotations en *frames* est intégrée dans une liste *n-best* qui est une mémoire de *frames* sémantiques. L'intégration d'un nouveau graphe de *frames* sémantiques dans un autre graphe de mémoire de *frames* est décrit en partie 3.3.2.

Puis le DM procède à l'intégration des résultats des requêtes à la base de données. Il fournit alors au modèle POMDP résumé une partie de l'information contenue dans le vecteur de croyance à travers l'observation résumée.

Une fois les actions maîtres générées, la prise de décision s'effectue en intégrant les scores du POMDP avec des scores *ad hoc* définis de façon experte.

Lors de l'apprentissage, pour générer le corpus sur lequel le modèle de transitions sera appris, le SLU fournit une observation exacte. L'état exact du système alors est connu et correspond à un fonctionnement en *1-best* sans erreur.

Génération de texte (NLG)

Un mécanisme par modèle (*templates*) de tour de parole du système permet de transformer la représentation symbolique en langage naturel. En pratique, cette génération de texte est effectuée en même temps que la génération de l'action maître. On trouvera des exemples dans la partie 2.2.1 : « Création d'actions maîtres ».

Synthèse vocale (TTS)

Le module de génération de parole utilisé est celui de Mbrola [Dutoit et al. 1996], bien qu'il n'utilise pas les méthodes récentes fondées sur une utilisation de chaînes de Markov

(1) DBN : *Dynamic Bayesian Network*.

(2) CRF : *Conditionnal Random Field*.

cachées, il a été choisi car c'est un des rares outils de synthèse vocale en français qui soit *open-source*. Ces travaux portant sur les modules de compréhension et de gestion du dialogue, aucun effort particulier n'a été réalisé pour améliorer ce module de synthèse vocale.

3.3.2 Modèle sémantique : mise à jour d'un graphe sémantique

Algorithme 3.10 : Algorithme de la fonction « `intégrerGraphe(G_1, G_2)` » permettant d'intégrer un nouveau graphe de frames sémantiques G_1 dans un graphe mémoire G_2 .

Entrées : G_1 graphe mémoire initial.

G_2 nouvel arbre à intégrer dans G_1 .

Sortie : G_1 graphe modifié.

Intégration de graphe : *intégrer le graphe G_2 dans G_1*

fonction `intégrerGraphe`(graphe G_1 , graphe G_2)

pour chaque frame de tête f_2 du nouveau graphe G_2

label 1:

chercher une frame f_1 dans G_1 où faire une réunion de frames (*merge*) pour f_2

si on a trouvé une frame f_1 pour la réunion

`intégrerFrame`(f_1, f_2)

sinon

créer une frame parente pour f_2 dans G_2 d'après une ontologie

aller à label 1

Intégration de frame : *intégrer la frame de f_2 dans f_1*

fonction `intégrerFrame`(frame f_1 , frame f_2)

pour chaque élément e_2 de f_2

chercher un élément e_1 de f_1 pour une réunion d'éléments (*merge*)

si on a trouvé un élément pour la réunion

`intégrerÉlément`(e_1, e_2)

sinon

ajouter e_2 dans f_1

Intégration d'élément de frame : *intégrer l'élément e_2 dans e_1*

fonction `intégrerÉlément`(élément e_1 , élément e_2)

si e_2 et e_1 sont de simples champs,

adapter la valeur du champ e_1 ainsi que le statut d'ancrage en fonction de e_2

si e_2 et e_1 sont des liens vers d'autres frames f_1' et f_2' ,

`intégrerFrame`(f_1', f_2')

Après traitement du signal acoustique par les modules ASR et SLU, de nouvelles informations représentées par une liste *n-best* de graphes de frames sémantiques sont disponibles

au gestionnaire de dialogue pour être intégré dans la représentation courante de l'état de dialogue maintenue par le système.

En pratique, dans notre modélisation POMDP, cette étape de mise à jour de la mémoire courante correspond à une mise à jour du vecteur de croyance maître. Ainsi, chacun de ces graphes est intégré dans chacun des graphes constituant le vecteur de croyance maître du système selon un produit croisé décrit en partie 2.3.2 : Espaces maîtres.

L'intégration d'un nouveau graphe de *frames* dans un graphe d'état, élément du vecteur de croyance maître est décrit dans l'algorithme 3.10. Il repose sur les principes exposés dans la partie 2.1.2.

3.3.3 Modèle sémantique d'ancrage

Dans cette partie, nous poursuivons le cours du processus de traitement d'un tour de parole de l'utilisateur en décrivant comment les statuts d'ancrage des différents éléments d'information contenus dans un graphes de *frames* sémantiques sont mis à jour en fonction des actes de dialogue présent dans les actions des interlocuteurs.

Afin d'utiliser un modèle d'ancrage pour les graphes de *frames* sémantiques, nous en avons développé un, inspiré du modèle d'ancrage utilisé avec des modèles sémantiques de formulaire.

Le modèle d'ancrage représente le statut de chaque champ dans l'esprit de chaque interlocuteur [Traum 1994; Roque et Traum 2008]. Le statut d'ancrage complet d'un champ est défini par l'historique des actes de dialogue qui l'ont concerné. Ces historiques sont regroupés en statuts d'ancrage résumés en utilisant une machine à états finis (FSM¹). On a également développé une FSM pour gérer les états d'ancrage, à l'instar de [Young et al. 2007], en ajoutant un statut d'ancrage sur les champs des *frames* sémantiques.

Pour un graphe de dialogue représentant un tour de dialogue, de l'utilisateur ou du système, chaque champ est tagué par un acte de dialogue d'ancrage. Pour le graphe représentant la mémoire de *frames*, chaque champ est muni d'une étiquette d'état d'ancrage. Les tableaux 3.11 et 3.12 détaillent les états d'ancrage et les actes d'ancrage qui ont été utilisés dans l'implémentation du modèle d'ancrage dans le prototype. Le graphe 3.5 illustre la FSM comportant les statuts d'ancrage (nœuds) et les actes de dialogue d'ancrage (transitions).

(1) FSM : *Finite State Machine*.

État d'ancrage	Signification
init	Aucun acte de dialogue n'a concerné cet élément d'information.
u/sInformed ²	Ce champ est vide et a été créé par l'utilisateur/le système.
u/sInformed(X)	Ce champ est rempli et a été créé par l'utilisateur/le système.
u/sRequested	Demande d'information sur ce champ (qui est vide) par l'utilisateur/le système.
u/sRequested(X)	Demande d'information sur ce champ (qui est rempli) par l'utilisateur/le système.
Relaxed(X)	La valeur de ce champ est obsolète et doit être ignorée.
DB(X)	Résultat d'une requête à la base de données.
Grounded(X)	L'utilisateur et le système sont d'accord sur la valeur de ce champ.
Denied(X)	L'utilisateur a marqué son désaccord avec la valeur de ce champ.

Tableau 3.11: Liste des états d'ancrage dans l'implémentation du prototype.

Acte d'ancrage	Signification
u/sInform	Ce champ est vide mais est évoqué par l'utilisateur/le système lors d'un acte de dialogue fournissant des informations.
u/sInform(X)	Une valeur pour ce champ fournie par l'utilisateur/le système.
u/sRequest	Demande d'information sur ce champ par l'utilisateur/le système.
u/sDeny	Refus de l'utilisateur/du système.
sOffer	Proposition de solution du système concernant ce champ.
sRelax	Proposition d'élimination d'une contrainte concernant ce champ par le système.
None	Pas d'acte de dialogue associé.

Tableau 3.12: Liste des actes d'ancrage dans l'implémentation du prototype.

(2) u/s : « u » pour l'utilisateur, « s » pour le système.

3.3.4 Passage aux espaces résumés

Le système continue le traitement d'un tour de parole en projetant la nouvelle croyance maître en une observation résumée. Et, pendant la génération du corpus exact de référence, en projetant l'état maître en un état résumé. Le rôle des projections dans les espaces résumés a été décrit en détail dans la partie 2.4. Nous présentons ici les résultats du *clustering* de graphes et de listes de graphes.

Les données utilisées ont été obtenues par simulation avec le système *HCSum* décrit *infra* avec simulation de l'environnement d'utilisateur et un niveau de bruit de $TER = 0,4$. On a utilisé 7650 graphes pour le *clustering* d'état ($M_s(s)$) et, du fait de la plus grande complexité de calcul de distance entre deux croyances, le *clustering* de croyance ($M_o(b)$) a été réalisé avec 912 listes *3-best* de graphes.

Les critères que nous utiliserons pour évaluer l'efficacité du *clustering* sont la mesure de Davies-Bouldin [Davies et Bouldin 1979] et les distances moyennes intra- et inter-clusters. La mesure de Davies-Bouldin est définie par :

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{i \neq j} \left\{ \frac{\Delta_i + \Delta_j}{S_{ij}} \right\} \quad (84)$$

où Δ_i est le diamètre du *cluster* i et S_{ij} est la distance entre les *clusters* i et j .

Les tableaux 3.13 et 3.14 ne montrent pas de claires différences entre les différents *clustering* effectués mais confirment que le *clustering* est correctement réalisé (distance inter-cluster moyenne nettement supérieure au diamètre moyen).

Nombre de clusters		Mesure de Davies-Bouldin	Distance inter-cluster moyenne	Diamètre moyen
Initiaux	Après convergence			
10	6	1,09	576,97	137,45
20	14	1,14	536,77	115,60
50	19	1,09	539,40	83,23

Tableau 3.13: Résultats du *clustering* de graphes pour la conception de la fonction résumée $M_s(s)$ définissant l'état résumé à partir de l'état maître.

Nombre de clusters		Mesure de Davies-Bouldin	Distance inter-cluster moyenne	Diamètre moyen
Initiaux	Après convergence			
10	5	1,08	608,62	214,93
20	10	1,03	612,90	207,18
50	27	1,22	589,69	160,09

Tableau 3.14: Résultats du *clustering* de listes *n-best* de graphes pour la conception de la fonction de projection résumée $M_o(b)$ définissant l'observation résumée à partir de la croyance maître.

3.3.5 Retour à l'espace maître

Pour préparer la sélection de l'action de retour dans l'espace maître, les actions maîtres potentielles sont générées à partir des actions résumées selon le mécanisme décrit en 2.2.1.

3.3.6 Couplage entre règles et POMDP

Le POMDP possède à présent l'information nécessaire à l'utilisation de sa stratégie au niveau résumé. Celle-ci est exprimé à travers la fonction Q qui permet d'affecter un score à chaque action résumée. Comme décrit dans la partie 2.2.2, différentes actions peuvent donc avoir le même score. Afin de désambigüiser les actions maîtres de même score et d'obtenir une action maître optimale, on utilise des règles prédéfinies. Dans notre prototype, les règles sont implémentées très simplement en utilisant une hiérarchie d'actions dont un extrait est donné dans la table 3.15. Cette hiérarchie associe à chaque action un score $Q^0(a)$ selon une fonction heuristique déterministe.

Par exemple, pour les actions maîtres correspondant à l'action résumé `constQ` (poser une question contrainte), on pourra considérer que lorsqu'une question doit être posée à propos d'un élément d'information, il est préférable de demander la ville (`AskCity`), puis des contraintes de confort (`AskConstraints`), puis des dates de réservations (`AskDates`). Il suffit pour cela de définir une hiérarchie assurant les inégalités suivantes :

$$Q^0(\text{AskCity}) > Q^0(\text{AskConstraints}) > Q^0(\text{AskDates}) \quad (85)$$

Les actions fortement liées à la résolution de la tâche ont été favorisées en leur associant un score élevé (par exemple, le fait de proposer des réponses, de terminer le dialogue ou de chercher à relaxer les contraintes données par l'utilisateur).

Modèle d'action maître (a)	Action résumée associée (\tilde{a})	Score $Q^0(a)$
G-close : « <i>Au revoir.</i> »	Closing	1006
G-3-offer : « <i>Je vous propose trois hôtels, l'hôtel ..., l'hôtel ... et l'hôtel ...</i> »	Offer	1004
G-2-offer : « <i>Je vous propose deux hôtels, l'hôtel ... et l'hôtel ...</i> »	Offer	1003
G-1-offer : « <i>Je vous propose l'hôtel ...</i> »	Offer	1002
G-0-relax : « <i>Aucun hôtel avec X. Je vous propose l'hôtel ... sans X</i> »	Relax	1001
G-0-offer : « <i>Désolé, aucun hôtel ne correspond à votre demande.</i> »	Offer	1000
AskCity : « <i>Quelle ville ?</i> »	ConstQ	3
AskConstraints : « <i>Quels équipements particuliers ?</i> »	ConstQ	2
AskDate : « <i>Quelles dates de réservation ?</i> »	ConstQ	1

Tableau 3.15: Exemple de hiérarchie des scores $Q^0(a)$ des actions maîtres.

3.4 Optimisation de stratégie POMDP par simulations

***Résumé** : Les stratégies du POMDP résumé sont entraînées par renforcement sur des corpus obtenus par simulation. Elles obtiennent de meilleures performances que le système sans POMDP. Le second suivi de croyance (dans les espaces résumés) permet de palier les imperfections du premier suivi de croyance (dans les espaces maîtres). Le résumé automatique par clustering donne des résultats comparables à ceux obtenus en utilisant un résumé manuel si le nombre de clusters est suffisant.*

L'étude des performances d'un système reposant sur le paradigme de POMDP à double suivi de croyance que nous proposons est essentielle pour valider l'intérêt de cette approche. C'est pourquoi nous présentons à présent l'influence de différents paramètres sur le comportement du système.

Après la présentation de la mise en œuvre des simulations et du procédé d'optimisation (partie 3.4.1), on présente les performances du système POMDP pour différents niveaux de bruit (partie 3.4.2). Puis, on étudie l'utilité de réaliser un suivi de croyance dans les espaces maîtres (partie 3.4.3) et dans les espaces résumés (partie 3.4.4). L'efficacité des projections résumées conçues automatiquement par *clustering* est présentée en partie 3.4.5. L'influence du couplage entre le système POMDP résumé et le système déterministe est présenté en partie 3.4.6. Les limites des résultats présentés sont analysées en partie 3.4.7.

3.4.1 Mise en œuvre des simulations

i Procédé itératif

Comme décrit en partie 2.5.3, les stratégies sont apprises par un processus itératif en utilisant l'utilisateur simulé UMLIB. Il fonctionne sur un modèle d'agenda et a été modifié pour générer des graphes de *frames* sémantiques.

L'algorithme itératif d'optimisation est illustré par la figure 3.6. La stratégie initiale est la stratégie de référence. Déterministe, elle ne repose que sur une hiérarchie de règles déterministes et n'utilise pas de méthode statistique (stratégie de référence sans POMDP). Elle permet de générer un premier corpus. Ce dernier est utilisé pour calculer une stratégie POMDP résumé optimale π (algorithme Perseus). Cette stratégie permet de générer d'autres dialogues qui sont ajoutés au corpus précédemment obtenu. On itère le processus en générant à chaque étape un nombre de dialogues croissant de façon exponentielle. Ce choix de générer un plus petit nombre de dialogues au début de l'optimisation permet de minimiser l'influence de la stratégie initiale sur le corpus final.

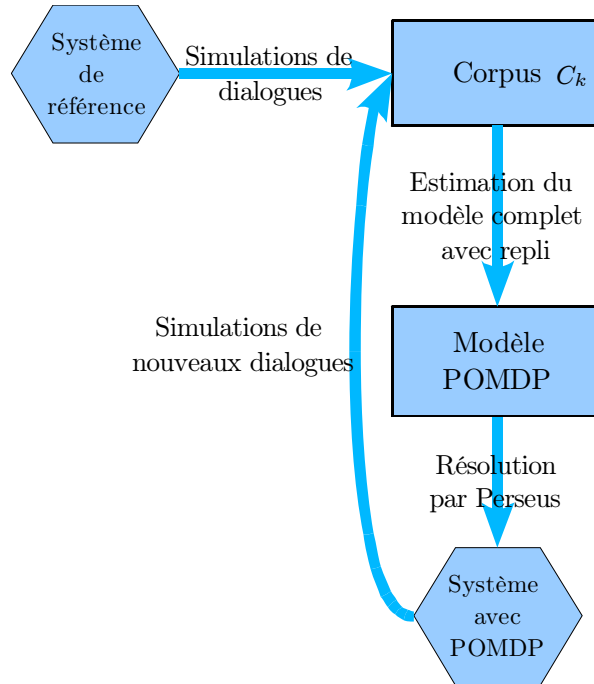


Figure 3.6: Application du procédé itératif d'optimisation de stratégie POMDP.

Itération (i)	$50 \times 1,5^i$	Nombre de dialogues générés pour l'itération i	Nombre total de dialogues générés
1	50	50	50
2	75	50	100
3	112	100	200
4	168	150	350
5	252	250	600
6	378	350	950
7	567	550	1500
8	850	850	2350
9	1275	1250	3600
10	1912	1900	5500

Tableau 3.16: Le nombre de dialogues générés à chaque itération suit une croissance géométrique de raison 1,5 de premier terme 50 (arrondi au multiple de 50 inférieur).

ii Niveau de bruit

Le niveau de bruit considéré est le taux d'erreur du tour de parole (TER¹). Dans le cadre de l'utilisation de n -best, c'est le taux d'erreur sur l'élément de la liste n -best ayant le

(1) TER : *Turn Error Rate*.

score de confiance le plus élevé. Le tableau 3.17 montre un exemple illustrant cette définition. Lors d'expériences avec des utilisateurs réels, le TER observé a été de 0,44. Les expériences par simulations ont été réalisées pour un TER variant de 0,1 à 0,6.

Observation	Status
[(<u>Nyons</u> ; 0,5) , (Lyon ; 0,1), (non ; 0,3)]	Correcte
[(Lyon ; 0,5), (<u>Nyons</u> ; 0,1), (non ; 0,3)]	Incorrecte

Tableau 3.17: Exemple de deux listes d'observations en 3-best. On suppose que le tour de dialogue exact est « Nyons », tandis que « Lyon » et « non » en sont des versions bruitées.

L'utilisateur simulé ([Schatzmann et al. 2007a; Schatzmann et al. 2007b] et voir aussi la partie 1.4), permet d'utiliser trois modèles d'erreur et de génération de scores n -best. Pour estimer la probabilité de confusion entre les différents actes de dialogue, le premier utilise un modèle uniforme. Le second utilise des matrices de confusion apprises sur un corpus. Le troisième, plus complexe, utilise la similarité acoustique (entre les phonèmes) de la représentation en mots des actes de dialogue. Les matrices de confusion étant apprises sur un corpus en langue anglaise, il aurait été nécessaire de se procurer un corpus adapté pour les adapter au français. Le corpus MEDIA pourra être utilisé en ce sens dans de futurs travaux. Dans un premier temps, la modélisation de l'utilisateur et du bruit sur son observation n'étant pas l'objet principal de cette thèse, nous avons choisi d'utiliser un modèle simple avec une probabilité uniforme sur les actes de dialogue.

iii Exploration

Dans un corpus réalisé avec un système utilisant une stratégie donnée, il est possible que certains types de dialogues ne soient jamais explorés. Pour obtenir un corpus suffisamment varié, il est nécessaire de permettre au système de s'engager dans des dialogues atypiques en ajoutant une composante aléatoire au système. Le taux d'exploration a été fixé à 5 % dans toutes les expériences présentées. Un tel taux est assez faible pour ne dégrader que faiblement les performances du système lors de l'apprentissage.

