



**HAL**  
open science

## Composition sémantique pour la langue orale

Frédéric Duvert

► **To cite this version:**

Frédéric Duvert. Composition sémantique pour la langue orale. Autre [cs.OH]. Université d'Avignon, 2010. Français. NNT : 2010AVIG0174 . tel-00609520

**HAL Id: tel-00609520**

**<https://theses.hal.science/tel-00609520v1>**

Submitted on 19 Jul 2011

**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.



ACADEMIE D'AIX-MARSEILLE  
UNIVERSITE D'AVIGNON ET DES PAYS DE VAUCLUSE

---

## THESE

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

**SPECIALITE : Informatique**

École Doctorale 536 « Sciences et Agrosociences »  
Laboratoire d'Informatique (EA 931)

### *Composition sémantique pour la langue orale*

par

**Frédéric Duvert**

sous la direction de

**Messieurs les Professeurs Rénato de Mori & Frédéric Béchet**

**Soutenue publiquement le 10 novembre 2010 devant un jury composé de :**

Mr. Stefano A. Cerri	Professeur, LIRMM, Montpellier	Rapporteur
Mr. Denis Jouvét	Professeur, LORIA, Nancy	Rapporteur
Mme Géraldine Damnat	Docteur, Orange Labs, Lannion	Examineur
Mr. Marc El-Bèze	Professeur, LIA, Avignon	Président du jury
Mr. Frédéric Béchet	Professeur, LIF, Marseille	Directeur de thèse
Mr. Renato de Mori	Professeur, LIA, Avignon	Directeur de thèse

---

Laboratoire d'Informatique d'Avignon



École Doctorale 536  
« Sciences et Agrosociences »



# Remerciements

A Messieurs et Madame, les Membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail, avec l'expression de tous mes respectueux remerciements.

A Monsieur le Professeur Renato De Mori, à qui je tiens à dire ici toute ma profonde reconnaissance pour avoir guidé mes recherches, de ses conseils éclairés, pour sa constante gentillesse, aussi et surtout pour son soutien moral et sa compréhension lors des problèmes personnels qui sont survenus pendant la préparation de ce doctorat.

A Monsieur Le Professeur Frédéric Béchet, qu'il soit assuré de ma sincère gratitude pour m'avoir prodigué ses précieux conseils.

A mes chers amis du laboratoire du LIA, qui ont été présents dans les 'coups durs', qui m'ont constamment entouré de leur présence et de la chaleur de leur amitié. Puissent-ils trouver là l'expression de mes plus sincères et profonds remerciements.

A Mesdames les Secrétaires et toutes les personnes du labo, pour leur gentillesse, pour leur accueil, pour leur sourire, pour leurs mots de réconfort. Sans oublier Eric, pour son petit café qui aide à tenir.

A mes parents et grands-parents et Audrey, qui m'ont soutenu et m'ont permis de m'amener jusqu'à ce doctorat et que j'aime très fort.

A vous tous qui avez partagé, ces années à mes côtés. Toute ma reconnaissance et mes sincères remerciements...



# Résumé

La thèse présentée ici a pour but de proposer des systèmes de détection, de composition de constituants sémantiques et d'interprétation dans la compréhension de la langue naturelle parlée. Cette compréhension se base sur un système de reconnaissance automatique de la parole qui traduit les signaux oraux en énoncés utilisables par la machine.

Le signal de la parole, ainsi transcrit, comporte un ensemble d'erreurs liées aux erreurs de reconnaissance (bruits, parasites, mauvaise prononciation...). L'interprétation de cet énoncé est d'autant plus difficile qu'il est issu d'un discours parlé, soumis à la disfluence du discours, aux auto-corrrections... L'énoncé est de plus agrammatical, car le discours parlé lui-même est agrammatical.

L'application de méthodes d'analyses grammaticales ne produit pas de bons résultats d'interprétation, sur des textes issus de transcriptions de la parole. L'utilisation de méthodes d'analyses syntaxiques profondes est à éviter. De ce fait, une analyse superficielle est envisagée.

Un des premiers objectifs est de proposer une représentation du sens. Il s'agit de considérer des ontologies afin de conceptualiser le monde que l'on décrit. On peut exprimer les composants sémantiques en logique du premier ordre avec des prédicats. Dans les travaux décrits ici, nous représentons les éléments sémantiques par des frames (FrameNet). Les structures de frames sont hiérarchisées, et sont des fragments de connaissances auxquels on peut insérer, fusionner ou inférer d'autres fragments de connaissances. Les structures de frames sont dérivables en formules logiques.

Nous proposons un système de compréhension de la parole à partir de règles logiques avec le support d'une ontologie, afin de pouvoir créer des liens à partir de composants sémantiques. Puis, nous avons mené une étude sur la découverte des supports syntaxiques des relations sémantiques. Nous proposons une expérience de composition sémantique afin d'enrichir les composants sémantiques de base. Enfin, nous présentons un système de détection de lambda-expression pour mettre en hypothèse les relations à trouver à travers le discours.



# Abstract

The thesis presented here is intended to provide detection systems, composition of components and semantic interpretation in the natural spoken language understanding. This understanding is based on an automatic speech recognition system that translates the signals into oral statements used by the machine.

The transcribed speech signal, contains a series of errors related to recognition errors (noise, poor pronunciation...). The interpretation of this statement is difficult because it is derived from a spoken discourse, subject to the disfluency of speech, for self-correction... The statement is more ungrammatical, because the spoken discourse itself is ungrammatical.

The application of grammatical analysis methods do not produce good results interpretation, on the outcome of speech transcription. The use of deep syntactic analysis methods should be avoided. Thus, a superficial analysis is considered.

A primary objective is to provide a representation of meaning. It is considered ontologies to conceptualize the world we describe. We can express the semantic components in first order logic with predicates. In the work described here, we represent the semantic elements by frames (FrameNet). The frames are hierarchical structures, and are fragments of knowledge which can be inserted, merge or infer other fragments of knowledge. The frames are differentiable structures in logical formulas.