3.4.2 Entraînement du POMDP résumé

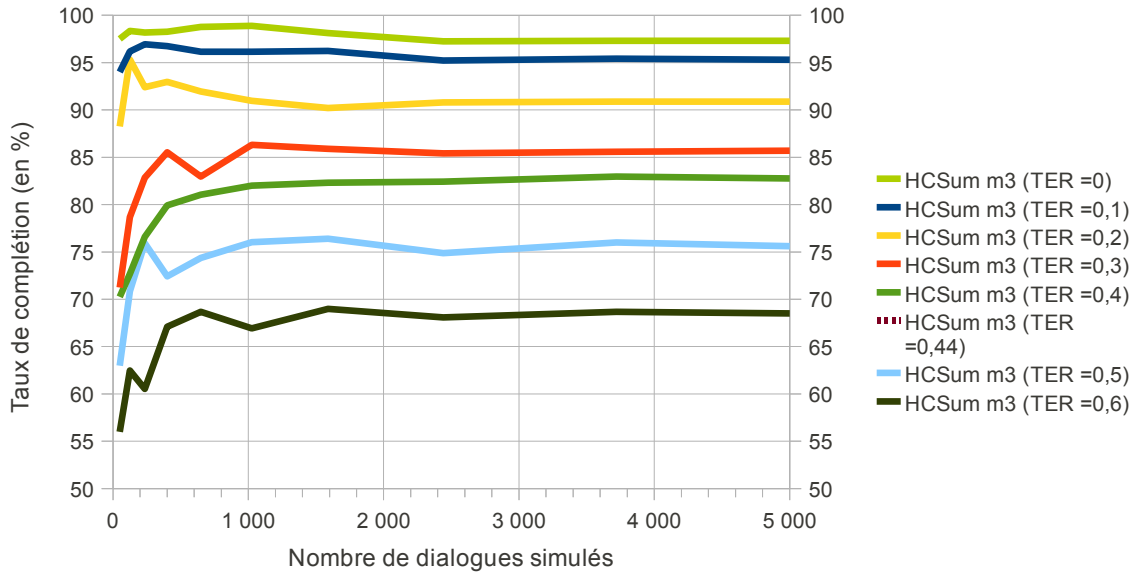


Figure 3.7: Taux de complétion du système HCSum durant l'apprentissage pour différents niveaux d'erreur (moyenne sur 10 expériences).

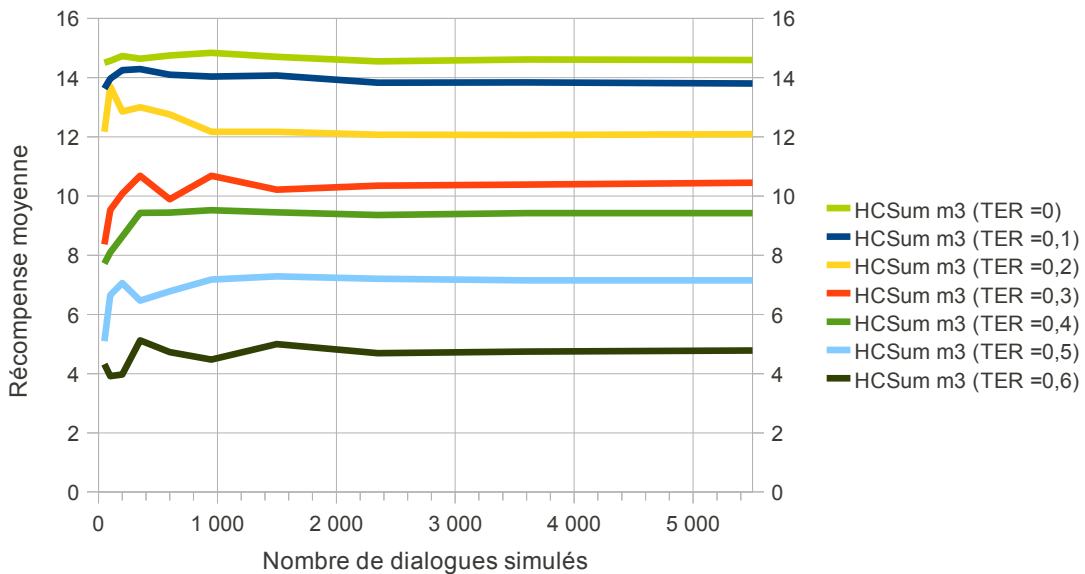


Figure 3.8: Récompense moyenne obtenue par le système HCSum durant l'apprentissage, pour différents niveaux d'erreur (la courbe est lissée en exécutant 10 fois l'expérience et en prenant la moyenne).

Les figures 3.7 et 3.8 montrent les performances du système (taux de complétion et récompense moyenne) au cours de l'entraînement, selon différents niveaux de bruit. Le système utilisé est ici *HCSum*, il comporte une stratégie POMDP et des fonctions résumées définies de façon *ad-hoc*. L'entraînement a été réalisé selon le procédé itératif décrit en 2.5 : «Optimisation de stratégie POMDP ».

On constate que les performances du système augmentent au cours des simulations par rapport au système de référence. L'optimisation de la stratégie à l'aide du POMDP résumée est donc effective et augmente sensiblement le taux de complétion de la tâche, ainsi que la récompense moyenne.

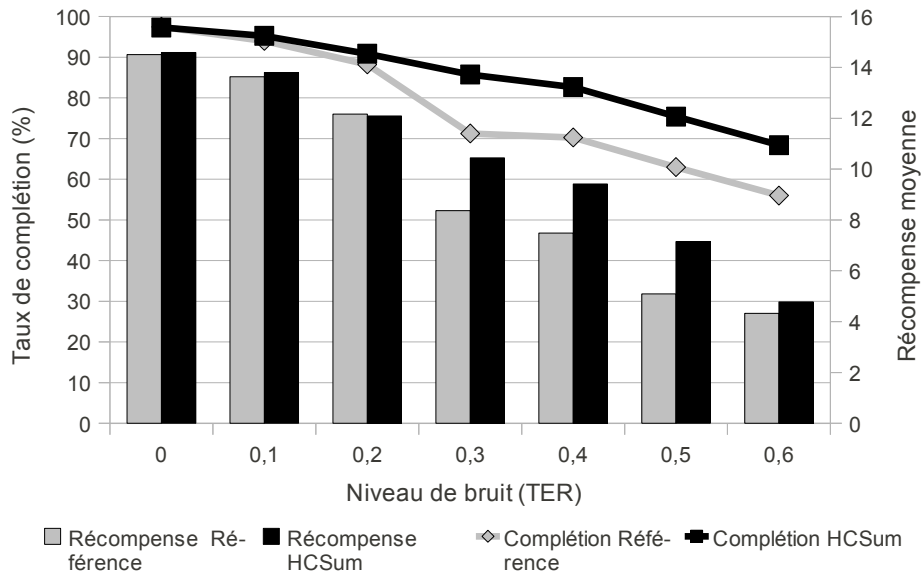


Figure 3.9: Taux de complétion et récompense moyenne du système POMDP entraîné comparé au système de référence selon le niveau de bruit.

Taux d'erreur	Récompense (HCSum)	Récompense (Référence)	Gain	Complétion (HCSum)	Complétion (Référence)	Gain
0,0	14,59	14,50	0,6%	97,3	97,5	-0,2%
0,1	13,80	13,63	1,2%	95,3	94,0	1,3%
0,2	12,09	12,17	-0,6%	90,9	88,3	3,0%
0,3	10,44	8,36	24,9%	85,8	71,3	20,4%
0,4	9,42	7,48	25,9%	82,7	70,3	17,7%
0,5	7,15	5,10	40,2%	75,4	63,0	19,7%
0,6	4,78	4,32	10,5%	68,4	56,0	22,2%

Tableau 3.18: Récompense moyenne du système POMDP entraîné comparé au système de référence selon le niveau de bruit.

La figure 3.9 montre l'évaluation d'une stratégie avec POMDP résumé obtenue à la fin de l'entraînement comparée à la stratégie de référence, pour différents niveaux de bruit (de 0,0 à 0,6).

On constate fort logiquement une dégradation des performances quand le bruit augmente. Un fort niveau de bruit (0,6) fait chuter les performances du système de référence : on passe d'un taux de complétion de 97,3 % à 68,4 %, la récompense moyenne chute également de 14,6 à 4,8.

En revanche, comparé au système de référence, le système POMDP est plus robuste face aux erreurs de compréhension. Par exemple, pour un niveau de bruit de 0,4, proche de celui de 0,44 observé lors des expériences avec des utilisateurs réels (voir partie 3.5 : « Tests utilisateurs »), l'utilisation de méthodes de renforcement permet de gagner +17,7 % relatifs de taux de complétion et +25,9 % relatifs de récompense moyenne.

Notons également que l'emploi de méthodes de renforcement par POMDP résumé n'a pas d'influence significative pour les faibles niveaux de bruit (entre 0,0 et 0,2), ce qui est cohérent avec l'hypothèse que l'emploi de modèles POMDP permet de rendre les systèmes plus robustes aux erreurs de reconnaissance et de compréhension

3.4.3 Suivi de croyance maître

Après avoir confirmé que le comportement d'un POMDP est celui qui est attendu, il est nécessaire de tester la pertinence de notre proposition de double suivi de croyance, dans l'espace maître et dans l'espace résumé. Cette partie présente l'étude de l'influence du suivi de croyance dans l'espace maître avec le système *HCSum* précédemment décrit.

Le suivi de croyance dans l'espace maître est réalisé en intégrant l'information contenue dans la liste de *n-best* en utilisant la formule de mise à jour (55). Cependant, dans l'implémentation pratique, un élagage du vecteur de croyance est réalisé en ne conservant que les m valeurs de croyance de poids le plus élevé. En fixant la valeur de m , on peut effectuer un suivi de croyance plus ou moins précis. Ainsi, $m = 1$ correspond à un système sans suivi de croyance dans l'espace maître. Le système standard *HCSum* précédemment décrit utilise la valeur $m = 3$.

Les figures 3.10 et 3.11 montrent les performances du système *HCSum* durant l'apprentissage avec trois niveaux ($m = 1, 3, 10$) de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).

Outre l'amélioration systématique par rapport au système de référence qui n'utilise pas d'apprentissage par renforcement, on constate que les systèmes implémentant un suivi de croyance dans l'espace maître ($m = 3, m = 10$) ont de meilleures performances que le système sans suivi de croyance ($m = 1$). Cela montre qu'il est effectivement utile de réaliser un suivi de croyance dans l'espace maître en plus d'une approche de POMDP résumé.

Le faible écart constaté entre les systèmes $m = 3$ et $m = 10$ montre cependant qu'il n'est pas nécessaire que ce suivi de croyance soit très précis. Un élagage peut être effectué, afin que seules quelques unes des meilleures hypothèses du vecteur de croyance soient prises en compte. L'enjeu est ici de trouver un compromis pour réduire la complexité des calculs par un élagage du nombre d'hypothèses tout en conservant un faisceau suffisamment large pour ne pas éliminer l'hypothèse correcte.

Notons de plus que même sans suivi de croyance dans l'espace maître ($m = 1$), le système optimisé améliore les performances.

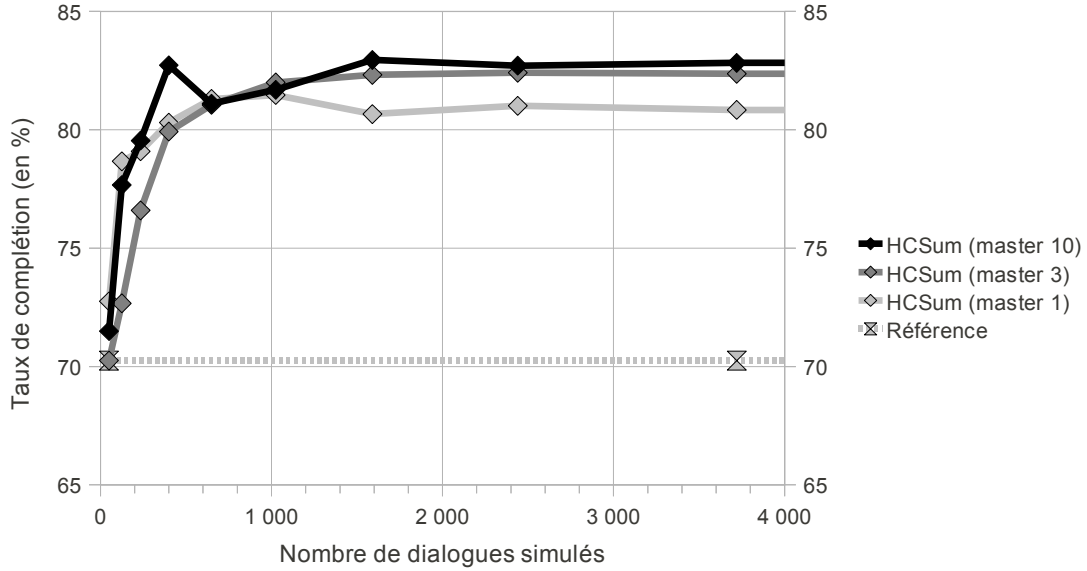


Figure 3.10: Taux de complétion du système HCSum durant l'apprentissage avec trois niveaux différents de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).

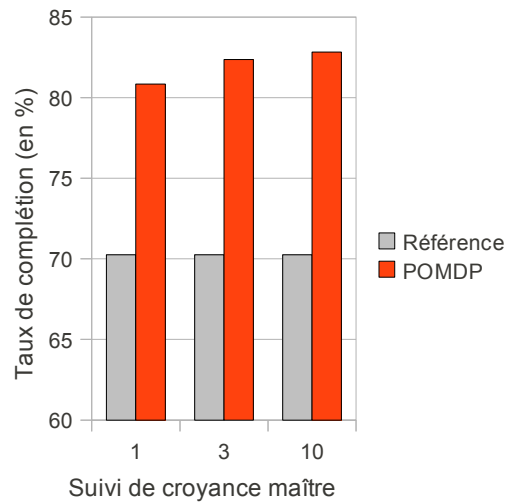


Figure 3.11: Taux de complétion du système HCSum avec trois niveaux différents de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).

3.4.4 Comparaison à un MDP

Toujours dans le but de tester la pertinence de notre proposition de double suivi de croyance, cette partie présente une comparaison entre le système avec double suivi de

croissance *HCSum-POMDP* et un système sans suivi de croyance dans l'espace résumé (MDP résumé).

Le système MDP résumé est issu du système *HCSum-POMDP* en faisant les simplifications suivantes sur le modèle de dépendances entre états et observations résumés :

- L'état résumé \tilde{s} est défini à partir de l'observation résumée : $\tilde{s} = \tilde{o}$. Notons que l'on utilise toujours la projection de la croyance maître pour l'observation résumée : $\tilde{o} = M_o(b)$.
- Le modèle d'observation est supposé parfait : $P(\tilde{o}|\tilde{s}) = 1 \iff \tilde{o} = \tilde{s}$.

Notons que ce système MDP résumé effectue le suivi de croyance dans l'espace maître exactement comme le système *HCSum-POMDP*. La différence entre les deux modèles MDP et POMDP se situe uniquement au niveau des espaces résumés. Le modèle de transitions entre états résumés est donc moins juste car il est appris sur des transitions entre observations.

Les figures 3.12 et 3.13 montrent les performances comparées des systèmes avec suivi de croyance dans l'espace résumé (POMDP résumé) et sans suivi de croyance dans l'espace résumé (MDP résumé).

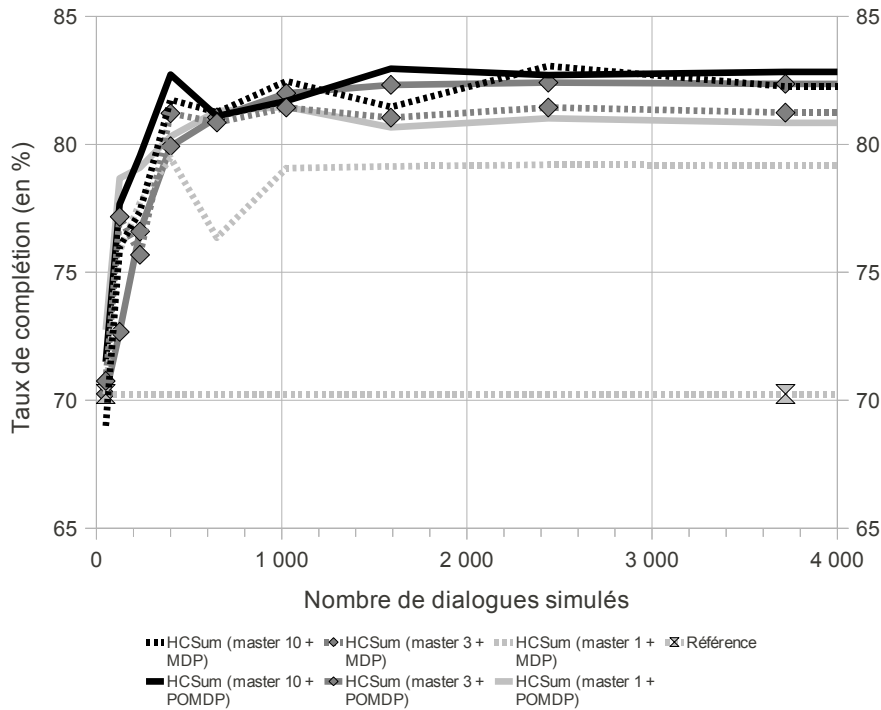


Figure 3.12: Taux de complétion du système *HCSum* durant l'apprentissage avec trois niveaux différents de suivi de croyance dans l'espace maître. Les systèmes intitulés *POMDP* et *MDP* sont respectivement avec ou sans suivi de croyance dans l'espace résumé (moyenne sur 10 expériences).

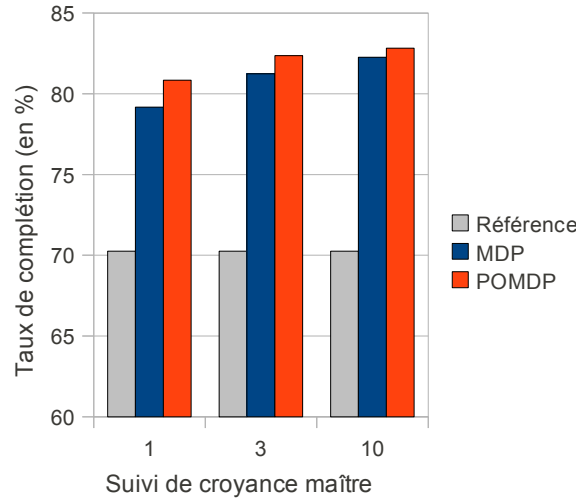


Figure 3.13: Taux de complétion après optimisation pour des systèmes MDP et POMDP selon la taille maximale de la distribution de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).

Niveau de suivi de croyance maître	MDP	POMDP	Référence
1	79,2% ($\sigma=2,8$)	80,8% ($\sigma=2,6$)	70,2%
3	81,2% ($\sigma=2,8$)	82,4% ($\sigma=1,7$)	70,2%
10	82,3% ($\sigma=2,0$)	82,8% ($\sigma=2,2$)	70,2%

Tableau 3.19: Les données de la figure 3.13 avec les écart-types.

On constate que le système avec double suivi de croyance (POMDP résumé) est systématiquement meilleur que le système avec suivi de croyance uniquement dans l'espace maître (MDP résumé). L'ajout d'un second suivi de croyance permet donc d'améliorer légèrement les performances du système. Les différences ne sont pas statistiquement significatives comme le montrent les écarts-types de l'ordre de 2% dans le tableau 3.19), mais elles sont cohérentes sur les trois niveaux de suivi de croyance maître.

D'autre part, les systèmes POMDP-résumé- $(m=1)$ et MDP-résumé- $(m=3)$ obtiennent des performances très similaires : 80,8% ($\sigma=2,6$) et 81,2% ($\sigma=2,8$). La différence n'est pas significative. Le suivi de croyance dans l'espace maître peut donc être remplacé par un suivi de croyance dans l'espace résumé. Cela peut s'expliquer par le fait que dans le système POMDP-résumé- $(m=1)$, l'observation résumée obtenue en résumant la croyance maître est très bruitée, l'ajout d'un deuxième suivi de croyance est donc utile. En revanche, lorsque le suivi de croyance est correctement réalisé dans l'espace maître, l'observation résumée \tilde{o} n'est pas très différente de l'état résumé exact \tilde{s} et le suivi de croyance résumé devient moins utile.

3.4.5 POMDP résumés par *clustering* automatique

On a présenté, en partie 2.4, une méthode de *clustering* pour appliquer le paradigme de projection dans un POMDP résumé en minimisant l'effort humain nécessaire pour concevoir un système de dialogue. Les résultats présentés ici comparent les performances du système *ClustSum* qui repose sur des fonctions résumées définies par *clustering* automatique avec le système *HCSum* dont les fonctions résumées ont été définies manuellement.

Trois niveaux de granularité ont été considérés pour le système *ClustSum* :

- *ClustSum-6/5* : 6 états résumés et 5 observations résumées. Ceci réduit drastiquement la taille du modèle POMDP résumé.
- *ClustSum-14/10* : 14 états résumés et 10 observations résumées.
- *ClustSum-27/19* : 27 états résumés et 19 observations résumées.

Le modèle POMDP du système *HCSum* contient quant à lui 13 états et 24 observations résumés.

	Récompense	Gain	Taux de complétion	Gain
<i>ClustSum-6/5</i>	8,59	14,8%	76,0	8,2%
<i>ClustSum-14/10</i>	9,35	25,0%	80,5	14,5%
<i>ClustSum-27/19</i>	9,37	25,3%	80,7	14,9%
<i>HCSum</i>	9,42	25,9%	82,7	17,7%
Référence	7,48	-	70,3	-

Tableau 3.20: Performances comparées des systèmes avec fonctions résumées ad hoc (*HCSum*) et automatiques (*ClustSum*). Les gains sont calculés relativement au système de référence (qui fonctionne sans apprentissage statistique).

Les résultats présentés dans le tableau 3.20 et détaillés dans les figures 3.14 et 3.15 montrent que les trois systèmes *ClustSum-14/10*, *ClustSum-27/19* et *HCSum* obtiennent des performances comparables (gains relatifs autour de 25% en récompense et autour de 15% en taux de complétion), tandis que le système *ClustSum-6/5* améliore beaucoup moins le taux de complétion ainsi que la récompense moyenne.

Ceci s'explique par le fait que lorsque le nombre de *clusters* est insuffisant (*ClustSum-6/5* utilise 6 *clusters* d'états résumés et 5 *clusters* d'observations résumées), la stratégie résumée est apprise sur un modèle POMDP manifestement trop approximatif. Certains états ou observations très différents sont confondus dans un même *cluster* et le système ne dispose pas de suffisamment d'information pour prendre la meilleure décision.

En raffinant la classification par un nombre de *clusters* plus élevé (*ClustSum-14/10*), les performances approchent celles du système *HCSum*. L'information extraite par les fonctions résumées contient presque autant d'information pertinente pour la prise de décision lorsqu'on utilise des fonctions résumées définies manuellement. Notons cependant que la projection automatique fournit cette information de façon plus concise : le nombre d'états résumés est presque identique (13 contre 14) et le nombre d'observations est réduit de moitié (10 contre 24).

Une augmentation supplémentaire du nombre de *clusters* (*ClustSum-27/19*) ne fournit pas d'amélioration. Il reste possible d'extraire une information plus pertinente à partir de

l'état maître ou de la croyance maître, puisque ceci est réalisé par le système *HCSum* à résumé *ad hoc*. Cependant, la méthode de *clustering* semble atteindre sa limite et on ne peut plus extraire d'information plus pertinente en le raffinant. Cela ne signifie cependant pas que d'autres méthodes de projections plus efficaces n'existent pas.

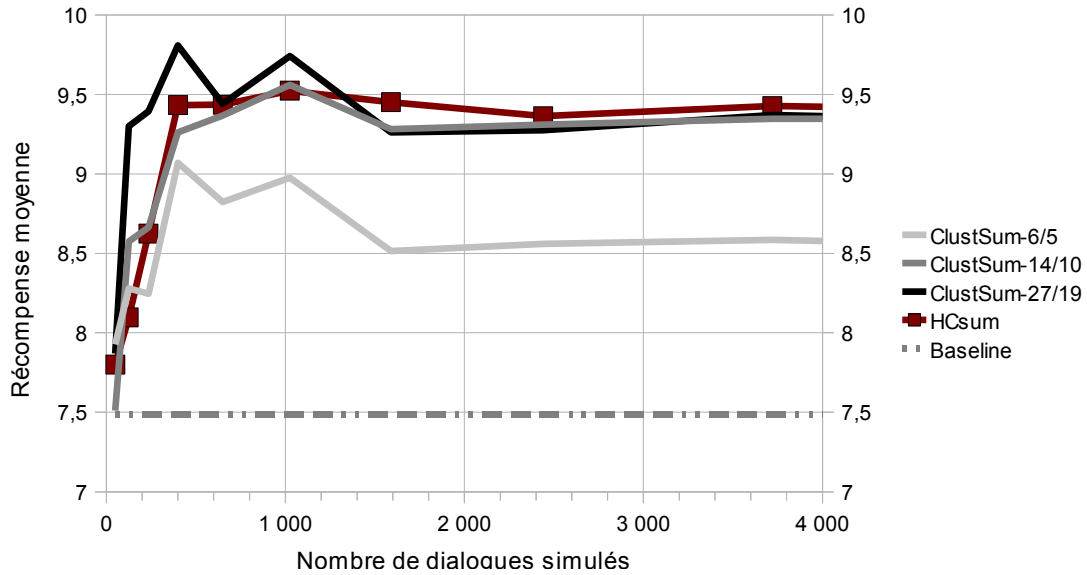


Figure 3.14: Récompense moyenne obtenue lors de l'entraînement par le système avec clustering automatique *ClustSum* pour différents nombres de clusters.

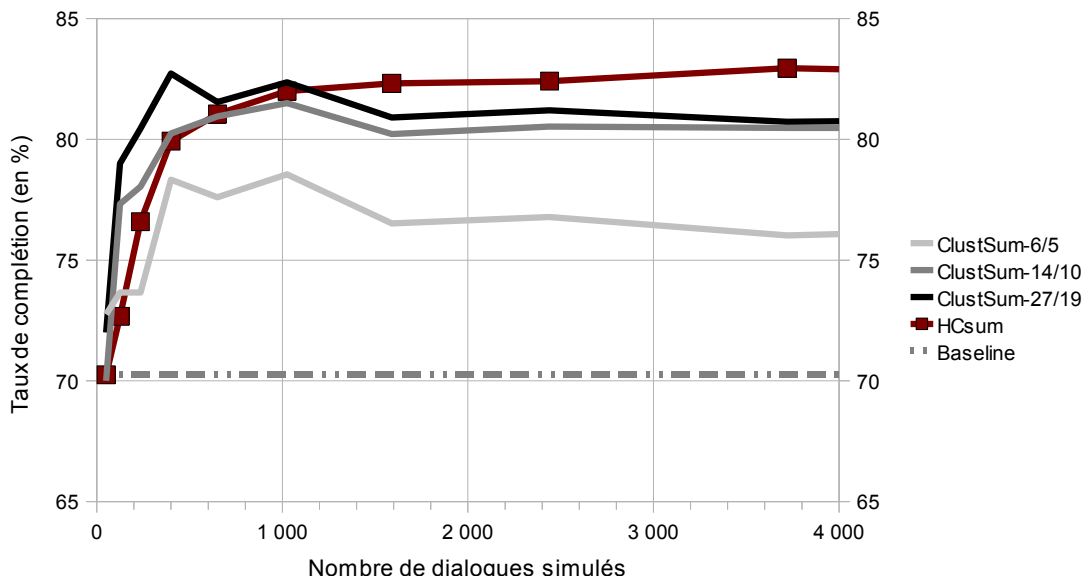


Figure 3.15: Taux de complétion lors de l'entraînement par le système avec clustering automatique *ClustSum* pour différents nombres de clusters.

3.4.6 Couplage de systèmes maître-résumé

Afin d'étudier l'influence du mécanisme de couplage entre les scores du système POMDP résumé et la hiérarchie d'actions définie de façon *ad hoc*, on a observé l'apprentissage de la stratégie avec différentes valeurs du coefficient de couplage λ de l'équation (45) rappelée ici : $score(\tilde{b}, b, a) = \lambda Q^\pi(\tilde{b}, a) + (1 - \lambda)Q^0(b, a)$ où Q^π correspond à la stratégie POMDP et Q^0 à la hiérarchie déterministe utilisée dans le système de référence.

	λ	Taux de complétion	Gain
<i>HCSum</i>	1	75,5	7%
<i>HCSum</i>	0,9	82,9	18%
<i>HCSum</i>	0,5	81,1	15%
Référence	0	70,3	0%

Tableau 3.21: Taux de complétion du système *HCSum* pour différents niveaux de couplage par le coefficient λ entre les deux stratégies maître Q^0 et résumée Q^π . Niveau de bruit 0,4.

On constate que le coefficient de couplage permet de contrôler l'influence du POMDP. S'il est trop important, le POMDP laissé trop libre a un comportement erratique lors de l'apprentissage. S'il est insuffisant, le système ne prend pas suffisamment en compte les informations statistiques issues de l'apprentissage.

3.4.7 Limites

Les courbes précédentes montrent les tendances moyennes en lissant les résultats grâce à 10 répétitions d'expériences. Cependant, lorsqu'on réalise une seule optimisation, les résultats sont très bruités. La figure 3.16 montre les performances sans lissage, en réalisant deux fois la même expérience.

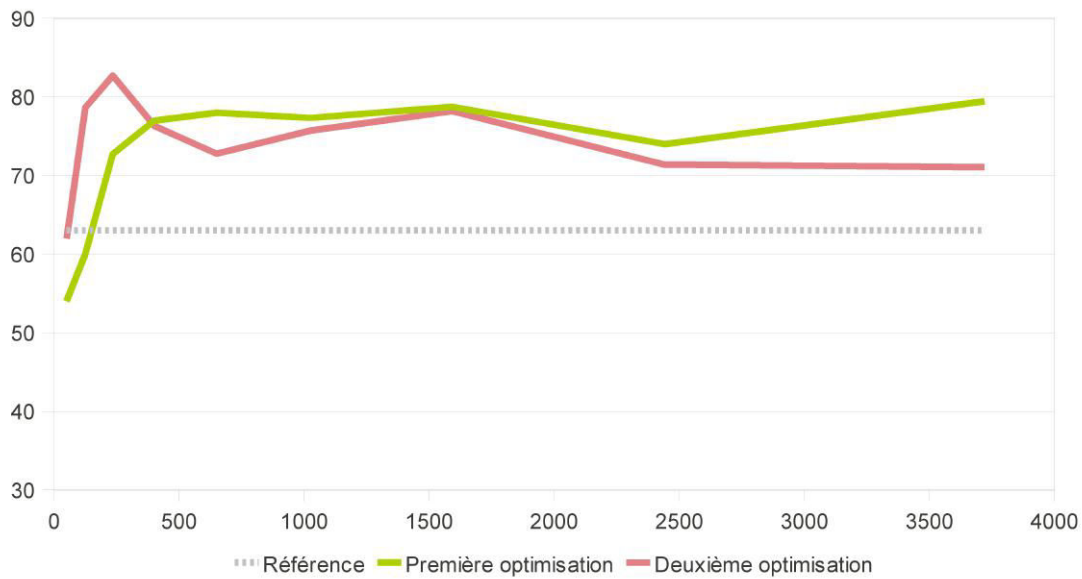


Figure 3.16: Taux de complétion obtenu lors de deux optimisations avec les mêmes paramètres. Niveau de bruit 0,5.

On constate que la convergence ne semble pas terminée et que le taux de complétion finale est difficile à estimer : 71% ou 80% ? Le gain (absolu) du système est-il de 9% ou de 18% ? Lorsqu'on cherche à comparer des systèmes entre eux, cette grande variabilité empêche de mettre en évidence les tendances profondes des systèmes.

Afin de mettre en évidence la tendance moyenne lors de l'apprentissage, nous avons répété les expériences 10 fois. La moyenne a ensuite été effectuée sur 8 valeurs seulement, en supprimant les deux valeurs extrêmes (minimum et maximum). Cet estimateur, entre une véritable moyenne et une médiane, offre une meilleure robustesse. La figure 3.17 montre les résultats de 10 répétitions de l'expérience.

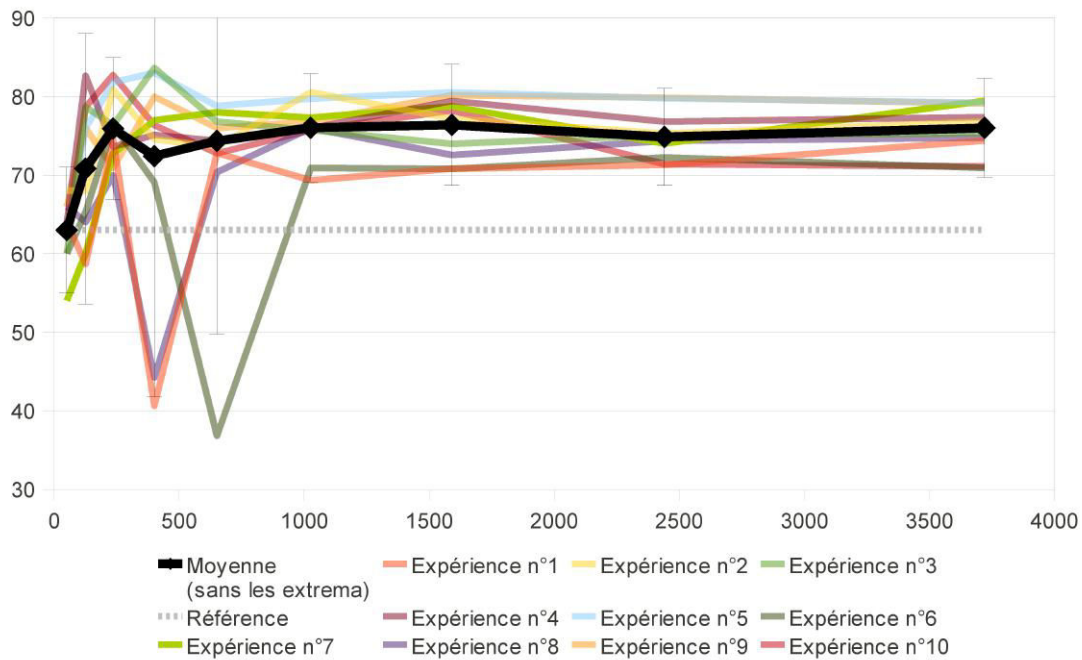


Figure 3.17: Performances obtenues durant l'optimisation lors de chacune des 10 expériences (en couleur). La courbe moyenne (en noir) est la moyenne des dix courbes en supprimant les minimum et maximum (moyenne sur 8 valeurs). Niveau de bruit 0,5.

L'utilisation de la moyenne permet de comparer différentes méthodes de conception de systèmes. On a pu observer, par exemple, les effets des deux suivis de croyance dans les espaces maître et résumé. L'analyse d'une seule expérience, présentant une trop grande variabilité, n'aurait pas été suffisante pour comparer les tendances moyennes de différents systèmes.

Dans l'optique de réaliser un véritable système de dialogue, il est nécessaire de répéter le processus d'optimisation un grand nombre de fois et de sélectionner la stratégie présentant les meilleures performances. Bien que cela ne soit pas très contraignant si l'on dispose d'un environnement de simulation, cette absence de contrôle du processus d'optimisation montre que de plus amples recherches sont nécessaires pour comprendre plus complètement les effets des mécanismes mis en œuvre.

En revanche, on peut chercher à appliquer le paradigme proposé en générant les corpus de dialogues par une interaction avec un véritable utilisateur. Dans ce cas, il est possible que, lors d'une itération, la stratégie du système soit temporairement très mauvaise. Ce risque, non nul, n'est pas toujours acceptable dans un cadre industriel [Pieraccini et Huerta 2005].

3.5 Tests utilisateurs

Résumé : Dans un contexte de dialogues réalisés avec de véritables utilisateurs, les systèmes POMDP obtiennent de meilleures performances que le système initial sur les critères objectifs. En revanche, d'après l'utilisateur (critères subjectifs), ils dégradent les performances du système initial.

Le prototype décrit ci-dessus (partie 3.3.1) a permis de réaliser des expériences avec des utilisateurs réels.

Les systèmes comparés sont les suivants :

- un système à base de règles logiques *ad hoc* fondées sur la hiérarchie d'actions décrites plus haut (sans aucun modèle statistique dans le gestionnaire de dialogue).
- le système précédent couplé (le mécanisme de couplage est détaillé dans la partie 2.2) à un POMDP optimisé par renforcement sur un corpus généré par simulation selon les méthodes décrites dans la partie 3.4.

Les coûts associés à de tels tests ne nous ont pas permis d'évaluer avec de véritables utilisateurs toutes les stratégies et configurations de système décrites dans la partie 3.4.

Cette section débute par la mise en œuvre des expériences à travers la sélection des tâches, des systèmes et des sujets (3.5.1). Puis, on présente les résultats selon les critères d'analyse retenus, objectifs (3.5.2) et subjectifs (3.5.3). Une synthèse et une analyse des limites de ces résultats sont présentées ensuite (3.5.4).

3.5.1 Mise en œuvre

Sujets

Les 25 sujets « volontaires » ont été recrutés parmi les chercheurs et les étudiants du Centre d'Enseignement et de Recherche en Informatique d'Avignon (CERI). Une partie d'entre eux sont spécialistes en traitement automatique de la parole, mais aucun en systèmes de dialogue, la notion d'expertise de l'utilisateur ne nous a donc pas semblé significative.

Systèmes évalués

Les deux systèmes évalués sont le système de référence et le système POMDP.

Le premier (référence) n'utilise pas de stratégie apprise par renforcement. Il a été complètement conçu de manière experte, pour répondre à la tâche d'information touris-

tique. Il utilise des règles de création d'actions maîtres ainsi qu'une hiérarchie fixant les scores de ces actions maîtres.

Le second (POMDP) ajoute une stratégie de POMDP résumé qui est couplée à la hiérarchie de règles par les méthodes décrites *supra*. Il s'agit du système noté *HCSum*, pour HandCrafted-Summary POMDP.

Protocole

Les deux tâches sur lesquelles les systèmes sont testés sont décrites dans la partie 3.1: « Méthodologie d'évaluation ». Le protocole d'expériences consiste à choisir une répartition des configurations possibles système-tâche-utilisateur. On a choisi un plan d'expérience orthogonal, où chacun des utilisateurs est confronté à chacun des deux systèmes pour les deux tâches données, soit 4 dialogues par utilisateur. L'ordre dans lequel les 4 dialogues sont effectués est également alterné pour réduire l'effet du gain d'expertise de l'utilisateur au cours de l'expérience. Pour chaque dialogue, un scénario aléatoire a été généré tel que décrit dans partie 3.1.

Déroulement des expériences

Après avoir défini, pour chaque utilisateur, les scénarios ainsi que les systèmes correspondants à tester, les expériences ont été effectuées par un expérimentateur unique, avec dans la mesure du possible les mêmes conditions d'enregistrement (même salle, fenêtres fermées, même réglage de microphone, même ordinateur, même usage du CPU, mêmes applications ouvertes, etc.). Les instructions aux expérimentateurs sont présentées en annexe B.1.

Un questionnaire est rempli par chaque utilisateur après chaque dialogue et à la fin d'une session de quatre dialogues. Ce questionnaire a été réalisé sous la forme d'un formulaire Web avec traitement des réponses en PHP et sauvegardée au format YAML. Comparée à un formulaire papier, cette démarche permet d'assurer une meilleure traçabilité des expériences et de traiter ensuite facilement les données recueillies.

C'est dans ce formulaire que les testeurs donnent leur consentement quant à l'utilisation en tant que corpus de recherche des données audio anonymisées obtenues lors de la réalisation des expériences. Le formulaire est présenté en annexe B.2.

Le corpus ainsi réalisé comporte donc 100 dialogues (4 x 25). La longueur moyenne d'un dialogue est de 6,95 tours de dialogue. Le taux moyen de complétion de la tâche est de 45 % et la récompense moyenne est de 2,98.

Évaluation

Les critères d'évaluation sont décrits dans la partie 3.1 : « Méthodologie d'évaluation ». Nous présentons ici les résultats pour chacun de ces critères.

3.5.2 Évaluation objective

3.5.2.1 Récompense

Les figures 3.18 et 3.19 présentent les récompenses obtenues en fin de dialogue. Il s'agit de la récompense immédiate obtenue pour chaque action, cumulée sur tout le dialogue telle que décrite dans la partie 3.2.3 : « Évaluation ».

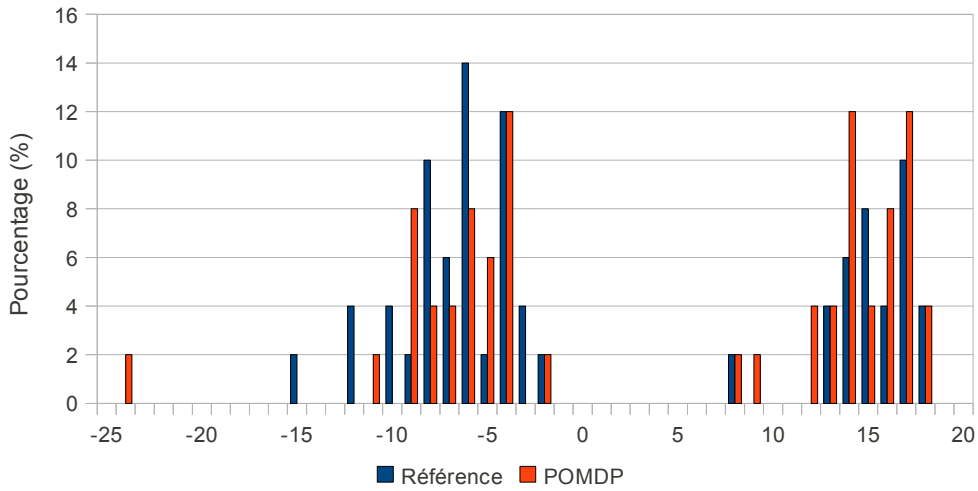


Figure 3.18: Récompense cumulée obtenue en fin de dialogue.