We propose a system for speech understanding from logical rules with the support of an ontology in order to create links from semantic components. Then, we conducted a study on the discovery supports syntactic semantic relationships. We propose a compositional semantics experience to enrich the basic semantic components. Finally, we present a detection system for lambda-expression hypothesis to find the relationship through discourse.





# Table des matières

<b>Résumé</b>	<b>5</b>
<b>Abstract</b>	<b>7</b>
<b>Introduction</b>	<b>13</b>
<b>I Motivation et contexte</b>	<b>17</b>
<b>1 Représentation du sens dans la Compréhension du Langage Naturel</b>	<b>19</b>
1.1 Premières tentatives d'utilisation des méthodes d'interprétation de textes pour la Compréhension du Langage Parlé . . . . .	20
1.2 Représentation du sens . . . . .	21
1.3 Extraction du sens dans les phrases . . . . .	25
<b>2 Interprétation basée sur l'analyse syntaxique complète</b>	<b>31</b>
2.1 Grammaires à réseaux de transition . . . . .	32
2.2 Grammaires et analyseurs utilisés pour la Compréhension du Langage Parlé dans le projet Air Travel Information System . . . . .	35
2.2.1 Le système DELPHI . . . . .	35
2.2.2 Le système Gemini et les système basés sur la "concordance de modèles" (Template Matcher) . . . . .	37
2.3 Grammaires Lexicalisées . . . . .	41
2.4 Grammaires stochastiques pour l'interprétation . . . . .	49
2.4.1 Le système TINA . . . . .	49
2.4.2 Le système HUM . . . . .	51
2.5 Le modèle à vecteurs d'états cachés de Cambridge . . . . .	52
2.6 Conclusions . . . . .	54
<b>3 Interprétation basée sur l'analyse syntaxique partielle</b>	<b>55</b>
3.1 Les analyseurs partiels . . . . .	56
3.2 Les modèles génératifs . . . . .	58
3.2.1 Les modèles à états finis . . . . .	58
3.2.2 Interprétation avec des transducteurs à états finis stochastiques . . . . .	60
3.2.3 Les Réseaux Bayésiens Dynamiques . . . . .	65

3.3	Les modèles basés sur un apprentissage discriminant . . . . .	68
3.3.1	Les arbres de classification sémantique . . . . .	68
3.3.2	Les Machines à Vecteurs de Support . . . . .	71
3.3.3	Utilisation de la connaissance préalable . . . . .	72
3.4	Modélisation log-linéaire . . . . .	75
3.4.1	Fonctions caractéristiques . . . . .	75
3.4.2	Les “Conditional Random Fields ( C.R.F. )” . . . . .	76
3.4.3	Normalisation du niveau de position . . . . .	77
3.4.4	L’entraînement . . . . .	77
3.4.5	Processus de décision . . . . .	78
3.5	Utilisation des Conditional Random Fields , pour l’analyse superficielle	78
3.5.1	Segmentation avec les Conditional Random Fields . . . . .	80
3.6	Conclusions . . . . .	81
<b>4</b>	<b>Systèmes avancés</b>	<b>83</b>
4.1	Méthodes récentes . . . . .	84
4.1.1	Approche de traduction automatique . . . . .	84
4.1.2	Quelques applications récentes . . . . .	87
4.2	Architectures avancées . . . . .	90
4.2.1	Le système de Microsoft . . . . .	90
4.2.2	Le système d’AT & T . . . . .	94
4.2.3	Le système d’IBM . . . . .	97
4.2.4	Le projet Européen LUNA . . . . .	100
<b>II</b>	<b>Compréhension de la parole et compositions sémantiques</b>	<b>103</b>
<b>5</b>	<b>Composition sémantique</b>	<b>105</b>
5.1	Connaissance fragmentée et composition sémantique . . . . .	106
5.1.1	Etiquettes “concepts” et connaissance fragmentée . . . . .	106
5.1.2	Composition par fusion de fragments . . . . .	110
5.1.3	Composition par attachements . . . . .	111
5.1.4	Composition par attachements et inférences . . . . .	112
5.1.5	Probabilités de frames . . . . .	113
<b>6</b>	<b>Contribution à la compréhension automatique de la parole</b>	<b>115</b>
6.1	Composition par règles . . . . .	116
6.1.1	Le corpus MEDIA et la génération des constituants de base . . . . .	117
6.1.2	Composer des relations sémantiques dans les structures . . . . .	118
6.1.3	Annotation progressive du corpus par des structures sémantiques	119
6.1.4	Résultats préliminaires . . . . .	121
6.1.5	L’outil d’annotation et de composition . . . . .	122
6.1.6	Utilisation de fragments de frames et inférences . . . . .	125
6.2	Acquisition progressive des caractéristiques syntaxiques supportant les relations sémantiques . . . . .	127
6.2.1	Annotation des relations binaires . . . . .	129

6.2.2	Processus de composition sémantique . . . . .	131
6.3	Expérience de composition sémantique . . . . .	133
6.3.1	Résultats . . . . .	135
6.3.2	Désambiguïsation sémantique . . . . .	137
6.3.3	Exemple de désambiguïsation d'un tour de dialogue . . . . .	138
6.4	Détection de lambda-expressions . . . . .	141
6.4.1	Calcul des probabilités . . . . .	147
6.4.2	Les fonctions caractéristiques . . . . .	148
6.4.3	Expérimentations . . . . .	149
<b>Conclusions et Perspectives</b>		<b>157</b>
6.5	Conclusions . . . . .	157
6.6	Perspectives . . . . .	159
<b>Liste des illustrations</b>		<b>161</b>
<b>Liste des tableaux</b>		<b>163</b>
<b>Bibliographie</b>		<b>165</b>
<b>Publications Personnelles</b>		<b>183</b>
<b>III Appendices</b>		<b>185</b>
<b>A Corpus utilisé et Ontologie du domaine</b>		<b>187</b>
A.1	Le Corpus <i>MEDIA</i> . . . . .	188
A.1.1	Dictionnaire sémantique <i>MEDIA</i> . . . . .	189
A.1.2	Annotation . . . . .	190
A.2	Ontologie du domaine lié à <i>MEDIA</i> . . . . .	191
<b>B Protocole d'évaluation</b>		<b>195</b>