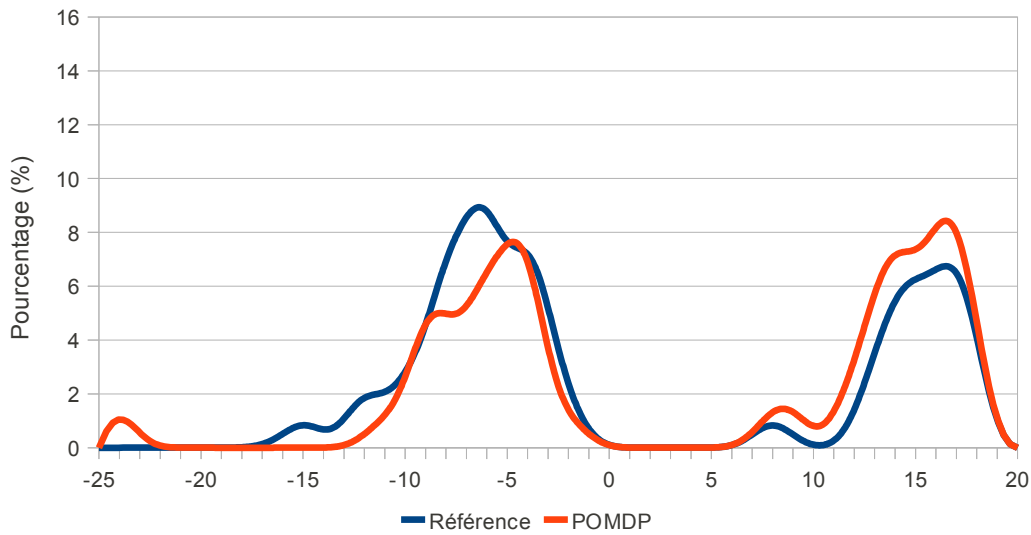


Figure 3.19: Récompense cumulée en fin de dialogue, avec lissage.

Le groupe de dialogues situés autour de la valeur de récompense 17 correspond à des dialogues réussis. Le deuxième groupe situé autour de la valeur -7 correspond à des dialogues se terminant par un échec.

On peut remarquer que le système POMDP (en rouge) obtient un peu plus souvent des dialogues à forte récompense et moins souvent des dialogues à faible récompense. Ce qui se traduit par une meilleure récompense moyenne : 4,34 (POMDP) contre 1,62 (Référence).

3.5.2.2 Complétion de la tâche

Le taux de complétion objectif confirme les observations issues de l'analyse de la récompense. Le système POMDP obtient un taux de complétion clairement plus élevé que le système de référence : 52 % (POMDP) contre 38 % (Référence).

3.5.2.3 Longueur du dialogue

Les figures 3.20 et 3.21 représentent les durées des dialogues. On ne constate pas de différence flagrante entre les deux systèmes. Les moyennes montrent que les dialogues avec le système POMDP sont légèrement plus longs : 7,18 (POMDP) contre 6,72 (référence).

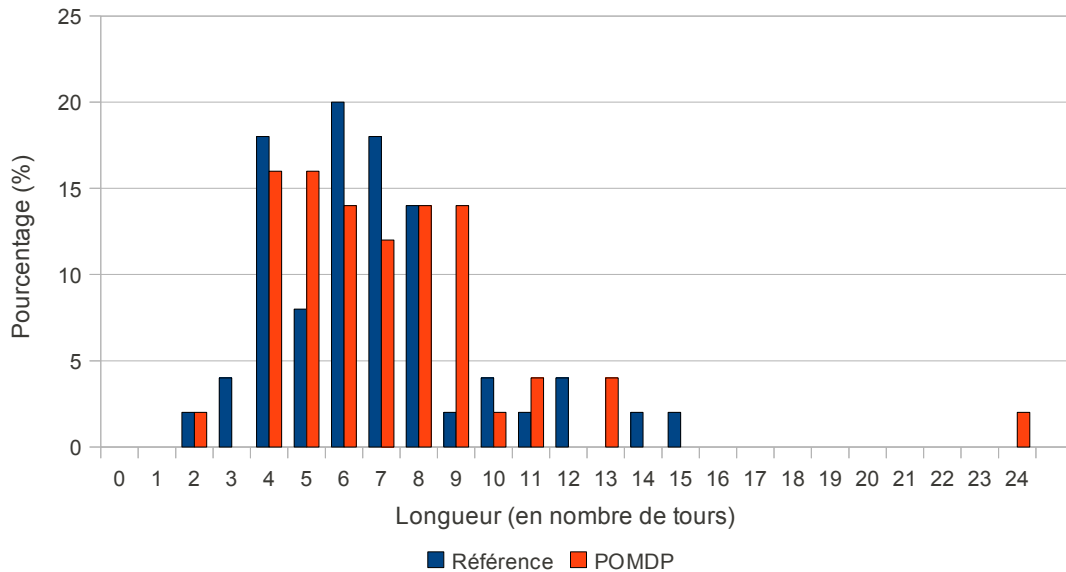


Figure 3.20: Répartition de la durée des dialogues.

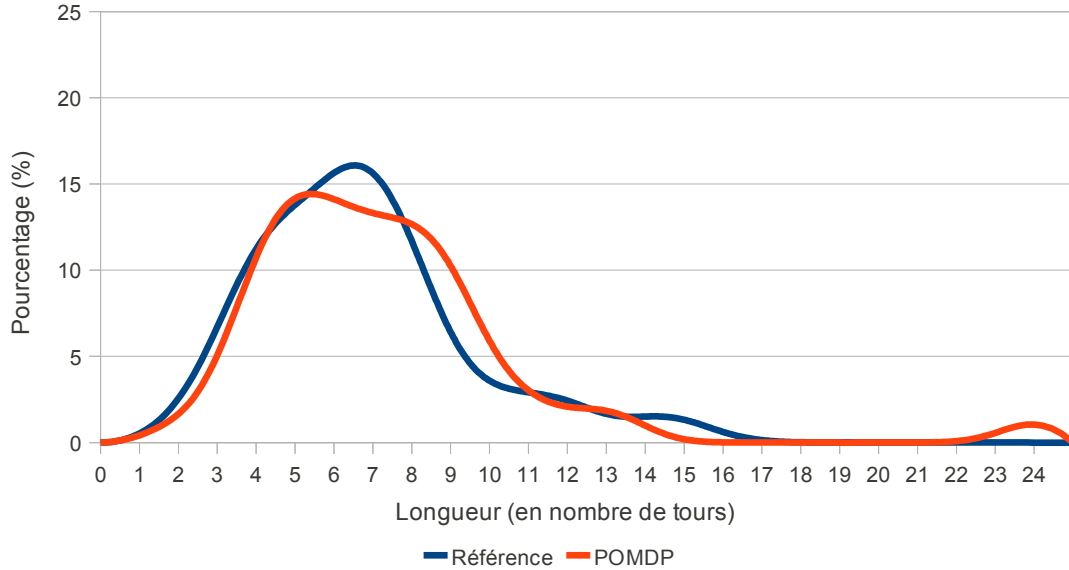


Figure 3.21: Répartition de la durée des dialogues, avec lissage.

3.5.2.4 Synthèse

En termes de critères objectifs, le système POMDP obtient de meilleures performances que le système de référence.

	Longueur (en nombre de tour)	Taux de complétion	Récompense
Référence	6,72	38%	1,62
POMDP	7,18	52%	4,34
Différence relative	+6,80%	+36,8%	+168%

Tableau 3.22: Synthèse des résultats de l'évaluation objective lors des tests avec des utilisateurs humains.

3.5.3 Évaluation subjective

Les critères d'évaluation subjective sont dits subjectifs car ce sont des critères du point de vue de l'utilisateur (voir partie 3.1).

Le taux de succès étant le principal indicateur que nous voulons étudier, nous avons défini différentes façons de l'estimer afin de permettre de recouper les résultats. On dispose donc d'un critère quantitatif (une note sur 10) et de deux critères qualitatifs (Q2 et Q3).

Les questions 4, 5 et 6 du questionnaire visent à évaluer d'autres aspects du système de dialogue que la stratégie du gestionnaire. Elles ont été incluses pour permettre une analyse contrastive *a posteriori* avec un système dans lequel on aurait modifié le module de compréhension sémantique par exemple, ou sur une autre tâche.

Dans cette partie, on passe en revue tous les critères subjectifs (parties 3.5.3.1 à 3.5.3.6) avant de proposer une synthèse (partie 3.5.3.7).

3.5.3.1 Taux de complétion par le questionnaire

Le taux de succès subjectif est évalué à travers une série de trois questions. Il y a deux types de dialogues réussis :

- Le système trouve une réponse correspondant aux attentes de l'utilisateur.
- Aucune réponse ne correspond aux attentes de l'utilisateur et le système arrive à en informer l'utilisateur.

La première question sert simplement à distinguer ces deux cas :

Q1 : « <i>A votre avis, le scénario donné était-il réalisable ? (était-il possible au système de répondre à votre demande d'après sa base de données ?)</i> »	Oui	Non	Je ne sais pas
---	-----	-----	----------------

On considère évidemment qu'un dialogue fournissant une réponse correspondant aux attentes initiales de l'utilisateur est correct. C'est l'objet de la deuxième question :

Q2 : « <i>Si oui, le système a-t-il réussi à vous fournir ce que vous lui demandiez ?</i> »	Oui	Plutôt oui	Plutôt non	Non
---	-----	------------	------------	-----

Mais on considère aussi comme un dialogue réussi le cas où la demande de l'utilisateur est impossible à satisfaire et où le système réussit à en informer l'utilisateur :

Q3 : « Si non, le système a-t-il réussi à vous communiquer que votre demande était impossible à satisfaire ? »	Oui	Plutôt oui	Plutôt non	Non
--	-----	------------	------------	-----

Les figures 3.22 et 3.23 présentent les résultats des réponses aux deux questions Q2 et Q3 pour les deux systèmes comparés.

En utilisant une échelle de Lickert afin d'affecter un score compris entre 0 et 3 à chaque réponse (oui = 3, plutôt oui = 2, plutôt non = 1, non = 0), les moyennes obtenues pour chaque question sont présentées dans le tableau 3.23.

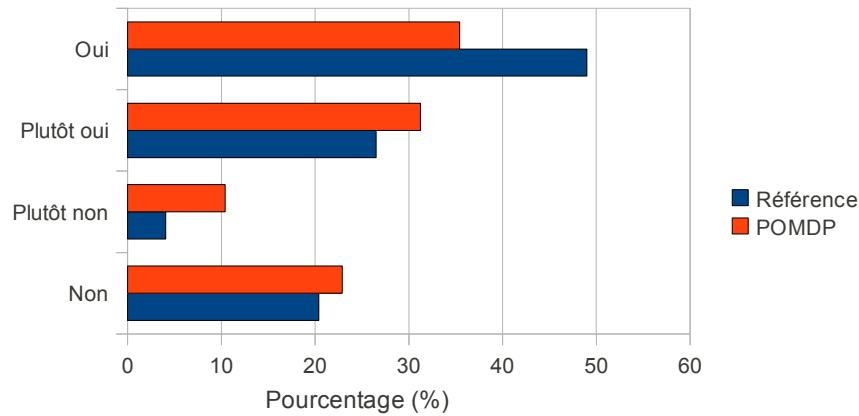


Figure 3.22: Proportions des réponses à la question « Si oui, le système a-t-il réussi à vous fournir ce que vous lui demandiez ? »

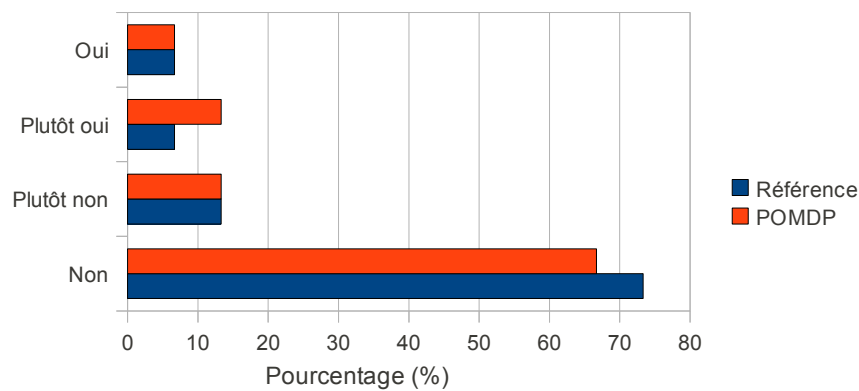


Figure 3.23: Proportions des réponses à la question « Si non, le système a-t-il réussi à vous communiquer que votre demande était impossible à satisfaire ? »

	Q2	Q3
Référence	2,04	0,4
POMDP	1,8	0,65

Tableau 3.23: Moyennes obtenues pour les questions Q2 et Q3
(valeurs comprises entre 0 et 3)

On constate que le système entraîné par renforcement obtient de moins bonnes performances que le système de référence en termes de complétion subjective de la tâche lorsqu'une réponse est possible (Q2), mais qu'il est meilleur pour informer l'utilisateur de l'impossibilité de répondre à sa demande (Q3).

3.5.3.2 Taux de succès par une note

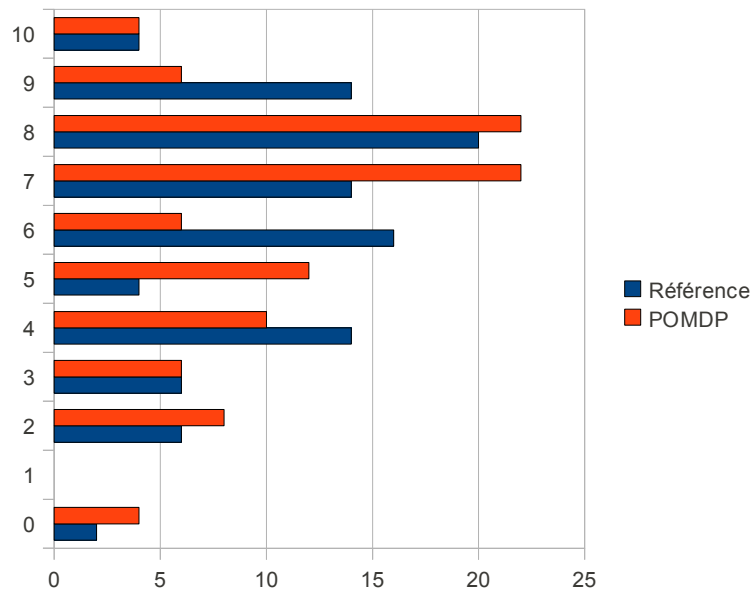


Figure 3.24: Succès du dialogue, évalué par une note sur 10
(critère subjectif quantitatif).

Les scores moyens obtenus sont de 6,26 (référence) et 5,92 (POMDP). Il conduit à la conclusion que les utilisateurs sont plus souvent satisfaits de leur interaction avec le système de référence qu'avec le système POMDP entraîné par renforcement.

3.5.3.3 Taux de succès par clic final

Le taux de succès subjectif est aussi évalué par le clic final de utilisateur sur l'un des deux boutons « Raccrocher (Réussi) » ou « Raccrocher (Abandon) ». Pour cette variable binaire, on associe un succès à la valeur 1 et l'échec à la valeur 0. Les scores sont alors de 0,66 (référence) et 0,6 (POMDP).

3.5.3.4 Compréhension

La question permettant de juger du bon fonctionnement du système de compréhension est la suivante :

Q4 : « Les réponses du système étaient-elles plutôt appropriées à la situation ou bien hors de propos ? »

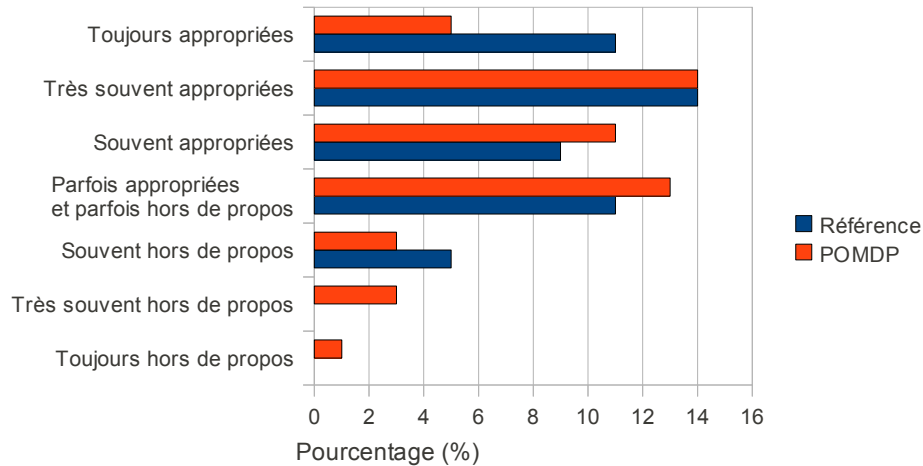


Figure 3.25: Évaluation de la compréhension du système (question Q4).

Les scores moyens sur une échelle de 0 (toujours hors de propos) à 6 (toujours appropriées) sont de 4,3 (référence) et 3,84 (POMDP).

3.5.3.5 Initiative partagée

Dans le questionnaire, la question permettant d'évaluer le partage de l'initiative lors du dialogue est la suivante :

Q5 : « Qui a principalement guidé la conversation ? »

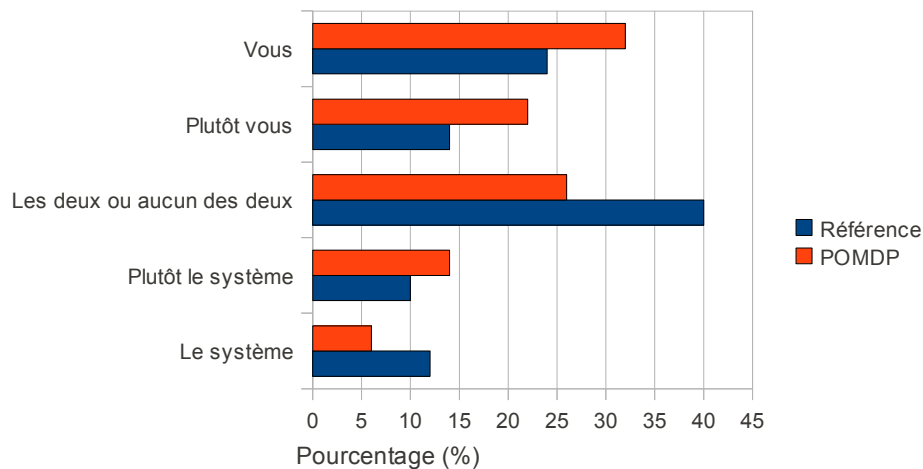


Figure 3.26: Réponse à la question visant à évaluer le partage d'initiative lors du dialogue pour les deux systèmes comparés.

On constate que le système POMDP tend à laisser plus d’initiative à l’utilisateur. En utilisant une échelle de 0 à 4, il obtient une moyenne de 2,6, à comparer à 2,28 pour le système de référence.

	Q5
Référence	2,28
POMDP	2,6

Tableau 3.24: Moyennes obtenues pour la question Q5

3.5.3.6 Facilité d’utilisation

Dans le questionnaire, la question permettant d’évaluer la facilité d’utilisation du système lors du dialogue est la suivante :

Q6 : « *Était-il facile et intuitif de communiquer avec le système ?* »

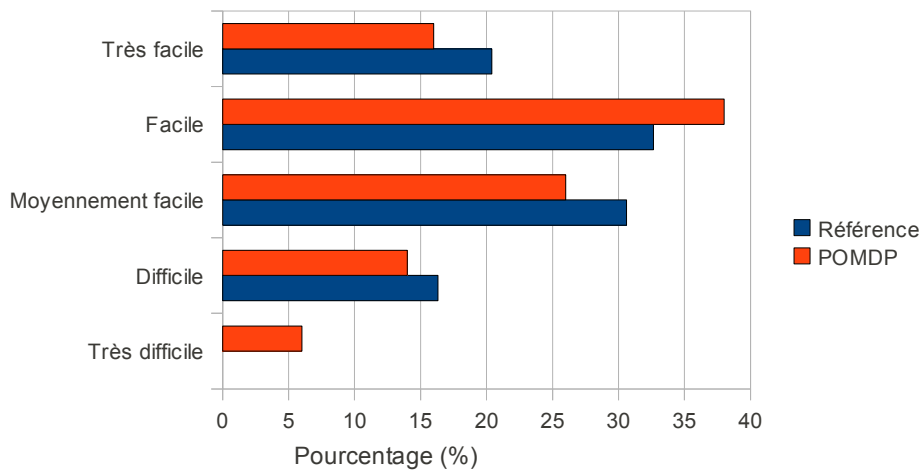


Figure 3.27: Réponse à la question visant à évaluer la facilité d’utilisation du système pour les deux systèmes comparés.

Les moyennes sur une échelle de 0 (Très difficile) à 4 (Très facile) montrent également que le système de référence apparaît comme plus facile d’utilisation avec une moyenne de 2,573 contre 2,44 pour le système POMDP.

3.5.3.7 Synthèse

Comme indiqué dans le tableau suivant, le système de référence est légèrement meilleur que le système POMDP en terme de performance globale, selon la plupart des critères subjectifs. Notons que les différences entre les deux systèmes apparaissent comme minimales, lorsque l'on compare l'écart constaté entre les deux systèmes avec l'écart maximum théoriquement possible. Par exemple, pour le critère quantitatif de la note attribuée par l'utilisateur, l'écart entre les systèmes est de 0,34, contre un maximum est de 10. Pour le critère **Q2**, on obtient 0,24 à comparer avec un maximum de 3.

	Note utilisateur	Q2 : Réussite	Q3 : Réussite	Q4 : Compréhension	Q5 : Initiative	Q6 : Facilité d'utilisation	Clic utilisateur
Référence	6,26	2,04	0,41	4,30	2,28	2,57	66%
POMDP	5,92	1,80	0,65	3,84	2,60	2,44	60%
Différence relative	-5,4%	-11,8%	+58,50%	-10,7%	+14,0%	-5,1%	-9,1%

Tableau 3.25: Synthèse des résultats de l'évaluation par des utilisateurs humains.

3.5.4 Conclusion et limites

En considérant les tables 3.22 et 3.25 synthétisant les résultats des critères objectifs et subjectifs, on observe des indicateurs montrant que le système POMDP réussit à optimiser le taux de succès sur lequel il a été optimisé. En revanche les indicateurs de succès subjectifs, perçus par l'utilisateur, montrent généralement une dégradation des performances.

Le système POMDP améliore les performances selon les critères objectifs. Cette constatation valide donc l'approche proposée en ce qui concerne l'apprentissage par renforcement.

Par contre, on constate que ces critères objectifs sont en désaccord avec les critères subjectifs fournis par l'utilisateur. Il est donc nécessaire d'approfondir le mécanisme d'évaluation.

Biais de l'évaluation subjective

Une explication du phénomène réside dans un certain biais lors de l'évaluation subjective. En effet, il arrive fréquemment que l'utilisateur soit persuadé que le système lui a permis de résoudre la tâche alors qu'il n'en est rien. Ceci provient d'une imperfection du mécanisme de confirmation comme l'illustre l'exemple 3.26.

<p>Scénario : Vous avez prévu un séjour à Lyon, vous voulez donc réserver un hôtel du 28 juin au 2 juillet. Vous êtes prêt à accepter n'importe quel hôtel mais vous tenez à avoir un hôtel qui accepte les animaux.</p>	
<p>USR 1 : <i>Je voudrais aller avec mon doberman à Lyon du 28 juin au 2 juillet.</i> ASR 1 : je voudrais aller avec mon euh à Lyon du 28 juin au 2 juillet</p>	
<p>SYS 2 : <i>Je vous propose un hôtel à Lyon, le Grand Ritz, d'accord ?</i></p>	
<p>USR 3 : <i>Oui, à Lyon, d'accord, et je viendrai avec mon doberman.</i> ASR 3 : oui à Lyon d'accord et je viendrai avec euh euh</p>	
<p>SYS 4 : <i>Confirmez-vous votre réservation pour l'hôtel du Grand Ritz à Lyon ?</i></p>	
<p>USR 5 : <i>Oui</i> ASR 5 : Oui</p>	
<p>Critère Subjectif : Succès</p>	<p>Critère Objectif : Échec</p>
<p>L'utilisateur est pleinement satisfait. Il a répété deux fois qu'il avait un chien, et pense que son interlocuteur a répondu en tenant compte de cette information.</p>	<p>Dans la base de données, le Grand Ritz à Lyon n'accepte pas les animaux. Il ne fait pas partie des propositions correspondant au scénario donné.</p>

Tableau 3.26: Exemple de manque de confirmation et défaut de compréhension conduisant à un désaccord entre critères subjectif et objectif.

Dans cet exemple, le mot « doberman » n'a pas été compris car il n'a pas été appris lors de l'entraînement des modules de reconnaissance et de compréhension (ASR ou SLU). Bien que la confirmation implicite des informations données par l'utilisateur soit implémentée et joue pleinement son rôle (*feed-back*), le système n'a pas pu communiquer son incompréhension sur un mot qu'il ne connaît pas.

Conclusion et perspectives

Le dialogue oral homme-machine a fait des progrès indéniables durant ces dernières décennies : grammaires de dialogues, modèles d’ancrage, planification. En outre, depuis quelques années, l’industrie commence à automatiser certaines tâches de dialogue au moyen de dispositifs informatiques. De nombreuses entreprises ont en effet mis au point des plateformes pour implémenter des machines de dialogue à travers un langage standard : le VoiceXML. Si les systèmes implémentés par ce langage sont simples, voire simplistes, ils témoignent cependant du réel intérêt pour le domaine et révèle l’existence de débouchés pour ces techniques nouvelles. Les systèmes reposant sur le langage VoiceXML ne permettent cependant pas une interaction réellement similaire à celle de deux êtres humains puisqu’ils s’appuient sur des formulaires rigides que la machine cherche à remplir à partir du discours humain.

Un des enjeux fondamentaux pour le monde de la recherche dans le domaine du dialogue homme-machine est donc de parvenir à concevoir des systèmes utilisables pour une entreprise dans une situation réelle, et par conséquent suffisamment fiables, permettant un dialogue plus souple et conforme à la pratique orale du langage avec tous ses défauts (bruit, accents, agrammaticalité, etc.). Les propositions actuelles envisagent de sortir de la contrainte du formulaire et mettent au point des modèles sémantiques riches offrant des représentations plus flexibles. Elles cherchent à gagner en robustesse et à rendre les systèmes facilement adaptables aux changements de leur environnement. C’est le cas notamment des systèmes fondés sur des méthodes d’apprentissage par renforcement (POMDP).

Deux constatations fondamentales concernant les limitations des systèmes actuels sont à l’origine de nos recherches. La première consiste à remarquer que si les systèmes de dialogue oral homme-machine peuvent se contenter d’une représentation sémantique à structure simple (formulaire) pour des tâches simples, en revanche lorsqu’on cherche à développer des

systèmes plus évolués capables de réaliser des tâches complexes, une représentation sémantique plus riche et flexible devient nécessaire pour permettre à l'utilisateur de s'exprimer plus librement sans devoir s'adapter à la logique du concepteur du système. En effet, le but final recherché reste un dialogue homme-machine aussi naturel que possible pour l'utilisateur humain.

La deuxième consiste à remarquer que les méthodes d'apprentissage par renforcement à base de modèles MDP et POMDP qui ont été récemment proposées pour automatiser le processus de conception d'un système [Pietquin 2004; Roy et al. 2000; Young et al. 2010] sont difficilement applicables lorsque la représentation de l'état de dialogue devient trop complexe. En effet, les systèmes MDP ne s'avèrent pas suffisamment robustes et les systèmes POMDP proposés pour répondre à ce manque utilisent une variable d'état cachée qui implique l'introduction d'un vecteur de croyance, de taille gigantesque. Les algorithmes de résolution subissent alors une explosion combinatoire.

Aspects novateurs

Afin d'explorer des pistes de réflexion pour surmonter les limites évoquées ci-dessus, nous avons conçu dans ces travaux un paradigme de conception de systèmes de dialogue reposant sur des modèles statistiques d'apprentissage par renforcement et permettant de gérer une représentation sémantique riche.

Tout d'abord, en ce qui concerne la rigidité de la représentation sémantique, un modèle sous la forme de graphes sémantiques, inspiré de FrameNet [Baker et al. 1998], est utilisé. Cette représentation, plus flexible qu'un formulaire préétabli, permet de modéliser des situations de dialogues complexes, dépassant l'application rigide du mécanisme de remplissage d'une liste de couples champ-valeur. Tout l'enjeu de notre travail est donc d'associer ce modèle sémantique riche à un modèle POMDP permettant l'apprentissage par renforcement.

Pour ce faire, nous envisageons une simplification du modèle POMDP en le projetant dans des espaces plus petits à l'aide de fonctions résumées. À la différence de l'approche de [Young et al. 2010], nous mettons en place un double suivi de croyance : dans l'espace maître et dans l'espace résumé. L'optimisation de stratégie peut alors être réalisée dans les espaces résumés avec de véritables algorithmes POMDP. De plus, nous proposons d'optimiser la stratégie du modèle POMDP en utilisant un algorithme avec modèle complet, selon une méthode itérative pour assurer la cohérence entre le modèle et la stratégie optimisée.

Afin de tirer le meilleur parti de la nature stochastique des modèles POMDP, nous avons souhaité intégrer au système l'information statistique contenue dans les scores de confiance *n-best* que peut fournir le module de compréhension.

Toutefois, lors de nos premières tentatives pratiques, nous avons rencontré une difficulté intrinsèque à l'utilisation de POMDP résumés pour le dialogue, difficulté que souligne déjà [Pieraccini et Huerta 2005] : un comportement du système parfois aberrant qui est, dans notre cas, en partie dû au passage dans des espaces résumés qui entraîne une perte d'information. Pour remédier à cela, un système maître basé sur un ensemble de règles expertes permet d'assurer un meilleur contrôle du système POMDP résumé stochastique. Nous avons donc mis en place un système double associant un système maître et un système résumé.

Soucieux de ne pas oublier la dimension pratique de nos travaux théoriques, nous avons pris en compte l'aspect économique concernant les coûts de conception des systèmes. Un des principaux avantages attendu de l'utilisation de méthodes par renforcement consiste en effet à offrir des possibilités d'auto-apprentissage et d'adaptabilité à un changement d'environnement. Afin de prolonger ce principe, notre proposition de POMDP résumé doit pouvoir être appliquée avec un minimum d'intervention humaine. C'est pourquoi, nous avons mis au point les fonctions résumées de notre paradigme POMDP résumé à l'aide d'un *clustering* de graphes et listes de graphes sémantiques par k plus proches voisins.

Résultats

Outre la proposition théorique du paradigme évoqué ci-dessus, ces travaux de thèse ont abouti à sa mise en œuvre à travers le développement d'un prototype de système de dialogue. Nous avons évalué ses performances grâce à des simulations afin de pouvoir étudier l'influence des nombreux paramètres en jeu. Nous avons également mis en œuvre des tests avec des utilisateurs humains pour tester notre prototype dans des situations les plus proches possibles de la réalité. La réalisation concrète de ce prototype a permis de montrer que l'application de ce paradigme est possible dans un cadre réaliste.

Malgré la complexité du modèle sémantique retenu qui, *a priori*, rend impossible l'emploi du POMDP, l'utilisation de fonctions résumées a permis une application effective de techniques d'optimisation de POMDP d'apprentissage par renforcement. Nous avons montré que les algorithmes sur modèle complet (de type PBVI) sont utilisables dans ce cadre et, bien que la stabilité des stratégies obtenues reste encore problématique, on a constaté une amélioration des performances moyennes des stratégies optimisées. Le principe d'optimisation itérative que nous avons proposé permet, en outre, l'adaptation des stratégies de dialogue aux variations de conditions du dialogue. Ce premier travail de combinaison entre une représentation sémantique complexe et un POMDP avec résolution sur modèle complet semble donc ouvrir la voie à un champ d'investigation prometteur.

Afin d'obtenir des corpus de données en quantité suffisante pour réaliser des estimations des modèles de probabilité des POMDP considérés, nous avons eu recours à un utilisateur simulé reposant sur un modèle d'agenda. Le modèle de bruit a également été adapté pour obtenir un environnement de simulation complet. Si ce type d'environnement de simulation a déjà été utilisé par [Schatzmann et al. 2007a], nous avons montré que sa généralisation à un modèle sémantique riche de graphes de *frames* sémantiques donne des résultats satisfaisants.

L'évaluation a été réalisée sur des tâches de demandes d'information touristique et de réservation de chambres d'hôtel, le domaine considéré était par conséquent restreint. La réalisation du prototype a permis une évaluation avec des utilisateurs réels et la mise en évidence des différences entre les critères de performance objectifs et subjectifs. Lors de tests avec des utilisateurs réels, le gain de performances objectives (le taux réel de complétion de la tâche et la récompense moyenne cumulée) des stratégies optimisées confirme l'intérêt d'utiliser l'apprentissage par renforcement pour le dialogue.

À travers les tests dans un environnement de simulation, nos travaux ont également permis de confirmer le gain de robustesse obtenu par l'utilisation de méthodes statistiques que propose [Young et al. 2010]. L'application de ces modèles POMDP demande de réaliser un suivi de croyance. Lors de cette mise à jour de la croyance, la simple observation déter-

ministe issue du module de compréhension peut être remplacée par une liste *n-best* probabilisée associée à des scores de confiance. Cela accroît clairement les performances du système. En revanche, lorsque ce premier suivi de croyance dans l'espace maître est réalisé de façon exacte, la taille de la croyance subit une explosion combinatoire : une approximation (un élagage) est alors nécessaire. Nous montrons alors que l'utilisation d'un deuxième suivi de croyance dans les espaces résumés permet de palier le manque d'exactitude du premier suivi de croyance dans l'espace maître.

Enfin, dans le but d'automatiser au maximum l'application de notre paradigme, nous montrons que la conception de fonctions de projections dans les espaces résumés, qui sont généralement définies de façon *ad hoc*, peut être effectuée par *clustering* automatique avec des performances comparables. Afin d'appliquer les méthodes de *clustering* par *k* plus proches voisins, nous avons dû définir une notion de distance appropriée entre listes *n-best*.

Limites

Malgré les avancées réalisées au cours de nos travaux, il reste encore des difficultés à résoudre avant de pouvoir mettre en place, dans un système de dialogue déployé, le paradigme de POMDP que nous proposons. En effet, même si, au cours de l'apprentissage, le gain de performances montre une amélioration nette lorsque la stratégie du système est optimisée par renforcement, on constate une très forte variabilité de performances lors du processus d'optimisation et la convergence vers une bonne stratégie est instable en pratique.

D'autre part, lors de l'apprentissage, on a remarqué que la stratégie POMDP utilisée seule dégrade fortement les performances, une intervention experte reste nécessaire pour implémenter les mécanismes de création d'actions maîtres. Il a été constaté que la prise en compte d'une pondération sur les actions maîtres créées améliorerait le fonctionnement du système. Ceci reste une limitation de notre système, même si elle permet par ailleurs une meilleure prise en compte de la complétude de conception. [Laroche 2010; Williams 2008b] ont fait des propositions en ce sens.

Perspectives

À propos du modèle sémantique riche

Ces travaux de thèse, en réussissant à associer une structure sémantique par graphes de *frames* sémantiques et un paradigme de POMDP, offrent une ouverture vers une utilisation de modèles sémantiques riches et flexibles. Cependant, cette structure par graphes n'est pas encore suffisante pour représenter toute la richesse d'un dialogue. Notre paradigme de projection par *clustering* dans un modèle POMDP résumé reste toutefois valable pour toute autre représentation plus complexe, dès lors qu'une notion de distance est correctement définie. Il demeure que la conception de fonctions résumées peut être automatisée de la manière que nous présentons pour toute représentation sémantique, même bien plus complexe, tant qu'elle s'exprime sous la forme d'un graphe.

À propos des espaces résumés

Lors de la projection dans des espaces résumés, une certaine perte d'information est inévitable. Le recours à des distances appropriées permet de minimiser la perte d'information utile. Cependant les méthodes de *clustering* par *k* plus proches voisins que nous avons

employées ne tiennent compte de la proximité entre des graphes ou listes de graphes que de manière standard, sans se préoccuper de l'usage spécifique auquel sont destinés les clusters. Étant donné le rôle fondamental joué par la fonction valeur dans le modèle POMDP, il pourrait être intéressant de l'utiliser pour guider le *clustering*. Un tel mécanisme permettrait d'extraire une information plus pertinente, en rapport plus étroit avec l'optimisation de cette fonction valeur.

On pourrait envisager un autre axe de recherche, centré cette fois sur l'optimisation au sein du modèle POMDP : il s'agirait d'établir une interaction plus étroite entre l'algorithme de résolution et le mécanisme de clustering afin de l'utiliser comme une heuristique guidant cet algorithme plus rapidement vers la stratégie optimale.

À propos des relations du gestionnaire de dialogue (DM) avec les modules de reconnaissance vocale et de compréhension

Dans nos travaux, nous avons cherché à faire passer plus d'information à travers le canal de communication entre le SLU et le DM. Pour cela, nous l'avons rendu probabiliste, en transmettant une observation sous la forme d'une liste *n-best* probabilisée.

Cette volonté s'explique par notre constat que les composants en amont du DM (SLU mais aussi ASR) sont « myopes » : ils ne disposent pas d'information globale concernant l'ensemble du dialogue, mais se focalisent sur le traitement d'un seul tour de parole. Dès lors, cette intégration d'informations probabilistes permet de repousser la sélection du *n-best* jugé exact, afin que la décision soit prise par un module plus à même de le faire : le DM qui a une vision plus globale.

L'utilisation de modèles sémantique complexes, tant du point de vue de leur richesse de représentation que de leur calculabilité, offre la promesse de systèmes de dialogue plus adaptés au langage naturel d'utilisateurs humains. Il paraît donc évident que de nombreuses expériences restent utiles, et nécessaires, pour délimiter le contour d'une utilisation efficace du paradigme d'apprentissage par renforcement.

Annexes

Sommaire

Annexe A : Détails d'implémentation du prototype.....	172
Annexe B : Protocole expérimental.....	175
B.1 Aide-mémoire pour les expérimentateurs.....	175
B.2 Questionnaire pour les utilisateurs.....	175
Annexe C : Démonstrations mathématiques.....	178
C.1 Distance de Skorokhod.....	178
C.2 Propriété de la distance d_1	179
C.3 Propriété de la mesure de similarité d_2	180

Annexe A : Détails d'implémentation du prototype

Cette annexe décrit le fonctionnement interne du prototype, explicitant certains des choix d'implémentations réalisés.