# Introduction

La compréhension du langage parlé ( C.L.P. ) est l'interprétation de signes portés par le signal de parole. L'épistémologie est la science de la connaissance utilisée pour l'interprétation. Elle considère une donnée comme une unité de base. Une donnée peut être un objet, une action ou un événement dans le monde considéré et peut avoir des coordonnées spatio-temporelles, des aspects multiples et des qualités qui la rendent différentes des autres. Une donnée peut être représentée par une image ou être résumée et représentée par un concept. Un concept peut être empirique, structurel ou juste un a priori. Il peut exister des relations parmi les données.

Le langage naturel fait référence à des données dans le monde et à leurs relations. Les phrases dans le langage naturel sont des séquences de mots appartenant à un lexique de mots du monde. Les mots d'une séquence associent une ou plusieurs données conceptualisées appelées aussi *significations* qui peuvent être sélectionnées et composées pour former le sens de la phrase. Des phrases correctes dans un langage satisfont des contraintes décrites par la syntaxe et la grammaire de la langue.

La sémantique traite de l'organisation des significations et de leurs relations dans le monde des signes observables. La conceptualisation humaine du monde n'est pas très bien comprise et il n'est pas évident de bien décrire les phénomènes d'organisation de la pensée humaine. Néanmoins, de bons modèles sémantiques ont été proposés, en supposant que des constituants sémantiques basiques soient organisés au sein de structures conceptuelles. Dans [Jackendoff \(2002\)](#), il est suggéré que la sémantique est un système génératif indépendant corrélé avec une syntaxe à travers une interface.

La sémantique computationnelle effectue une conceptualisation du monde en utilisant des procédés informatiques, pour composer une structure de représentation du sens à partir de signes disponibles et de leurs caractéristiques, par exemple à partir des mots et des phrases.

La compréhension du langage parlé est une tâche difficile car les signes exprimant du sens sont codés dans le signal avec d'autres informations, comme l'identité du locuteur et l'environnement acoustique. Les phrases en langage naturel sont souvent diffi-

ciles à analyser. En outre, des messages oraux sont souvent incorrects et peuvent contenir des disfluences, tels : les interruptions, les auto-corrrections et d'autres événements.

La conception d'un système de compréhension automatique de la parole doit être basée sur un processus implémentant une stratégie d'interprétation, qui utilise des modèles calculatoires pour divers types de connaissances. Le processus doit prendre en compte deux situations : d'une part, les modèles informatiques sont imparfaits ; et d'autre part, les transcriptions automatiques des énonciations des utilisateurs réalisées par la Reconnaissance Automatique de la Parole ( [R.A.P.](#) ) sont sujettes aux erreurs.

La compréhension automatique du langage parlé est effectuée par des procédés qui utilisent des connaissances pour transformer du langage parlé dans un langage de représentation du sens. Comme certains éléments de cette représentation peuvent être implicites dans ce qui est observé, l'interprétation peut également impliquer l'inférence. Le résultat de l'interprétation peut contenir des structures sémantiques complexes obtenues en composant des fragments de l'hypothèse des relations syntaxiques locales des données observables. Les constituants sémantiques et les structures sémantiques peuvent être exprimés dans un langage de représentation sémantique qui décrit les modèles logiques et les relations.

Historiquement, les premiers systèmes d'interprétation utilisaient des méthodes de compréhension du langage naturel et des modèles pour le traitement d'une séquence d'hypothèses de mots générée par un module de [R.A.P.](#) avec des méthodes et des modèles non probabilistes.

Les premières tentatives et les expériences ont montré qu'il y a beaucoup de manières pour exprimer le même concept et que les relations entre une hypothèse d'interprétation et son support sur des motifs de données sont souvent imprécises. En outre, en particulier pour la compréhension automatique du langage parlé, les signes utilisés pour l'interprétation sont extraits du signal de parole avec un processus qui n'est pas parfait.

Différents types de modèles probabilistes ont été introduits par la suite pour tenir compte de l'imperfection des connaissances et des erreurs possibles dans les hypothèses de mots à utiliser pour les interprétations. Certaines de ces méthodes développées pour la compréhension ont également été utilisées pour l'interprétation du texte écrit.

Des méthodes d'apprentissage automatique pour la caractérisation automatique des modèles des constituants sémantiques et de leurs relations ont aussi été développées. L'évaluation des résultats obtenus jusqu'ici montrent que l'interprétation de la sortie de la [R.A.P.](#) est affectée par plusieurs types d'erreurs. Certaines erreurs peuvent être évitées en utilisant des classifieurs pour le traitement des N-meilleures listes ou des treillis d'hypothèses.

De nouveaux progrès peuvent venir probablement de l'amélioration des méthodes

d'apprentissage, capables de remplir la généralisation effective des connaissances observées dans un nombre non infini d'exemples. En particulier, il semble intéressant d'inclure les relations entre les connaissances, dans le processus d'apprentissage. Il est possible d'apprendre de cette manière les liens sémantiques étant donné les relations syntaxiques exprimées sur des dépendances syntaxiques de longue distance.

Dans la pratique, il est utile de développer des systèmes de Compréhension du Langage Parlé ( C.L.P. ) qui effectuent une automatisation partielle lorsque des interprétations fiables ont été obtenues. Une interprétation n'est acceptée comme fiable que si le système a une grande confiance sur son hypothétisation. Cela exige l'utilisation de mesures de confiance et de méthodes d'apprentissage efficaces. Elles sont passés en revue avec l'état de l'art en C.L.P. dans De Mori et al. (2008).

Plusieurs raisons nous motivent à utiliser un système d'analyse partielle des données d'entrées de bas en haut. L'analyse syntaxique complète des données a un certain nombre de problèmes. Pour assurer une couverture de la langue parlée, des grammaires stochastiques hors-contexte ont été proposées, elles sont décrites dans le chapitre 2. Le calcul des probabilités des arbres syntaxiques générés par ces grammaires, est fait par des modèles qui ne caractérisent pas de façon adéquate les dépendances entre composants syntaxiques distants. L'effort pour produire la grammaire nécessaire ainsi que l'effort d'annotation requis pour l'analyseur syntaxique complet sont plus importants que pour une analyse superficielle. L'analyse superficielle utilise certains procédés comme des étiqueteurs, des segmenteurs, qui sont rapides, peu coûteux en effort d'annotations.

L'analyse, l'interprétation, la compréhension de la parole sont des tâches plus complexes et difficiles que des tâches portant sur le texte écrit. La difficulté provient en partie des erreurs dues au système de Reconnaissance Automatique de la Parole , qui est imparfait, à la disfluence du discours spontané, aux hésitations, erreurs, corrections humaines, etc.

Au travers de cette thèse, nous allons nous efforcer d'apporter quelques contributions de caractérisation de segments de paroles qui expriment des fragments de connaissances sémantiques et à leurs compositions.

Nous présentons nos différents travaux portant sur la composition sémantique. Notre étude a porté, dans un premier temps sur un prototype de composition sémantique à partir de règles logiques et d'une ontologie. Le système effectue la composition sur les fragments de connaissance et des constituants sémantiques basiques. Nous avons réalisé ce prototype en nous servant du corpus disponible MEDIA .

Nous avons ensuite essayé de trouver le moyen d'apprendre automatiquement les supports des relations sémantiques, dans la transcription de la parole. Cet apprentissage automatique permet de caractériser les mots de liaisons, les supports entre les constituants sémantiques dans la transcription de la parole sans utiliser de grammaires du langage.

Nous avons expérimenté la composition sémantique en utilisant notre prototype comme moyen d'enrichir l'annotation de base des constituants sémantiques. Ces améliorations permettent à des systèmes d'apprentissage, d'avoir des éléments supplémen-



taires dans la prise de décisions sur les éventuelles relations sémantiques à effectuer.

Enfin nous avons produit une méthode pour résoudre la détection des références dans le discours. La difficulté est de trouver dans le discours des éléments qui permettent de relier des constituants entre eux, à travers l'historique du dialogue. Notre détection des supports de références dans un tour de dialogue s'appuie sur des règles logiques combinées à un apprentissage.

Nous allons dans une première partie passer en revue, les deux types d'analyses, profondes et superficielles. Une représentation du sens sera mise en place dans le chapitre 1, pour pouvoir ensuite présenter dans le chapitre 2, les analyseurs syntaxiques profonds ainsi que les grammaires associées et des systèmes qui les utilisent. Le chapitre 3 étudie les analyseurs de surface et présente les systèmes se servant des techniques d'analyses superficielles et partielles. Puis une description des systèmes avancés dans le chapitre 4, présentera les principaux systèmes actuels ou historiquement récents qui permettent de modéliser et interpréter une phrase issue de la Reconnaissance Automatique de la Parole .

Dans une deuxième partie, nous introduirons le formalisme de la composition sémantique et des frames dans le chapitre 5, qui sera utilisé pour la représentation sémantique du contenu de phrases. Puis nous présentons, dans le chapitre 6, les expressions de détection et composition des fragments sémantiques qui peuvent contenir des lambda-expressions. Les résultats ont été obtenus avec le corpus MEDIA décrit dans l'appendice A. Les mesures utilisés dans l'évaluation sont décrites dans l'appendice B.

Ces travaux sont supportés par le 6<sup>ème</sup> Programme de Recherche de l'Union Européenne, Projet LUNA, IST, contrat : 33549. Pour plus d'informations sur le Projet LUNA, voir [LUNA \(2009\)](#).

**Première partie**

**Motivation et contexte**



# Chapitre 1

## Représentation du sens dans la Compréhension du Langage Naturel

### Sommaire

---

1.1	Premières tentatives d'utilisation des méthodes d'interprétation de textes pour la Compréhension du Langage Parlé . . . . .	20
1.2	Représentation du sens . . . . .	21
1.3	Extraction du sens dans les phrases . . . . .	25

---

## 1.1 Premières tentatives d'utilisation des méthodes d'interprétation de textes pour la Compréhension du Langage Parlé

Des idées basiques pour la représentation du sens ont été appliquées dans les premiers systèmes de Compréhension du Langage Parlé ( C.L.P. ). Un effort considérable dans la recherche de la C.L.P. a été fait avec un projet ARPA en 1971. Le projet, revu dans Klatt (1977), a suivi principalement une approche d'Intelligence Artificielle (IA) pour la compréhension de la langue naturelle. Des hypothèses de mots, générées par un système de R.A.P. , sont transformées dans une représentation sémantique, en utilisant des méthodes similaires, a fortiori égales à celles employées pour l'interprétation de textes et schématisées par 1.1.