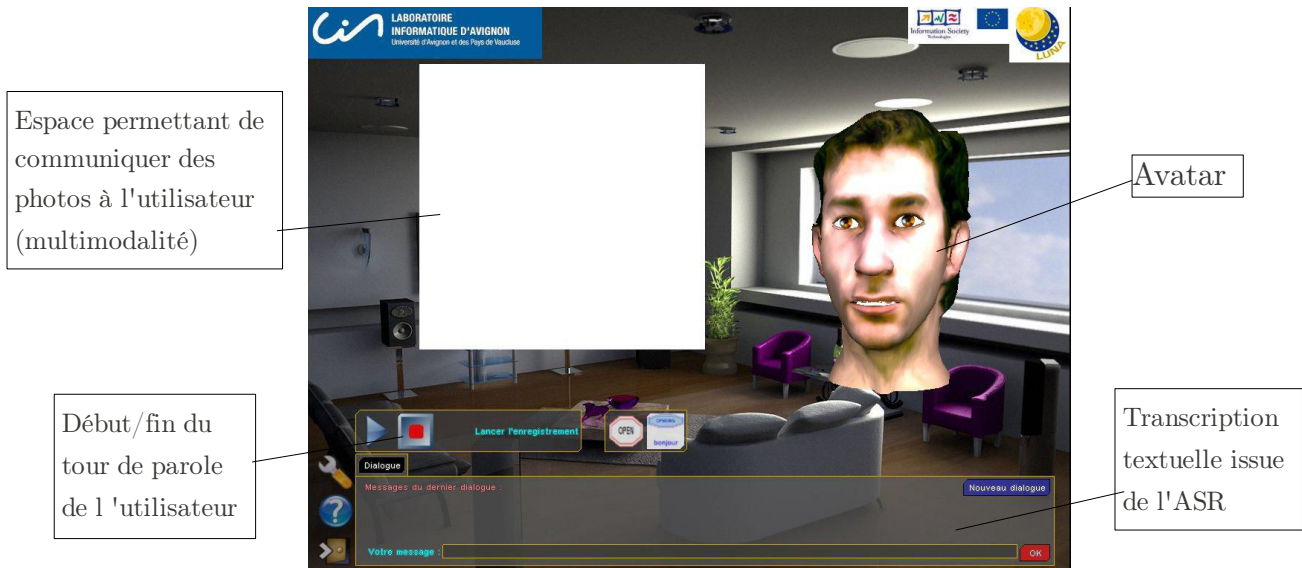


Figure A.1: Interface graphique du prototype de système de dialogue oral homme-machine.

Le prototype est agencé autour d'une architecture de communication centralisée où un serveur central transmet les communications aux autres composants du système.

Le modules constituant le système sont les suivants :

- Serveur de communication : met en relation les différents composants entre eux et communique par Corba ou par socket TCP.
- Reconnaissance vocale (ASR) : on a choisi le logiciel de reconnaissance vocale *Speeral* [Linarès et al. 2007] développé au Laboratoire d'Informatique d'Avignon (CERI). Ce module reçoit le signal acoustique au format *.wav* et fournit une transcription *1-best* ou un treillis de mots.
- Compréhension (SLU) : Ce composant a été développé selon les principes de [Meurs 2009] et [Duvert et al. 2008]. Il produit une annotation en concepts et en *frames*. Reçoit un le *1-best* ou le treillis de mots et fournit une liste *n-best* de concepts et de *frames*.
- Gestionnaire de dialogue (DM) : Le Gestionnaire de Dialogue met à jour la croyance courante dans l'espace d'état maître : les mémoires de *frames*. (généralement trois au maximum). Il extrait la valeur d'observation résumée. Il déduit des actions possibles à partir de règles. Il utilise les scores des DM-Pomdp et DM-rules pour prendre un décision. Enfin, il génère des actions réponses (NLG)

- DM-Pomdp: ce module met à jour le vecteur de croyance dans l'espace résumé, il dispose d'une stratégie optimisée sous forme de alpha-vecteurs et il l'utilise pour fournir un score sur les actions proposées par le DM.
- Génération de texte (NLG) : Le module de génération de texte est intégré au gestionnaire de dialogue.
- Synthèse vocale (TTS) : pour le composant de synthèse vocale, on utilise un synthétiseur Mbrola [Dutoit et al. 1996].
- Client graphique : le client graphique assure l'interface avec l'utilisateur. Il utilise une librairie graphique permettant d'afficher un avatar (figure A.1). Il est portable et s'exécute sous Windows ou Linux. Il est également possible de lancer le client en mode texte afin de tester le système.
- Utilisateur simulé : Pour les simulations, on a choisi de réutiliser la librairie logicielle de l'utilisateur simulé développé à par [Schatzmann et al. 2007a]. On a ajouté un module transformant les actes de dialogue simples de cette librairie en représentation en *frames* sémantiques. Ce module génère des données au niveau intentionnel destinés à être directement envoyé au Gestionnaire de Dialogue : des *n-best* d'annotation en *frames*, au même format que le module de SLU.

Ressources propres au système : accès à une base de données et mémoire

Selon la tâche considérée, le système peut avoir accès à des ressources non disponibles à l'utilisateur. Ces ressources peuvent être une base de données pour une tâche de consultation de compte bancaire, ou des commandes de moteurs pour une tâche de guidage de robot. Ce type de ressources est généralement considéré comme faisant partie intégrante du système et est directement géré par le gestionnaire de dialogue.

De plus, sauf pour des tâches très spécifiques ne nécessitant pas d'assurer un suivi du dialogue, un mécanisme de mémoire de travail est également ajouté au gestionnaire de dialogue.

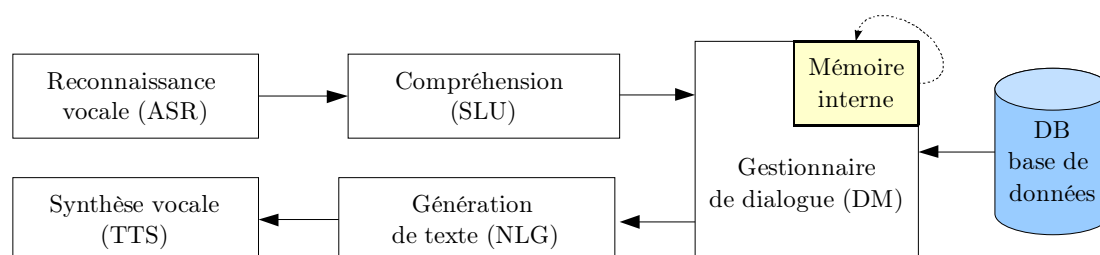


Figure 2: Système de dialogue classique.


```
function merge(g1,g2) // merge g2 into g1
  foreach f2 in g2.get_head_frames
    label start_loop:
      f1 = g1.get_mergepoint(f2)
      if f1
        f1.merge(f2)
      else
        f2 = create_parent(f2, ontologie)
        g2.insert(f2)
        goto start_loop

function merge_frames(f1,f2)
  foreach e2 in f2
    e1 = f1.get_mergepoint(e2)
    if e1
      merge_frame_element(e1,e2)
    else
      f1.append(e2)

function merge_frame_element(e1, e2)
  if e1.is_slot() and e2.is_slot()
    e1.value.update(e2.value)
    e1.grounding_state.update(e2.grounding_act)
  else // e1 and e2 links to others frames
    f1 = e1.target
    f2 = e2.target
    merge_frames(f1, f2)
```

Tableau A.1: Intégration de nouvelles informations sémantiques dans un graphe mémoire de frames (pseudo-code).master cite

Annexe B : Protocole expérimental

B.1 Aide-mémoire pour les expérimentateurs

Matériel nécessaire :

- Liste des sujets avec leur numéro d'anonymat.

Pour la première session :

- Faire lire l'explication et signer le consentement.
- Expliquer si nécessaire.

Pour chaque dialogue :

- Préparation - Configuration :
 - Réinitialiser l'application pour éviter tout effet de bord.
(vitesse différente ou perturbation dû à la consommation mémoire)
 - Entrer les données de configuration :
 - numéro d'anonymat
 - numéro de dialogue et de session
 - Charger le scénario.
- Lors du dialogue : **Ne pas intervenir !**
- Questionnaire du dialogue :
 - Donner le questionnaire de dialogue à remplir .
 - Sauver les logs, vérifier.

A la fin de la session :

- Questionnaire de la session.
- Vérifier les logs.

B.2 Questionnaire pour les utilisateurs

Explication de l'expérience et consentement

Vous allez participer à une expérience avec un système de dialogue homme-machine au Centre d'Études et de Recherche en Informatique (Université d'Avignon et des Pays de Vaucluse).

Le but de ces expériences est d'étudier les systèmes de dialogue Homme-Machine.

Vous permettez, par le fait de participer à l'expérience, que les données audio, transcriptions et informations anonymisées (âge, genre, langue maternelle) soient enregistrées, utilisées et publiées dans un but de recherche et développement.

Vous pourrez changer d'avis et refuser votre consentement en l'annonçant à la fin de l'expérience.

Nom :

Prénom :

j'accepte je refuse

Expérience

L'expérience consiste en une série de 4 dialogues. Un scénario vous sera fourni pour chaque dialogue. Entre chaque dialogue, vous aurez un questionnaire à remplir, et à la fin de l'expérience, un dernier questionnaire.

Vous allez jouer le rôle d'un touriste interagissant avec un système de réservation d'hôtels et d'informations touristiques. Vous pourrez parler librement mais dans le but de réaliser le scénario.

Votre but n'est pas d'analyser le fonctionnement du système ni de chercher ses limites, mais de jouer le scénario donné.

Attention : Pour des raisons techniques, veuillez toujours commencer le dialogue en disant simplement bonjour à votre interlocuteur.

Cette expérience prendra environ 20 minutes.

Données anonymisées

Genre : Homme Femme

Degré d'expertise : Expert TAL Utilisateur normal

Age (facultatif) :

Numéro d'anonymat :

Questionnaire dialogue 1 (quatre exemplaires de 1 à 4)

Dialogue 1 : A votre avis, le scénario donné était-il réalisable ? (était-il possible au système de répondre à votre demande d'après sa base de données ?)

- Oui Non Je ne sais pas

Dialogue 1 : Si oui, le système a-t-il réussi à vous fournir ce que vous lui demandiez ?

- Oui Plutôt oui Plutôt non Non

Dialogue 1 : Si non, le système a-t-il réussi à vous communiquer que votre demande était impossible à satisfaire ?

- Oui Plutôt oui Plutôt non Non

Dialogue 1 : Les réponses du système étaient-elles plutôt appropriées à la situation ou bien hors de propos ?

- Toujours appropriées
 Très souvent appropriées
 Souvent appropriées
 Parfois appropriées et parfois hors de propos
 Souvent hors de propos
 Très souvent hors de propos
 Toujours hors de propos

Dialogue 1 : Qui a principalement guidé la conversation ?

- Vous. Plutôt vous. Les deux ou aucun des deux.
 Plutôt le système. Le système.

Dialogue 1 : Était-il facile et intuitif de communiquer avec le système ?

- Très facile Facile Moyennement facile Difficile Très difficile

Questionnaire session

Quelle note (sur dix) donneriez-vous à la performance globale du système pour chaque dialogue ?

- Premier dialogue : / 10
 Deuxième dialogue : / 10
 Troisième dialogue : / 10
 Quatrième dialogue : / 10

Annexe C : Démonstrations mathématiques

C.1 Distance de Skorokhod

La distance de Skorokhod a été proposée sous plusieurs variantes (distances J_1 , J_2 , M_1 et M_2) dans [Skorokhod 1956]. Elle est utilisée dans l'étude des limites des processus stochastique [Silvestrov 2006]. La distance de Skorokhod la plus couramment utilisée est la distance J_1 (formule 86). [Contreras 2009] fournit une introduction simple aux distances de Skorokhod.

Définition de J1

Elle s'applique aux fonctions, ayant la propriété càdlàg : continue à droite et avec limite à gauche

$$d_{J_1}(b, b') = \inf_{\lambda \in \Lambda} (\sup_t |t - \lambda(t)| \vee \sup_t |b(t) - b'(\lambda(t))|) \quad (86)$$

Ou, de façon équivalente :

$$d_{J_1}(b, b') = \inf_{\lambda \in \Lambda} (\|Id - \lambda\|_\infty \vee \|b - b' \circ \lambda\|_\infty) \quad (87)$$

où Λ est l'ensemble des fonctions strictement croissantes adaptées aux espaces de départ de b et b' .

La distance de Skorokhod est initialement une distance entre fonctions dont les espaces de départ et d'arrivée sont munis de normes. Or, l'espace des graphes, n'est pas muni d'une norme, mais seulement d'une distance. Une variante (la distance J_2) proposée dans le papier initial de Skorokhod permet d'écrire une définition de cette distance valable dans ce cas :

Définition de J2

$$d_{J_1}(b, b') = \inf_{\lambda \in \Lambda} (d_\infty(Id, \lambda) \vee d_\infty(b, b' \circ \lambda)) \quad (88)$$

$$d_\infty(Id, \lambda) = \sup_t d(t, \lambda(t)) \quad (89)$$

$$d_{J_1}(b, b') = \inf_{\lambda \in \Lambda} (\sup_t d(t, \lambda(t)) \vee \sup_t d(x(t), y(\lambda(t)))) \quad (90)$$

La distance J_2 est cette fois une véritable distance. Cependant, elle est complexe à calculer. De plus, si elle est justifiée dans un cadre d'analyse fonctionnelle, elle n'a pas de justification du point de vue probabiliste.

C.2 Propriété de la distance d_1

Rappel de la définition

Soit deux distributions de probabilité b et b' , soient

$$\mathbb{X}_b = \{X \text{ variable aléatoire de loi } b\}$$

$$\mathbb{X}_{b'} = \{X' \text{ variable aléatoire de loi } b'\}$$

On définit la distance $d_1(b, b')$ par un minimum d'espérance de la distance entre deux variables de lois b et b' selon :

$$d_1(b, b') = \inf_{(X, X') \in \mathbb{X}_b \times \mathbb{X}_{b'}} \mathbb{E}(d(X, X')) \quad (91)$$

Réécriture sous la forme d'un problème d'optimisation de forme linéaire

Nous supposons que les distributions b et b' sont à supports finis ¹. Dans ce cas, nous pouvons réécrire la définition (91) sous la forme d'un problème de programmation linéaire.

On note (A_1, \dots, A_n) la suite finie des points constituant le support de b . Et on note (A'_1, \dots, A'_m) la suite finie des points constituant le support de b' . Le calcul de la d_1 consiste à minimiser $\mathbb{E}(d(X, X'))$ sur l'ensemble des lois possibles pour le couple X et X' de lois marginales respectives b et b' .

Variables

Notons $c_{ij} = P(X = A_i, X' = A'_j)$ les coefficients de la loi conjointe du couple (X, X') .

Les c_{ij} sont nos variables.

Contraintes

La condition « X est de loi » s'exprime par $\sum_j c_{ij} = a_i$ pour tout i . (92)

)

La condition « X' est de loi » s'exprime par $\sum_i c_{ij} = b_j$ pour tout j . (93)

)

De plus, tous les coefficients sont positifs. (94)

)

Fonction objective

Enfin, l'espérance $\mathbb{E}(d(X, X'))$ s'écrit :

$$\mathbb{E}(d(X, X')) = \sum_{i=1}^n \sum_{j=1}^m c_{ij} \cdot d(A_i, A'_j) \quad (95)$$

Le problème consiste à minimiser la forme linéaire (95) sous les contraintes (92), (93) et (94), qui sont des équations et inéquations linéaires.

(1) Une distribution est dite à support fini si elle est une somme finie de Dirac. Sa probabilité est non-nulle seulement sur un nombre fini de graphes.

Lemme : *La borne inférieure dans la définition de d_1 est atteinte :*

$$d_1(b, b') = \min_{(X, X') \in \mathbb{X}_b \times \mathbb{X}_{b'}} \mathbb{E}(d(X, X')) \quad (96)$$

Par conséquent, on a l'existence de deux variables X_0 et X'_0 de lois b et b' telles que

$$d_1(b, b') = \mathbb{E}(d(X_0, X'_0)) \quad (97)$$

Démonstration :

La forme linéaire (95) est continue. De plus, l'ensemble des contraintes (92), (93) et (94) définissent un ensemble fermé et borné (donc compacte en dimension finie). Par conséquent, la borne inférieure $\inf_{(X, X') \in \mathbb{X}_b \times \mathbb{X}_{b'}} \mathbb{E}(d(X, X'))$ est atteinte.

Propriétés : d_1 possède les propriétés suivantes.

$$\forall b, \quad d_1(b, b) = 0 \quad (98)$$

$$d_1(b, b') = 0 \Rightarrow b = b' \quad (99)$$

$$\forall b, \forall b', \quad d_1(b, b') = d_1(b', b) \quad (100)$$

$$\forall b, \forall b', \forall b'', \quad d_1(b, b') + d_1(b', b'') \geq d_1(b, b'') \quad (101)$$

Démonstration :

(98) et (99) sont évidents. (100) l'est également en considérant l'écriture de d_1 sous la forme de la formule (97).

Pour la propriété (101), écrivons les deux distances $d(b, b')$ et $d(b', b'')$ sous la forme (97). On peut trouver des variables aléatoires telles que

$$d_1(b, b') = \mathbb{E}(d(X_0, X'_0)) \quad \text{et} \quad d_1(b', b'') = \mathbb{E}(d(X'_1, X''_1))$$

Remarquons qu'il est possible de choisir la même variable pour la loi b' : $X'_0 = X'_1$. On obtient alors :

$$\begin{aligned} d_1(b, b') + d_1(b', b'') &= \mathbb{E}(d(X_0, X'_0)) + \mathbb{E}(d(X'_1, X''_1)) \\ &\geq \mathbb{E}(d(X_0, X''_1)) \\ &\geq d_1(b, b'') \end{aligned}$$

Propriété : d_1 est une distance.

La déduction est immédiate à partir des propriétés (98), (99), (100) et (101).

C.3 Propriété de la mesure de similarité d_2

Propriété : *La mesure de similarité (distance simplifiée) d_2 est une forme bilinéaire et s'écrit selon la formule (81 bis) suivante :*

$$d_2(b, b') = \sum_{\substack{1 < i \leq n \\ 1 < j \leq m}} b_i b'_j d(g_i, g'_j) \quad (81 \text{ bis})$$

Démonstration :

La mesure de similarité est déjà définie sur la partie $\Phi(G)$ de B , en effet $\Phi(G)$ est constituée de listes n -best réduites à un seul élément, de poids 1.0.

Pour deux éléments δ_g et $\delta_{g'}$ de $\Phi(G)$, on définit

$$d_2(\delta_g, \delta_{g'}) = d(g, g') \quad (102)$$

Comme $\Phi(G)$ est une base (non-orthonormée) de B , on peut étendre cette définition par linéarité.

Pour deux éléments b et b' de B :

$$b = \sum_{i=1}^n b_i \cdot \delta_{g_i} \text{ et } b' = \sum_{j=1}^m b'_j \cdot \delta_{g'_j} \quad (103)$$

On définit

$$d(b, b') \triangleq \sum_{\substack{1 < i \leq n \\ 1 < j \leq m}} b_i b'_j d(\delta_{g_i}, \delta_{g'_j}) \quad (104)$$

Soit, en utilisant (102),

$$d(b, b') = \sum_{\substack{1 < i \leq n \\ 1 < j \leq m}} b_i b'_j d(g_i, g'_j) \quad (105)$$

On a donc défini une mesure de similarité qui possède à la propriété (74).