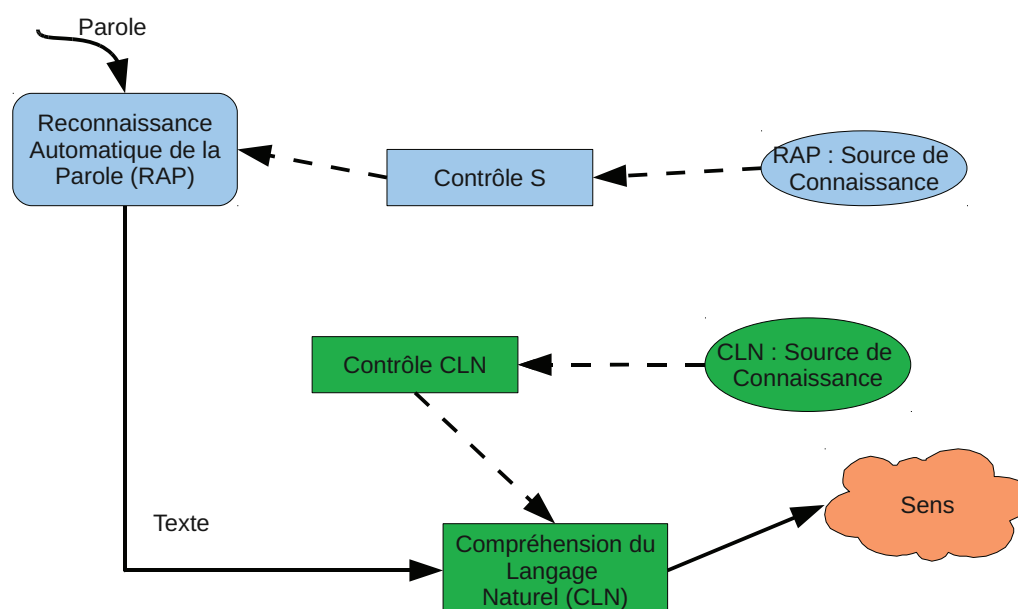


FIGURE 1.1 – Schéma des architectures des précédents systèmes de C.L.P.

Un système de transcription automatique de la parole met en oeuvre une stratégie de décodage, indiqué comme "contrôle S", sur une base acoustique, lexicale et les sources de connaissances de la langue, indiquées comme R.A.P. Source de Connaissance. L'interprétation est effectuée avec les mêmes procédés que pour le texte écrit par une stratégie de contrôle de compréhension du langage naturel en utilisant la syntaxe et les sources de connaissance sémantique indiquées comme C.L.P. Source de Connaissance, pour produire des hypothèses sur le sens du signal de parole analysé.

## 1.2 Représentation du sens

L'épistémologie informatique traite de la représentation des connaissances sémantiques dans un ordinateur. Une donnée est représentée par un objet. Les objets sont regroupés en classes selon leurs propriétés. Les classes sont organisées en *hiérarchies* communément appelés "ontologies". Ainsi, un objet est une *instance* d'une classe. Le jugement est exprimé par des *prédicats* qui décrivent les relations entre les classes. Les prédicats sont des arguments représentés par des variables dont les valeurs sont des instances de classes spécifiées et peuvent satisfaire à d'autres contraintes.

La sémantique informatique effectue une conceptualisation du monde en utilisant des éléments bien définis des langages de programmation. Les langages de programmation ont leur propre syntaxe et leur sémantique. La syntaxe définit la programmation des déclarations autorisées, la sémantique précise les opérations qu'une machine effectue lorsqu'une instruction syntaxiquement correcte est exécutée. L'analyse sémantique d'un programme d'ordinateur est basée sur les méthodes formelles et est exécutée pour la compréhension du comportement d'un programme et sa cohérence avec les concepts et ses objectifs. L'utilisation des méthodes de la logique formelle de la sémantique de l'ordinateur a été prise en considération pour l'interprétation automatique du langage naturel.

Même si les énoncés, en général, apportent du sens qui ne peut pas être exprimé en logique formelle [Jackendoff \(2002\)](#), des méthodes inspirées de l'analyse des programmes ont été considérées pour représenter la sémantique du langage naturel dans plusieurs domaines d'applications. Ces approches et limitations ont été discutées dans [Jackendoff \(1990\)](#); [Levesque et Brachman \(1985\)](#); [Woods \(1975\)](#). Un formalisme logique pour une interprétation du langage naturel devrait être en mesure de représenter une intension (l'essence d'un concept) et de l'extension (l'ensemble de tous les objets qui sont des instances d'un même concept).

Le formalisme doit également permettre, d'effectuer des inférences. Les connaissances sémantiques d'une application sont une base de connaissances (BC). Un fragment de la connaissance est décrit par une formule logique  $F$ . Le problème fondamental de l'inférence est de déterminer si,  $BC \models F$  :  $BC$  implique  $F$ ,  $F$  est vrai dans toutes les affectations possibles des variables (univers) pour lesquels  $BC$  est vrai. Si  $BC$  ne contient que des formules logiques de premier ordre, l'inférence peut être effectuée en apportant la preuve par le théorème.

Des prédicats expriment des relations pour la composition des objets dans une structure sémantique qui a un sens précis, plus riche que l'ensemble des sens des composants pris séparément. Souvent, la composition doit satisfaire des contraintes spécifiques. Par exemple, une date est une composition de mois et de chiffres qui doivent prendre des valeurs dans les relations spécifiques et des intervalles. Des structures prototypiques contiennent des variables non bornées et sont représentées par des classes

qui sont instanciées dans des objets et relient toutes les variables.

Des relations sémantiques d'une BC peuvent être représentées graphiquement par un réseau sémantique, dans lequel les relations sont représentées par des liens entre les nœuds représentant les entités décrites par les classes. Un fait avéré peut également être représenté par un nœud dans un réseau sémantique. Les liens peuvent être affirmés et correspondent à des prédicats vrais pour le nœud. Il peut y avoir des *liens structurels*. Une discussion sur ce que peut exprimer un lien est présentée dans [Woods \(1975\)](#).

Une structure sémantique, correspondant à une partie d'un réseau sémantique, peut être représentée par un schéma appelé *frame*. Une frame a une tête identifiant d'une entité structurée et des emplacements. Chaque emplacement correspond à un prédicat décrivant une relation liant l'entité représentée par la frame et une propriété. Les emplacements sont remplis par des "*fillers*" qui lient les variables "prédicat" à leur valeur. Ces valeurs peuvent être obtenues par des *fonctions* liées aux emplacements. Les contraintes de remplissage sont exprimées par des *facettes*. Des procédures pour remplir les emplacements en respectant les contraintes exprimées par les facettes, peuvent être attachées aux emplacements. Un emplacement peut être rempli par un objet qui est l'instance d'une autre frame, dont les emplacements eux-mêmes peuvent être remplis par d'autres instances de frames. Ainsi, des instances de structure de frames sont obtenues par composition. Une *instance* d'une structure de frames est obtenue en remplissant un, plusieurs ou la totalité des emplacements de la frame dans la structure.

Des structures de frames acceptables pour les représentations sémantiques dans un domaine sont générées par des grammaires de frames. Ces grammaires sont des modèles génératifs pour les Langages de Représentations Sémantiques ( L.R.S. ). Les fondations théoriques des langages basés sur les frames sont présentées dans [Kifer et al. \(1995\)](#).

Les structures de frames ont été proposées dans [Fikes et Nilsson \(1971\)](#) pour représenter des **fragments de connaissance** du monde réel, à l'opposé des connaissances lexicalisées basées sur les mots et phrases complètes. La frame suivante est un exemple simple représentant un prototype de connaissance pour la structure sémantique d'une adresse. Des facettes exprimant les contraintes pour le remplissage des emplacements, sont entre crochets dans l'exemple suivant.

```
{adresse
  localité           [VILLE]
  zone               [DEPARTEMENT ou PROVINCE ou ETAT]
  pays              [PAYS]
  rue                [NOMBRE et NOM]
  code_postal       [NOMBRE]
}
```

Une instance d'une structure d'adresse est représentée par la frame suivante.

```
{a0001
  instance_de      adresse
  localité        Avignon
  zone            Vaucluse
  pays            France
  rue             1 rue Descartes
  code_postal     84000
}
```

Ici "a0001" est un référent qui représente une instance de la classe "adresse" spécifiant la valeur du premier emplacement. Les autres emplacements définissent la liste des propriétés de cette instance particulière.

La frame ci-dessus peut être dérivée selon Nilsson (1980), après skolémisation<sup>1</sup>, dans la formule logique suivante :

$$\left( \begin{array}{l} \text{est\_un}(x, \text{adresse}) \wedge \\ \text{localite}(x, \text{Avignon}) \wedge \\ \text{zone}(x, \text{Vaucluse}) \wedge \\ \text{pays}(x, \text{France}) \wedge \\ \text{rue}(x, \text{1 rue Descartes}) \wedge \\ \text{code\_postal}(x, 84000) \end{array} \right) \quad (1.1)$$

Des frames décrivant une classe et une instance de celle-ci sont similaires mais leur sémantique est différente.

Les verbes sont des composants fondamentaux des phrases dans le langage naturel. Ils représentent des actions pour lesquelles les différentes entités jouent différents rôles. Un cas est le nom d'un rôle particulier qu'une phrase nominale ou une autre structure syntaxique prend dans l'état ou l'activité exprimée par le verbe. Il existe une structure de cas pour chaque verbe principal. Un modèle de frame pour un verbe a été proposé dans Fillmore (1968). Parmi les cas, pour un verbe, on devrait envisager l'Agent (A), l'Object (O), le Lieu (L) et autres.

Les significations possibles pour un verbe, sont représentées par des frames dans l'exemple suivant :

```
{accepter
  est_un      :   verb
  A          :   [humain...]
  O          :   [thème...]
```

1. En logique mathématique, la skolémisation d'une formule du calcul des prédicats est une transformation de cette formule, qui consiste à éliminer toutes les occurrences de quantificateur existentiel en utilisant de nouveaux symboles de fonction (un par quantification existentielle), ce de façon à conserver la satisfaisabilité de la formule.



```
.....
Autres rôles : [.....]
}
```

Une frame est associée à un verbe dans une phrase spécifique de la manière suivante :

```
{V003
instance_de : accepter
sujet      : utilisateur
thème     : [service_004]
O         : [thème...]
.....
Autres rôles : [.....]
}
```

Il est communément admis que les relations sémantiques entre un prédicat et ses arguments sont caractérisées par des rôles sémantiques, bien qu'il n'y ait pas d'accord sur le choix des rôles d'une description générale d'une langue. Ceci et d'autres considérations utiles sont discutés dans [Lowe et al. \(1997\)](#), où une base de connaissances sémantiques lexicales y est décrite et est basée sur les annotations effectuées dans le projet `FrameNet`. Dans cette approche, un ensemble de frames est associé à chaque mot d'un lexique pour représenter tous les sens possibles du mot.

Les actions du discours sont des instances des actes illocutoires ou actes du langage, identifiés comme des verbes. Ces actes du langage déterminent le type de phrase. Diverses tentatives ont été faites pour identifier les actes du langage qui sont indépendants du domaine. Une possible taxonomie de ces actes du langage est formulée dans le "Dialogue Act Markup in Several Layers" (DAMSL) [Core et Allen \(1997\)](#).

Les actes de dialogues sont des unités du discours significative, comme le questionnement ou la déclaration. Les actes de dialogue et autres événements liés au dialogues tels que les expressions subjectives, sont reliées à des segments du discours qui peuvent contenir plusieurs phrases. Un exemple d'actes de dialogue représentant une question : "Est-ce qu'il y a un vol Air France de Paris à Montpellier?".

```
{Q001
est_un QUESTION [
  q_type Oui/Non
  q_theme CONNECTION [
    c_sujet Air France
    c_chemin CHEMIN [
      origine Paris
      destination Montpellier
    ]
  ]
]
```

}

Des frames différentes peuvent partager des emplacements par des liens de similarité. Ils peuvent être *nécessaires* ou alors *optionnels*. Les “fillers” peuvent être obtenus par rattachement de procédures ou des détecteurs (par exemple groupes nominaux, héritage, valeurs par défaut). Les descriptions attachées aux emplacements pour spécifier des contraintes incluent les connectives, les coréférences (ces descriptions attachées à un emplacement sont attachées aux emplacements d’autres frames et vice-versa), les conditions déclaratives, les expressions quantifiées.