Index des illustrations

Figure 1: Décomposition d'un système de dialogue en ses différents modules.....	9
Figure 1.1: Boucle de fonctionnement d'un système de dialogue homme-machine.....	19
Figure 1.2: Élargissement des canaux de communications entre certains modules (fat pipeline).20	
Figure 1.3: Arbre sémantique utilisé comme modèle pour définir l'ontologie du système HIS. Extrait de [Young et al. 2010].....	24
Figure 1.4: Relations de la frame <code>operating_a_system</code> avec les frames voisines dans FrameNet [Fillmore 1985].....	26
Figure 1.5: Système de dialogue avec modèle sémantique flexible.....	35
Figure 1.6: Graphe de décision d'un MDP, est l'état à la date t et est l'état à la date $t+1$	38
Figure 1.7: Graphe de décision d'un POMDP.....	39
Figure 1.8: Représentation détaillée d'un POMDP.....	40
Figure 1.9: Représentation compacte d'un POMDP.....	40
Figure 1.10: Résumé de croyance :	44
Figure 1.11: Résumé d'action :	45
Figure 1.12 : Résumés simultanés de croyance et d'action :	46
Figure 1.13: Diagramme d'influence d'un POMDP dont l'état est décomposé en trois composants : but, action utilisateur et historique.....	50
Figure 1.14: Factorisation de l'état de dialogue en trois composantes.....	50
Figure 1.15: Modèle de POMDP résumé de Williams lors de l'optimisation : un résumé de l'état et de l'observation.....	56
Figure 1.16 : Modèle de POMDP résumé de Williams lors de l'exécution : un résumé de la croyance.....	56
Figure 1.17: Raffinement de la partition utilisée pour représenter la croyance dans le système HIS (extrait de [Young et al. 2010]).....	59
Figure 1.18: Résumé de croyance avec le système HIS.....	61
Figure 1.19: Réseau bayésien du modèle de Pietquin de simulation de l'utilisateur (extrait de [Pietquin 2004]).....	68
Figure 2.1: Une représentation sémantique à ontologie flexible.....	78
Figure 2.2 : État de dialogue permettant de générer plusieurs requêtes.....	82
Figure 2.3: POMDP sur résumé de croyance (double suivi de croyance).....	89
Figure 2.4: Graphe de frames sémantiques 1.....	102
Figure 2.5: Graphe de frames sémantiques 2.....	102
Figure 2.6: Graphe réduit 1.....	102
Figure 2.7: Graphe réduit 2.....	102
Figure 2.8: Élaboration des états et des observations du corpus résumé.....	109
Figure 2.9: Procédé itératif d'optimisation de stratégie POMDP.....	113
Figure 3.1: Élaboration des états et des observations du corpus résumé.....	125
Figure 3.2 : Système « SLU », sans erreur de reconnaissance (annotation de référence pour l'ASR) et avec erreur de compréhension SLU.....	127
Figure 3.3 : Système « ASR+SLU », avec erreur de reconnaissance ASR et de compréhension SLU.....	127
Figure 3.4: Effet du bruit ASR sur les performances du système résumé selon différents horizons de résolution avec l'algorithme PBVI.....	128
Figure 3.5: Graphe de transitions du modèle d'ancrage. Les nœuds sont les états d'ancrage. Les flèches rouges (dont les étiquettes commencent par u) représentent les transitions dues aux	

actions de l'utilisateur. Les flèches bleues (dont les étiquettes commencent par s) représentent les transitions dues aux actions du système..... 135

Figure 3.6: Application du procédé itératif d'optimisation de stratégie POMDP.....139

Figure 3.7: Taux de complétion du système HCSum durant l'apprentissage pour différents niveaux d'erreur (moyenne sur 10 expériences)..... 141

Figure 3.8: Récompense moyenne obtenue par le système HCSum durant l'apprentissage, pour différents niveaux d'erreur (la courbe est lissée en exécutant 10 fois l'expérience et en prenant la moyenne)..... 141

Figure 3.9: Taux de complétion et récompense moyenne du système POMDP entraîné comparé au système de référence selon le niveau de bruit..... 142

Figure 3.10: Taux de complétion du système HCSum durant l'apprentissage avec trois niveaux différents de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).....144

Figure 3.11: Taux de complétion du système HCSum avec trois niveaux différents de suivi de croyance dans l'espace maître (moyenne sur 10 expériences).....144

Figure 3.12: Taux de complétion du système HCSum durant l'apprentissage avec trois niveaux différents de suivi de croyance dans l'espace maître. Les systèmes intitulés POMDP et MDP sont respectivement avec ou sans suivi de croyance dans l'espace résumé (moyenne sur 10 expériences)..... 145

Figure 3.13: Taux de complétion après optimisation pour des systèmes MDP et POMDP selon la taille maximale de la distribution de suivi de croyance dans l'espace maître (moyenne sur 10 expériences)..... 146

Figure 3.14: Récompense moyenne obtenue lors de l'entraînement par le système avec clustering automatique ClustSum pour différents nombres de clusters.....148

Figure 3.15: Taux de complétion lors de l'entraînement par le système avec clustering automatique ClustSum pour différents nombres de clusters.....148

Figure 3.16: Taux de complétion obtenu lors de deux optimisations avec les mêmes paramètres. Niveau de bruit 0,5..... 150

Figure 3.17: Performances obtenues durant l'optimisation lors de chacune des 10 expériences (en couleur). La courbe moyenne (en noir) est la moyenne des dix courbes en supprimant les minimum et maximum (moyenne sur 8 valeurs). Niveau de bruit 0,5.....151

Figure 3.18: Récompense cumulée obtenue en fin de dialogue.....154

Figure 3.19: Récompense cumulée en fin de dialogue, avec lissage.....154

Figure 3.20: Répartition de la durée des dialogues..... 155

Figure 3.21: Répartition de la durée des dialogues, avec lissage.....156

Figure 3.22: Propositions des réponses à la question « Si oui, le système a-t-il réussi à vous fournir ce que vous lui demandiez ? »..... 158

Figure 3.23: Propositions des réponses à la question « Si non, le système a-t-il réussi à vous communiquer que votre demande était impossible à satisfaire ? ».....158

Figure 3.24: Succès du dialogue, évalué par une note sur 10 (critère subjectif quantitatif).....159

Figure 3.25: Évaluation de la compréhension du système (question Q4).....160

Figure 3.26: Réponse à la question visant à évaluer le partage d'initiative lors du dialogue pour les deux systèmes comparés.....160

Figure 3.27: Réponse à la question visant à évaluer la facilité d'utilisation du système pour les deux systèmes comparés.....161

Figure A.1: Interface graphique du prototype de système de dialogue oral homme-machine.....172

Figure 2: Système de dialogue classique..... 173

Index des tables

Tableau 1.1: Exemples de phrases difficiles à annoter avec les paradigmes DAMSL, MRDA ou DATE.....	22
Tableau 1.2: Exemple de dialogue entre deux interlocuteurs (A et B) illustrant l'évolution de l'état d'ancrage d'un élément d'information.....	22
Tableau 1.3: Définition FrameNet de la frame sémantique <code>operating_a_system</code>	25
Tableau 1.4 : Exemple d'annotation en frames du corpus Media-Luna. La frame VOULOIR pointe sur la frame RÉSERVATION qui pointe sur les frames HÔTEL et PÉRIODE.	27
Tableau 1.5: Récapitulatif non-exhaustif des prototypes de systèmes de dialogue publiés.	34
Tableau 1.6: Publication d'évaluations de systèmes de dialogue.....	72
Tableau 1.7: Questionnaire de satisfaction utilisateur utilisé dans les tests de [Walker et al. 2000].	73
Algorithme 2.1 : Intégration d'un nouveau graphe sémantique dans un graphe mémoire	81
Tableau 2.2 : Listes des requêtes à la base de données (générées à partir de l'état courant de la figure 2.2).....	82
Tableau 2.3: Résultats de la bases de données, obtenus à partir des requêtes autorisées du tableau 2.2.	84
Tableau 2.4: Actions maîtres correspondant à l'action résumée Offre.....	85
Tableau 2.5: Actions maîtres correspondant à l'action résumée ConstQ.....	85
Tableau 2.6: Retour à l'espace maître proposé dans [Young et al. 2007].....	87
Tableau 2.7: Couplage de systèmes résumé et par règles proposé dans cette thèse.....	87
Tableau 2.8: Fonction de compatibilité des actions avec les but et historique courants.....	91
Tableau 2.9: Extrait de la définition de la projection d'état résumé	94
Tableau 2.10 : Extrait de la définition de la projection d'observation résumée	94
Tableau 2.11: Modèle d'observation :	95
Tableau 2.12: Modèle de transition d'état :	95
Tableau 2.13: Exemples de caractéristiques primaires extraites de l'historique de structures de frames.....	97
Tableau 2.14: Exemples de caractéristiques secondaires élaborées à partir des caractéristiques primaires.....	98
Tableau 2.15: Exemple de chemin d'édition entre les graphes réduits 1 et 2.....	103
Tableau 2.16: Matrice d'adjacence conjointe des deux graphes.....	103
Tableau 2.17: Nombre de degrés de liberté du modèle de probabilité d'un POMDP.....	110
Tableau 2.18: Extrait d'une simulation de dialogue Umlib avec les représentations en actes de dialogue et en frames sémantiques associées.....	111
Algorithme 2.19 : Principe de simulations itératives.....	113
Tableau 3.1: Trois niveaux de granularité pour l'évaluation.....	118
Tableau 3.2: Exemple de scénario pour la tâche 1 : réservation d'hôtel.....	119
Tableau 3.3: Exemple de scénario pour la tâche 2 : recherche de prix.....	119
Tableau 3.4: Exemple de scénario pour la tâche 3 : réservation négociée.....	120
Tableau 3.5: But utilisateur sous forme de formulaire.....	120
Tableau 3.6: Définition du succès et de l'échec d'un dialogue avec connaissance du but utilisateur et de la base de données.....	121
Tableau 3.7: Récompense immédiate avec information sur le but exact de l'utilisateur et sur la base de données.....	122
Tableau 3.8: Récompense immédiate rudimentaire.....	122

Tableau 3.9: Disponibilité des critères d'évaluation des dialogues selon les conditions de réalisation.....	124
Algorithme 3.10 : Algorithme de la fonction « intégrerGraphe(G1 , G2) » permettant d'intégrer un nouveau graphe de frames sémantiques G1 dans un graphe mémoire G2.	132
Tableau 3.11: Liste des états d'ancrage dans l'implémentation du prototype.....	134
Tableau 3.12: Liste des actes d'ancrage dans l'implémentation du prototype.....	134
Tableau 3.13: Résultats du clustering de graphes pour la conception de la fonction résumée définissant l'état résumé à partir de l'état maître.....	136
Tableau 3.14: Résultats du clustering de listes n-best de graphes pour la conception de la fonction de projection résumée définissant l'observation résumée à partir de la croyance maître.	136
Tableau 3.15: Exemple de hiérarchie des scores des actions maîtres.....	137
Tableau 3.16: Le nombre de dialogues générés à chaque itération suit une croissante géométrique de raison 1,5 de premier terme 50 (arrondi au multiple de 50 inférieur).....	139
Tableau 3.17: Exemple de deux listes d'observations en 3-best. On suppose que le tour de dialogue exact est « Nyons », tandis que « Lyon » et « non » en sont des versions bruitées. .	140
Tableau 3.18: Récompense moyenne du système POMDP entraîné comparé au système de référence selon le niveau de bruit.....	142
Tableau 3.19: Les données de la figure 3.13 avec les écart-types.....	146
Tableau 3.20: Performances comparées des systèmes avec fonctions résumées ad hoc (HCSum) et automatiques (ClustSum). Les gains sont calculés relativement au système de référence (qui fonctionne sans apprentissage statistique).....	147
Tableau 3.21: Taux de complétion du système HCSum pour différents niveaux de couplage par le coefficient entre les deux stratégies maître et résumée . Niveau de bruit 0,4.....	149
Tableau 3.22: Synthèse des résultats de l'évaluation objective lors des tests avec des utilisateurs humains.....	156
Tableau 3.23: Moyennes obtenues pour les questions Q2 et Q3 (valeurs comprises entre 0 et 3)	159
Tableau 3.24: Moyennes obtenues pour la question Q5.....	161
Tableau 3.25: Synthèse des résultats de l'évaluation par des utilisateurs humains.....	162
Tableau 3.26: Exemple de manque de confirmation et défaut de compréhension conduisant à un désaccord entre critères subjectif et objectif.	163
Tableau A.1: Intégration de nouvelles informations sémantiques dans un graphe mémoire de frames (pseudo-code).master cite	174

Bibliographie

- H. Ai et D. J. Litman** (2008). *Assessing dialog system user simulation evaluation measures using human judges*, Association for Computational Linguistics (ACL) : 622-629.
- J. Allen** (1995). *Natural language understanding*, Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA.
- J. Allen, B. Miller, E. Ringger et T. Sikorski** (1996). *A Robust System for Natural Spoken Dialogue*, Association for Computational Linguistics (ACL) : 62-70.
- J. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. G. Martin, B. W. Miller, M. Poesio et D. R. Traum** (1994). *The TRAINS Project: A case study in building a conversational planning agent*, Journal of Experimental and Theoretical AI 7: 7-48.
- H. A. Almohamad et S. O. Duffuaa** (1993). *A Linear Programming Approach for the Weighted Graph Matching Problem*, IEEE Trans. Pattern Anal. Mach. Intell. 15: 522-525.
- C. Amato** (2007). *Solving POMDPs using quadratically constrained linear programs*, Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI) : 2418-2424.
- J. Andrés-Ferrer et H. Ney** (2009). *Extensions of absolute discounting (Kneser-Ney method)*, International Conference on Acoustics, Speech, and Signal Processing (ICASSP) : 4729-4732.
- J. Austin** (1962). *How to do thing with words*, Oxford Clarendon Press.
- C. Baker, C. Fillmore et J. Lowe** (1998). *The Berkeley FrameNet project*, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING).
- A. Barto et S. Mahadevan** (2003). *Recent advances in hierarchical reinforcement learning*, Discrete Event Dynamic Systems 13: 41-77.
- T. Bazillon, Y. Estève et D. Luzzati** (2008). *Manual vs assisted transcription of prepared and spontaneous speech*, Language Resources and Evaluation Conference (LREC).
- R. Bigot** (2005). *La diffusion des technologies de l'information dans la société française*, Centre de recherche pour l'étude et l'observation des conditions de vie (CREDOC).
- A. W. Black et M. Eskenazi** (2009). *The spoken dialogue challenge*, Proceedings of the 10th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 337-340.
- D. Bohus, A. Raux, T. Harris, M. Eskenazi et A. Rudnicky** (2007). *Olympus: an open-source framework for conversational spoken language interface research*, Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies : 32-39.
- D. Bohus et A. Rudnicky** (2002). *LARRI: A Language-Based Maintenance and Repair Assistant*, Spoken multimodal human-computer dialogue in mobile environments : 203-218.
- D. Bohus et A. I. Rudnicky** (2003). *Ravenclaw: dialog management using hierarchical task decomposition and an expectation agenda*, Eurospeech : 597-600.
- D. Bohus et A. I. Rudnicky** (2009). *The RavenClaw dialog management framework:*

- Architecture and systems*, Computer Speech and Language (CSL) 23: 332 - 361.
- H. Bonneau-Maynard et F. Lefèvre** (2006). *Un modèle stochastique de compréhension de la parole à 2+1 niveaux*, Actes des Journées d'Etude sur la Parole (JEP).
- H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn et D. Mostefa** (2005). *Semantic Annotation of the French Media Dialog Corpus*, 9th European Conference on Speech Communication and Technology.
- H. Bonneau-Maynard, K. Mctait, D. Mostefa, L. Devillers, S. Rosset, P.Paroubek, C. Bousquet, K. Choukri, J. Goulian, J-Y Antoine, F. Béchet, O. Bontron, L. Charnay, L. Romary, N.Vergnes, N. Vigourous**, (2004). *Constitution d'un corpus de dialogue oral pour l'évaluation automatique de la compréhension hors et en contexte du dialogue*, Actes des Journées d'Etude sur la Parole (JEP 2004), Fès, Maroc, avril 2004.
- J. Bos, E. Klein, O. Lemon et T. Oka** (2003). *DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture*, Proceedings of the 4th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 115-124.
- H. Bunke** (1997). *On a relation between graph edit distance and maximum common subgraph*, Pattern Recognition Letters 18: 689 - 694.
- S. Chetramohan, M. Geist et O. Pietquin** (2010). *Optimizing Spoken Dialogue Management with Fitted Value Iteration*, Proceedings of 11th Annual Conference of the International Speech Communication Association (Interspeech 2010) : 86-89.
- Z. Chen et H. Ji** (2010). *Graph-Based Clustering for Computational Linguistics: A Survey*, Proceedings of TextGraphs, Workshop on Graph-based Methods for Natural Language Processing : 1-9.
- A. Cheyer et D. Martin** (2001). *The Open Agent Architecture*, Journal of Autonomous Agents and Multi-Agent Systems 4 : 143-148.
- N. Chomsky** (1956). *Three models for the description of language*, Information Theory 2 : 113-124.
- G. E. Churcher, E. S. Atwell et C. Souter** (1997). *Dialogue management systems: a survey and overview*, University of Leeds.
- L. V. Contreras** (2009). *Skorokhod Metric*, <http://hdl.handle.net/10139/2750> [obtenu le 10 juin 2011].
- R. Cooper, S. Larsson, C. Matheson, M. Poesio et D. Traum** (1999). *Coding Instructional Dialogue for Information States*, Trindikit Deliverable D1.1.
- M. Core, J. Allen et D. Traum** (1997). *Coding Dialogues with the DAMSL Annotation Scheme*, AAI Fall Symposium on Communicative Action in Humans and Machines : 28-35.
- G. Corona et F. Charpillet** (2010). *Distribution over Beliefs for Memory Bounded Dec-POMDP Planning*, Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI).
- H. Cuayáhuitl** (2009). *Human-Computer Dialogue Simulation Using Hidden Markov Models*, Thèse de doctorat. Institute for Communicating and Collaborative Systems School of Informatics, University of Edinburg.
- H. Cuayáhuitl, S. Renals, O. Lemon et H. Shimodaira** (2005). *Human-Computer Dialogue Simulation Using Hidden Markov Models*, ASRU : 290-295.
- N. Dahlbäck** (1997). *Towards a Dialogue Taxonomy*, Workshop on Dialogue Processing in Spoken Language Systems : 29-40.

- G. Damnati, F. Béchet et R. De Mori** (2007). *Experiments on the France Telecom 3000 Voice Agency corpus: academic research on an industrial spoken dialog system*, Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies : 48-55.
- D. L. Davies et D. W. Bouldin** (1979). *A Cluster Separation Measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1: 224-227.
- P. Dillenbourg et M. Baker** (1996). *Negotiation spaces in human-computer collaborative learning*, Proceedings of the International Conference on Cooperative Systems : 12-14.
- T. Dutoit, V. Pagel, N. Pierret, F. Bataille et O. V. der Vrecken** (1996). *The MBROLA project: Towards a Set of High Quality Speech Synthesizers Free of Use for Non Commercial Purposes*, Proceedings of ICSLP **3** : 1393-1396.
- F. Duvert, M. Meurs, C. Servan, F. Béchet, F. Lefèvre et R. de Mori** (2008). *Semantic composition process in a speech understanding system*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) : 5029-5032.
- W. Eckert, E. Levin et R. Pieraccini** (1997). *User modelling for spoken dialogue system evaluation*, Proceedings of the 6th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).
- K. Evanini, P. Hunter, J. Liscombe, D. Suendermann, K. Dayanidhi et R. Pieraccini** (2008). *Caller Experience: A method for evaluating dialog systems and its automatic prediction*, IEEE Spoken Language Technology Workshop : 129 -132.
- M. M. Fard, J. Pineau et P. Sun** (2008). *A Variance Analysis for POMDP Policy Evaluation*, AAAI.
- G. Ferguson et J. Allen** (1998). *TRIPS: An Intelligent Integrated Problem-Solving Assistant*, Proceedings of the 5th National Conference on Artificial Intelligence : 567-573.
- C. J. Fillmore** (1985). *Frames and the semantics of understanding*, Quaderni di Semantica, 6.2: 222-254.
- K. Forbes-Riley et D. Litman** (2011). *Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system*, Computer Speech and Language 25: 105-126.
- M. Frampton et O. Lemon** (2009). *Recent research advances in Reinforcement Learning in Spoken Dialogue Systems*, Knowledge Engineering Review 24: 375-408.
- X. Gao, B. Xiao, D. Tao et X. Li** (2010). *A survey of graph edit distance*, Pattern Analysis and Applications 13: 113-129.
- M. Gasic, F. Lefevre, F. Jurcicek, S. Keizer, F. Mairesse, B. Thomson, K. Yu et S. Young** (2009). *Back-off Action Selection in Summary Space-Based POMDP-based Dialogue Systems*, ASRU.
- M. Geist, O. Pietquin et G. Fricout** (2009). *From Supervised to Reinforcement Learning: a Kernel-based Bayesian Filtering Framework*, International Journal On Advances in Software 2: 101-116.
- K. Georgila, J. Henderson et O. Lemon** (2005). *Learning User Simulations for Information State Update Dialogue Systems*, Eurospeech.
- D. Goddeau et J. Pineau** (2000). *Fast reinforcement learning of dialog strategies*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) **2** : 1233-1236.
- H. Gonçalo Oliveira et P. Gomes** (2010). *Towards the Automatic Creation of a Wordnet from a Term-Based Lexical Network*, Proceedings of TextGraphs, Workshop on Graph-

based Methods for Natural Language Processing : 10-18.

H. Grice (1975). Cole & Morgan (Ed.), *Logic and Conversation*.

R. Grishman, C. Macleod et A. Meyers (1994). *Complex Syntax: Building a Computational Lexicon*, 15th International Conference on Computational Linguistics, (COLING), 268-272.

T. Gärtner, P. Flach et S. Wrobel (2003). *On graph kernels: Hardness results and efficient alternatives*, Conference on Learning Theory : 129-143.

E. Hansen (2008). *Sparse stochastic finite-state controllers for POMDPs*, Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence : 256-263.

J. Henderson et O. Lemon (2008). *Mixture model POMDPs for efficient handling of uncertainty in dialogue management*, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: 73-76.

J. Henderson, O. Lemon et K. Georgila (2008). *Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets*, Computational Linguistics 34 (4): 487-511.

J. Hoey, A. von Bertoldi, P. Poupart et A. Mihailidis (2007). *Assisting Persons with Dementia during Handwashing Using a Partially Observable Markov Decision Process*, Proceedings of the International Conference on Vision Systems (ICVS), Bielefeld, Germany..

J. Hoey, P. Poupart, A. von Bertoldi, T. Craig, C. Boutilier et A. Mihailidis (2009). *Automated Handwashing Assistance for Persons with Dementia Using Video and a Partially Observable Markov Decision Process*, Computer Vision and Image Understanding (CVIU).

K. S. Hone et C. Baber (1995). *Using a simulation method to predict the transaction time effects of applying alternative levels of constraint to user utterances within speech interactive dialogues*, SDS : 209-212.

K. S. Hone et R. Graham (2000). *Towards a Tool for the Subjective Assessment of Speech System Interfaces (SASSI)*, Natural Language Engineering 6: 287-305.

B. Jabaian, L. Besacier et F. Lefèvre (2010). *Investigating multiple approaches for SLU portability to a new language*, Proceedings of 11th Annual Conference of the International Speech Communication Association (Interspeech 2010).

H. Jaeger (2000). *Observable operator models for discrete stochastic time series*, Neural Computation 12: 1371-1398.

F. Jensen et T. Nielsen (2007). *Bayesian Networks and Decision Graphs*, Springer.

S. Jung, C. Lee, K. Kim, M. Jeong et G. G. Lee (2009). *Data-driven user simulation for automated evaluation of spoken dialog systems*, Computer Speech Language 23: 479-509.

D. Justice et A. Hero (2006). *A binary linear programming formulation of the graph edit distance*, Pattern Analysis and Machine Intelligence, IEEE Transactions on 28: 1200 -1214.

L. P. Kaelbling, M. L. Littman et A. W. Moore (1996). *Reinforcement Learning: A Survey*, Journal of Artificial Intelligence Research (JAIR).

S. M. Katz (1987). *Estimation of probabilities from sparse data for the language model component of a speech recogniser*, IEEE Transactions on Acoustics, Speech, and Signal Processing 35 (3): 400-401.

K. Kim, C. Lee, S. Jung et G. Lee (2008). *A frame-based probabilistic framework for spoken dialog management using dialog examples*, Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue : 120-127.

Laroche, Putois, Bretier et Bouchon-Meunier (2009). *Hybridisation of Expertise and Reinforcement Learning in Dialogue Systems*, Proceedings of 10th Annual Conference of the

- International Speech Communication Association (Interspeech 2009).
- R. Laroche** (2010). *Raisonnement sur les incertitudes et apprentissage pour les systèmes de dialogue conventionnels*, Thèse de doctorat. Université de Paris 6.
- S. Larsson** (2007). *Rapid Prototyping using Issue-based Dialogue Management in GoDiS*, Workshop on Advanced Dialogs. VoiceXML Forum Tools Committee, Advanced Dialogs Working Group..
- S. Larsson, R. Jonson, G. Amores, C. Garcia et J. Qesada** (2002). *Evaluation of Contribution of the Information State Based View of Dialogue*, SIRIDUS deliverable D3.4.
- C. Lee, S. Jung, J. Eun, M. Jeong et G. G. Lee** (2006). *A Situation-Based Dialogue Management using Dialogue Examples*, ICASSP 1 : I.
- F. Lefevre, M. Gasic, F. Jurcicek, S. Keizer, F. Mairesse, B. Thomson, K. Yu et S. Young** (2009). *k-Nearest Neighbor Monte-Carlo Control Algorithm for POMDP-based Dialogue Systems*, SIGDIAL.
- F. Lefèvre** (2007). *Dynamic Bayesian Networks and Discriminative Classifiers for Multi-Stage Semantic Interpretation*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
- F. Lefèvre, F. Mairesse et S. Young** (2010). *Cross-Lingual Spoken Language Understanding from Unaligned Data using Discriminative Classification Models and Machine Translation*, Proceedings of 11th Annual Conference of the International Speech Communication Association (Interspeech 2010).
- F. Lefèvre et R. de Mori** (2007). *Unsupervised state clustering for stochastic dialog management*, IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).
- J. Lehuen** (1997). *Un modèle de dialogue dynamique et générique intégrant l'acquisition de sa compétence linguistique*, Thèse de doctorat. Université de Caen.
- O. Lemon** (2011). *Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation*, Computer Speech and Language 25: 210 - 221.
- O. Lemon, K. Georgila, J. Henderson et M. Stuttle** (2006). *An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system*, Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL) : 119-122.
- E. Levin, R. Pieraccini et W. Eckert** (1998). *Using markov decision process for learning dialogue strategies*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
- E. Levin, R. Pieraccini et W. Eckert** (2000). *A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies*, IEEE Transactions on Speech and Audio Processing, 8 : 11-23.
- G. Linarès, P. Nocera, D. Massonié et D. Matrouf** (2007). *The LIA speech recognition system: from 10xRT to 1xRT*, Proceedings of the 10th International Conference on Text, Speech and Dialogue : 302-308.
- D. J. Litman et J. F. Allen** (1984). *A plan recognition model for clarification subdialogues*, Proceedings of the 10th International Conference on Computational Linguistics : 302-311.
- D. Luzzati** (1989). *Recherches sur le dialogue homme-machine, modèles linguistiques et traitements automatiques*, Thèse de doctorat. Paris III.

- S. Mannor, D. Simester, P. Sun et J. Tsitsiklis** (2004). *Bias and variance in value function estimation*, ACM International Conference Proceeding Series.
- M. P. Marcus, B. Santorini et M. A. Marcinkiewicz** (1993). *Building a Large Annotated Corpus of English: The Penn Treebank*, Computational Linguistics 19: 313-330.
- P. McKeivitt, D. Partridge et Y. Wilks** (1992). *Approaches to natural language discourse processing*, Artificial Intelligence Review 6.
- B. Messmer et H. Bunke** (1998). *A new algorithm for error-tolerant subgraph isomorphism detection*, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20: 493-504.
- N. Meuleau, L. Peshkin, K. eung Kim et L. P. Kaelbling** (1999). *Learning finite-state controllers for partially observable environments*, Proceedings of the 5th conference on Uncertainty in Artificial Intelligence : 427-436.
- M. Meurs** (2009). *Approche stochastique bayésienne de la composition sémantique pour les modules de compréhension automatique de la parole dans les systèmes de dialogue homme-machine*, Thèse de doctorat. Université d'Avignon et des Pays de Vaucluse.
- M. Meurs, F. Lefèvre et R. De Mori** (2009). *Spoken Language Interpretation: On the Use of Dynamic Bayesian Networks for Semantic Composition*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
- R. S. Michael Littman et S. Singh** (2002). *Predictive Representations of State*, Advances in Neural Information Processing Systems 14: 1555-1561.
- G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross et K. Miller** (1990). *WordNet: An on-line lexical database*, International Journal of Lexicography 3: 235-244.
- D. Molla** (2006). *Learning of Graph-based Question Answering Rules*, Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing : 37-44.
- R. Myers, R. C. Wilson et E. R. Hancock** (2000). *Bayesian graph edit distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence 22: 628-635.
- S. Möller et K.-P. Engelbrecht** (2008). *Towards a Perception-Based Evaluation Model for Spoken Dialogue Systems*, Proceedings of the 4th IEEE workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems : 306-309.
- S. Möller et N. G. Ward** (2008). *A framework for model-based evaluation of spoken dialog systems*, Proceedings of the 9th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 182-189.
- G. Navarro** (2001). *A guided tour to approximate string matching*, ACM Computer Survey 33: 31-88.
- M. Neuhaus et H. Bunke** (2002). *Self-organizing graph edit distance*, Michel Neuhaus : 83-94.
- C. R. Perrault et J. F. Allen** (1980). *A plan-based analysis of indirect speech acts*, Comput. Linguist. 6: 167-182.
- J. Peters et S. Schaal** (2008). *Natural Actor-Critic*, Neurocomput. 71: 1180-1190.
- J. Peters, S. Vijayakumar et S. Schaal** (2005). *Natural actor-critic*, Machine Learning: ECML 2005 : 280-291.
- M. Peterson** (2009). *An Introduction to Decision Theory*, Cambridge University Press.
- M. Phillips** (2010). *Vlingo*. http://www.research.att.com/articles/featured_stories/2010_12/201101_Vlingo_interview.html
- R. Pieraccini et J. Huerta** (2005). *Where do we go from here? Research and Commercial*

- Spoken Dialog Systems*, 6th SIGDIAL Workshop on Discourse and Dialog : 1-10.
- O. Pietquin** (2004). *A Framework for Unsupervised Learning of Dialogue Strategies*, Thèse de doctorat. Polytech de Mons.
- O. Pietquin** (2005). *A probabilistic description of man-machine spoken communication*, Proceedings of the 5th IEEE International Conference on Multimedia and Expo (ICME).
- O. Pietquin** (2006). *Consistent goal-directed user model for realistic man-machine task-oriented spoken dialogue simulation*, Proceedings of the 7th IEEE International Conference on Multimedia and Expo (ICME).
- O. Pietquin et H. Hastie** (2011). *A survey on metrics for the evaluation of user simulations*, Knowledge Engineering Review.
- F. Pinault et F. Lefèvre** (2011). *Semantic graph clustering for POMDP-based spoken dialog systems*, Proceedings of 12th Annual Conference of the International Speech Communication Association (Interspeech).
- F. Pinault et F. Lefèvre** (2011). *Unsupervised clustering of probability distributions of semantic frame graphs for POMDP-based spoken dialogue systems with summary space*, 7th Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI/KRPDS).
- F. Pinault et F. Lefèvre** (2010). *Structures de frames sémantiques pour le dialogue homme-machine par processus de décision markoviens*, Journée d'Étude de la Parole (JEP).
- F. Pinault, F. Lefèvre et R. De Mori** (2009). *Feature-based summary spaces for stochastic dialogue modeling with hierarchical semantic frames*, Proceedings of 10th Annual Conference of the International Speech Communication Association (Interspeech 2009).
- J. Pineau** (2004). *Tractable Planning Under Uncertainty: Exploiting Structure*, Thèse de doctorat. Robotics Institute, Carnegie Mellon University.
- J. Pineau, G. Gordon et S. Thrun** (2003). *Policy-contingent abstraction for robust robot control*, Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI), Acapulco, Mexico.
- J. Pineau, G. Gordon et S. Thrun** (2006). *Anytime Point-Based Approximations for Large POMDPs*, Journal of Artificial Intelligence Research (JAIR).
- P. Poupart** (2005). *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*, Thèse de doctorat. Department of Computer Science, University of Toronto, Toronto.
- P. Poupart et C. Boutilier** (2003). *Bounded finite state controllers*, Advances in neural information processing systems **16** : 823-830.
- P. Price** (1990). *Evaluation of spoken language systems: The ATIS domain*, DARPA Workshop on Speech and Natural Language.
- A. Raux et M. Eskenazi** (2004). *Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges*, Proceedings of INSTILL.
- A. B. L. A. W. B. & M. E. Raux** (2003). *LET'S GO: Improving Spoken Dialog Systems for the Elderly and Non-natives*, Eurospeech.
- R. C. Read et D. G. Corneil** (1977). *The graph isomorphism disease*, Journal of Graph Theory **1**: 339-363.
- V. Rieser** (2008). *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*, Thèse de doctorat. Saarland University, Saarbruecken Dissertations in Computational Linguistics and Language Technology.
- V. Rieser et O. Lemon** (2008). *Learning Effective Multimodal Dialogue Strategies from*

- Wizard-of-Oz data: Bootstrapping and Evaluation*, ACL.
- E. S. Ristad, A. Stolcke, R. Thomas, K. Yamanishi et A. Y. F. Their** (1995). *A Natural Law of Succession*, University of Princeton.
- A. Roque et D. Traum** (2008). *Degrees of Grounding Based on Evidence of Understanding*, SIGDIAL.
- S. Rosset et L. Lamel**. (2001). *Gestionnaire de dialogue pour un système d'informations à reconnaissance vocale*, TALN.
- A. Roulet, M. Auchlin, J. Schelling, Moeschler et C. Rubattel** (1991). Lang (Ed.), *L'articulation du discours en français contemporain*.
- N. Roy, J. Pineau et S. Thrun** (2000). *Spoken dialogue management using probabilistic reasoning*, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics : 93-100.
- M. Sadek, P. Bretier et F. Panaget** (1997). *ARTIMIS: Natural dialogue meets rational agency*, International Joint Conference on Artificial Intelligence **15** : 1030-1035.
- S. Sakr et G. Al-Naymat** (2010). *Graph indexing and querying: a review*, International Journal of Web Information Systems 6: 101-120(20).
- J. Schatzmann** (2008). *Statistical User and Error Modelling for Spoken Dialogue Systems*, Thèse de doctorat. University of Cambridge.
- J. Schatzmann, M. Stuttle, K. Weilhammer et S. Young** (2005). *Effects of the user model on simulation-based learning of dialogue strategies*, 9th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).
- J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye et S. Young** (2007a). *Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System*, HLT/NAACL.
- J. Schatzmann, B. Thomson et S. Young** (2007b). *Error simulation for training statistical dialogue systems*, 11th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2007) : 526-531.
- J. Schatzmann, K. Weilhammer, M. Stuttle et S. Young**. (2006). *A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies*, Knowledge Engineering Review.
- K. Scheffler et S. Young** (2000). *Probabilistic simulation of human-machine dialogues*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
- K. Scheffler et S. Young** (2001). *Corpus-based Dialogue Simulation for Automatic Strategy Learning and Evaluation.*, NAACL Workshop on Adaptation in Dialogue Systems.
- J. Searle** (1969). *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press.
- E. Shriberg, J. Ang, R. Dhillon, S. Bhagat et H. Carvey** (2004). *The ICSI meeting recorder dialog act (MRDA) corpus*, SIGDIAL.
- Sigaud et Buffet** (2008). Lavoisier (Ed.), *Processus décisionnels de Markov en intelligence artificielle*, Hermes.
- D. Silvestrov** (2006). *Limit theorems for randomly stopped stochastic processes*, Journal of Mathematical Sciences 138: 5467-5471.
- H. Sim, K. Kim, J. Kim, D. Chang et M. Koo** (2008). *Symbolic heuristic search value iteration for factored POMDPs*, National conference on Artificial intelligence **2**.
- S. Singh, M. R. James et M. R. Rudary** (2004). *Predictive state representations: a new*

theory for modeling dynamical systems, AUAI.

A. Skorokhod (1956). *Limit theorems for stochastic processes.*, Theory Probability Application 1: 261–290.

M. Spaan et N. Vlassis (2005). *Perseus: Randomized point-based value iteration for POMDPs*, Journal of Artificial Intelligence Research (JAIR).

S. B. M. S. H. & A. S. Stenchikova (2007). *RavenCalendar: A Multimodal Dialog System for Managing a Personal Calendar*, HLT-NAACL.

M. Stuttle, J. Williams et S. Young (2004). *A Framework for Dialog Systems Data Collection using a Simulated ASR Channel*, ICSLP.

D. Suendermann, J. Liscombe, K. Dayanidhi et R. Pieraccini (2009). *A Handsome Set of Metrics to Measure Utterance Classification Performance in Spoken Dialog Systems*, Proceedings of the 10th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 349-356.

R. Sutton et A. Barto (1998). *Reinforcement Learning: An Introduction*, MIT Press.

B. Thomson (2009). *Statistical methods for spoken dialogue management*, Thèse de doctorat. University of Cambridge.

B. Thomson, J. Schatzmann et S. Young (2008). *Bayesian Update of Dialogue State for Robust Dialogue Systems*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

B. Thomson et S. Young (2010). *Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems*, Computer Speech and Language (CSL) **24** : 562-588.

S. Thrun (2000). *Monte Carlo POMDPs*, Advances in Neural Information Processing Systems **12** : 1064-1070.

D. Traum (1994). *A computational theory of grounding in natural language conversation*, Thèse de doctorat. University of Rochester.

D. Traum et S. Larsson (2003). *The information state approach to dialogue management*, Current and New Directions in Discourse and Dialogue : 325-353.

D. R. Traum (2000). *20 Questions for Dialogue Act Taxonomies*, Journal of Semantics **17** : 7-30.

A. Turing (1950). *Computing machinery and intelligence*, Mind 59: 433-460.

S. Vargas, S. Quarteroni, G. Riccardi et A. Ivanov (2010). *Investigating Clarification Strategies in a Hybrid POMDP Dialog Manager*, Proceedings of the 11th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL).

S. Vargas, G. Riccardi et S. Quarteroni (2008). *Persistent information state in a data-centric architecture*, Proceedings of the 9th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 68-71.

S. Vargas, F. Weng et H. Pon-Barry (2006). *Interactive question answering and constraint relaxation in spoken dialogue systems*, Proceedings of the 7th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL) : 28-35.

VoiceXML (2004). Spécification W3C. <http://www.w3.org/TR/VoiceXML20>.

M. Walker (2000). *An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email*, Journal of Artificial Intelligence Research (JAIR) **12**: pp. 387-416.

M. Walker, J. Fromer et S. Narayanan (1998). *Learning Optimal Dialogue Strategies: A Case Study of a Spoken Dialogue Agent for Email*, 36th Annual Meeting of the Association

for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING).

M. Walker, C. Kamm et D. Litman (2000). *Towards developing general models of usability with PARADISE*, Natural Language Engineering 6: 363-377.

M. Walker, D. J. Litman, C. A. Kamm et A. Abella (1997). *PARADISE: a framework for evaluating spoken dialogue agents*, Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics : 271-280.

M. Walker et R. J. Passonneau (2001). *DATE: A Dialogue Act Tagging Scheme for Evaluation of Spoken Dialogue Systems*, Human Language Technology Conference.

C. Wang, S. Joshi et R. Khardon (2007). *First order decision diagrams for relational MDPs*, International Joint Conferences on Artificial Intelligence (IJCAI).

K. Wang (2002). *SALT: An XML Application for Web-based Multimodal Dialog*, Second NLP and XML Workshop.

Y.-K. Wang, K.-C. Fan et J.-T. Horng (1997). *Genetic-based search for error-correcting graph isomorphism*, IEEE Transactions on Systems, Man, and Cybernetics 27: 588 -597.

T. Watanabe, M. Araki et S. Doshita (1998). *Evaluating Dialogue Strategies under Communication Errors Using Computer-to-Computer Simulation*, IEICE transactions on information and systems 81: 1025-1033.

X. Wei et A. Rudnicky (1999). *An agenda-based dialog management architecture for spoken language system*, IEEE Automatic Speech Recognition and Understanding Workshop : 337-340.

J. Weizenbaum (1966). *ELIZA : a computer program for the study of natural language communication between man and machine*, Association for Computing Machinery (ACM).

J. Williams (2008b). *The best of both worlds: Unifying conventional dialog systems and POMDPs*, Proceedings of 9th Annual Conference of the International Speech Communication Association (Interspeech 2008).

J. Williams, P. Poupart et S. Young (2005). *Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management*, Proceedings of the 6th Special Interest Group on Discourse and Dialogue Conference (SIGDIAL).

J. Williams et S. Young (2005). *Scaling up POMDPs for Dialog Management: The "Summary POMDP" Method*, 9th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2005) : 177 -182.

J. Williams et S. Young (2006). *Scaling POMDPs for dialog management with composite summary point-based value iteration (CSPBVI)*, AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems.

J. D. Williams (2008a). *Integrating expert knowledge into POMDP optimization for spoken dialog systems*, Association for the Advancement of Artificial Intelligence (AAAI).

I. H. Witten et T. C. Bell (1991). *The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression*, IEEE Transactions on Information Theory 37: 1085-1094.

S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson et K. Yu (2010). *The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management*, Computer Speech and Language (CSL) 24 : 150-174.

S. Young, J. Schatzmann, K. Weilhammer et H. Ye (2007). *The Hidden Information State Approach to Dialog Management*, Proceedings of IEEE International Conference on

Acoustics, Speech, and Signal Processing (ICASSP).

B. Zhang, Q. Cai, J. Mao et B. Guo (2001). *Planning and acting under uncertainty: A new model for spoken dialogue system*, Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence.

Proceedings of TextGraphs, workshop on Graph Based Methods for Natural Language Processing (2006). Rada Mihalcea and Dragomir Radev (Ed.), Association for Computational Linguistics.