Dans KL-ONE [Brachman et Schmolze \(1985\)](#) les descriptions de concepts sont des structures internes avec des *Rôles*, des descriptions de “Fillers” et une structure de description (SD) globale. Les rôles ont des sous-structures avec des contraintes spécifiant les types et le nombre de “Fillers”. Les SDs sont des expressions indiquant comment les “Fillers” de rôles interagissent. Les descriptions de rôles contiennent des restrictions sur les valeurs. Les relations épistémologiques sont définies pour la composition de structures conceptuelles. Elles peuvent relier des objets formels du même type et compter pour de l’héritage.

Les langages de représentation sémantique riches ont été récemment développés et se basent sur les concepts introduits précédemment. Un exemple est OWL : Web Ontology Language [Patel-Schneider et al. \(2003\)](#) issu des travaux sur le web sémantique [Berners-Lee et al. \(2001\)](#) et des langages à balises comme DARPA Agent Markup Language décrit dans [Hendler et McGuinness \(2000\)](#).

### 1.3 Extraction du sens dans les phrases

L’interprétation d’une phrase pour obtenir une description dans un L.R.S. est une activité de résolution de problèmes qui peut être réalisée avec des méthodes et des modèles différents. Il n’est pas trivial de relier des concepts se rapportant à des mots. [Woods \(1975\)](#) énonce que la spécification pour la sémantique du langage naturel n’est pas limitée à la transformation d’une phrase en une représentation et peut exiger l’utilisation des informations qui ne sont pas codées dans la phrase, mais peut être déduite de la connaissance du système stockée dans la mémoire à long ou à court terme. En outre, dans [Jackendoff \(1990\)](#), chaque constituant syntaxique majeur d’une phrase correspond à un constituant conceptuel, mais l’inverse n’est pas vrai.

L’extraction d’informations sémantiques à partir d’une phrase peut être réalisée en premier lieu par l’analyse de la phrase pour obtenir une analyse syntaxique complète de celle-ci. En supposant que les langues naturelles sont sensibles aux mêmes types d’analyse sémantique des langages de programmation, dans [Montague \(1974\)](#), il est suggéré que chaque règle syntaxique d’un langage naturel à grammaire générative est associé à un procédé de construction sémantique qui transforme une phrase en une formule logique. L’interprétation peut donc être considérée comme une traduction de phrases en formules logiques de premier ordre afin d’exprimer le sens de la phrase. L’ontologie d’un domaine d’application fait partie de la connaissance du système et

peut être représentée par des ensembles de structures de “frames”. Chaque structure correspond à un ensemble de formules de la logique du premier ordre. Si la logique du premier ordre n’est pas adéquate pour représenter certains types de connaissances, les logiques d’ordre supérieur ou les logiques modales peuvent être utilisées. Quelques exemples peuvent être trouvés dans [De Mori \(1998\)](#).

Les formules décrivant le sens d’une phrase suivant l’approche proposée dans [Montague \(1974\)](#) sont obtenues par composition. Plus récemment, les grammaires catégorielles combinatoires ( [G.C.C.](#) ) ont été introduites [Steedman \(1996\)](#) pour préciser les formes logiques pour chaque phrase analysée sous le contrôle d’une grammaire catégorielle du type introduite dans [Lambek \(1958\)](#). Une méthode intéressante pour induire une [G.C.C.](#) à partir d’exemples est décrite dans [Zettlemoyer et Collins \(2005\)](#). Des développements récents avec utilisation des [G.C.C.](#) pour l’interprétation des phrases en langage naturel dans un dialogue sont décrits dans [Zettlemoyer et Collins \(2009\)](#).

Les formules logiques pour la représentation sémantique des [G.C.C.](#) sont des lambda-expressions. Un mot du lexique d’application est décrit par une catégorie  $B : g$  où  $B$  est un type syntaxique et  $g$  est un type sémantique. Les types syntaxiques sont composés de types primitifs comme les phrases nominales (NP) et de types complexes comme  $\sigma/F$  ou  $P \setminus \sigma$  où  $\sigma/F$  représente une chaîne de caractères  $\sigma$  qui doit être suivie de la chaîne de caractères  $P$  pour être complétée et  $P \setminus \sigma$  représente une chaîne  $\sigma$  qui doit être précédée par la chaîne  $P$  pour être complétée. Un type sémantique peut être une expression lambda qui est une fonction à partir de valeurs variables logiques vers des valeurs de vérité. Souvent, des expressions lambda sont utilisées pour obtenir les valeurs des arguments d’un élément de frame pour trouver les valeurs des variables pour lesquelles une lambda-expression retourne VRAI.

Par exemple, la phrase “une location près de l’opéra” est représentée par une lambda expression  $\lambda(x).location(x) \wedge near(x, Opera)$ . Chaque élément du lexique est défini comme suit :  $location := NP : location, Opera := NP : opera, near := (S/NP)/NP : \lambda(x).\lambda(y).near(x, y)$ . IL est possible d’analyser syntaxiquement la phrase et d’obtenir  $S : \lambda(x).location(x) \wedge near(x, Opera)$ , dans ce cas,  $\lambda(x)$  est une fonction qui retourne VRAI pour certains valeurs de la variable  $x$ .

Souvent, une phrase peut être analysée de différentes manières, sous le contrôle d’une grammaire. Pour cette raison, les grammaires probabilistes ont été introduites. L’analyse des résultats sous le contrôle de ces grammaires se fait pour un certain nombre d’arbres syntaxiques candidats. Chaque candidat est associé à la probabilité de représenter la production réelle de la phrase.

Les probabilités sont estimées avec un corpus d’apprentissage. En raison de la grande variété de possibilités et les limites des corpus disponibles, beaucoup de probabilités ne sont pas bien estimées à cause de la dispersion des données. Des améliorations ont été associées en ajoutant des contraintes à l’introduction de règles de grammaires et d’approximations dans les probabilités du modèle. Il a également été jugé utile de lexicaliser les grammaires en associant des mots clés pour les composants syntaxiques hypothésés lors de l’analyse. Des détails importants sur l’application de ces concepts peuvent être trouvés dans [Zettlemoyer et Collins \(2009\)](#) et ses références au [G.C.C.](#) et d’autres

grammaires lexicalisées.

Dans [Woods et al. \(1976\)](#), la connaissance pratique, syntaxique et sémantique sont intégrés dans des procédures sémantiques sensible au contexte appelées Grammaires à Réseaux de Transitions Augmentées ( [G.R.T.A.](#) ) ou “Augmented Transition Network Grammars ( [A.T.N.G.](#) )”. Les [G.R.T.A.](#) sont faits d’états et d’arcs représentant les transitions parmi les états. Les [G.R.T.A.](#) sont une extension des grammaires de réseaux de transitions ( [G.R.T.](#) ) indépendantes du contexte ou “transition network context-free grammars ( [T.N.G.](#) )”. Dans les [G.R.T.](#) , il y a trois types d’arcs, nommés CAT pour lire un symbole donné en entrée, PUSH pour transférer le contrôle de l’analyse à un autre [G.R.T.](#) , et POP pour transférer le contrôle à partir d’un sous-réseau qui a exécuté PUSH sur celui-ci.

Les [G.R.T.A.](#) sont une extension de [G.R.T.](#) sans laquelle les actions sur les conditions de tests et l’enregistrement des paramètres sont associés avec certains arcs. Les actions définissent le contenu des enregistrements avec des valeurs linguistiques paramétrées. On peut les utiliser pour construire des arbres d’analyse syntaxique. Il est également possible d’introduire des actions du type BUILD associées à un arc pour composer un arbre d’analyse ou pour générer des interprétations sémantiques. Les différentes [G.R.T.A.](#) s peuvent être utilisées en cascade pour l’analyse et l’interprétation. Dans ce cas, un type d’arc TRANSMIT transfère des structures syntaxiques à partir de la syntaxe d’une [G.R.T.A.](#) sémantique. Une étiquette d’arc TEST est introduite pour tester les conditions sur le contenu des registres ou des listes créées et remplies avec un formulaire LIST. Des règles pour définir les actions, les formulaires sont également introduites.

Un exemple simple de sous-réseau [G.R.T.A.](#) pour une phrase nominale est montré dans la [Figure 1.2](#). Une action “SETR SUBJ \*” est associée avec l’arc étiqueté avec un nom propre “N”. L’action modifie le registre SUBJ avec la valeur du mot en entrée correspondant au nom. Le contenu du registre peut être utilisé plus tard par une action “BUILD” associée à un autre arc pour la composition d’arbre syntaxique sur une phrase analysée ou sur l’instance d’une structure sémantique.

Dans [Walker \(1975\)](#) la connaissance sémantique est représentée par un réseau sémantique partitionné. Le premier meilleur analyseur est utilisé. Ses résultats déclenchent des activations dans le réseau sémantique partitionné avec lesquelles des déductions et des prévisions sont effectuées par propagation de l’activation nœud grâce à des liens.

Une approche décrivant une formalisation de la connaissance humaine pour relier les représentations d’actions à la syntaxe se trouve dans [Allen \(1988\)](#). Les relations sémantiques peuvent utiliser des composants hypothèses dans des phrases différentes et générer des structures qui peuvent appartenir à un langage dépendant du contexte. Les séquences de constituants conceptuels devraient pouvoir satisfaire à des contraintes qui sont différentes des contraintes imposées aux mots exprimant chaque constituant. Ces contraintes peuvent être représentées par des motifs de séquences de mots ainsi que des fonctions syntaxiques.

D’autres approches pour l’interprétation des phrases ont été proposées. En géné-

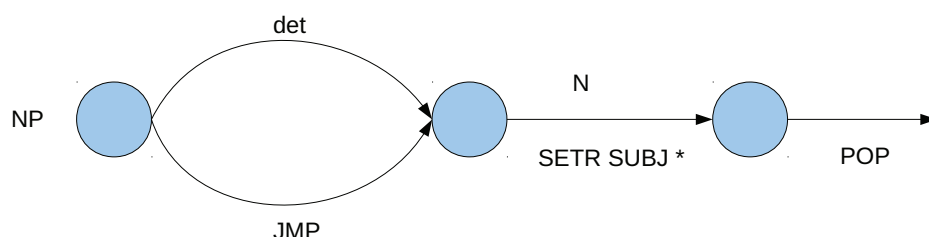


FIGURE 1.2 – Exemple de Grammaire à Réseaux de Transitions Augmentées

ral, l'interprétation peut être dirigée par l'analyse syntaxique ou par d'autres stratégies dans lesquelles l'analyse syntaxique et l'analyse sémantique de composition sont entrelacées. En outre l'analyse syntaxique, peut être basée sur l'analyse complète ou partielle. Les premières approches seront brièvement passées en revue dans les chapitres 2 et 3, tandis que d'autres approches plus récentes seront décrites dans le chapitre 4.

La sémantique peut aussi piloter l'analyseur syntaxique, l'amenant à faire des attachements dans l'arbre d'analyse. La sémantique peut résoudre les ambiguïtés et traduire des mots anglais en symboles sémantiques en utilisant un réseau de discriminant d'homonymie. L'analyse sémantique y est discuté dans [Tait \(1985\)](#).

Certaines approches antérieures effectuent l'interprétation comme un processus fondé sur des règles d'inférence. Les règles peuvent contenir des fonctionnalités d'analyse et des inférences peuvent être réalisées par un système de planification dont le but est d'obtenir une représentation L.R.S. . Plutôt que d'intégrer des règles de construction dans une grammaire, des architectures hiérarchiques ont été conçues pour produire des interprétations en entraînant l'application de règles de production [Erman et al. \(1980\)](#).

D'autres exemples de systèmes basés sur des règles proposées pour générer des instances de structures à ossature exprimées par des phrases du langage naturel peuvent être trouvés dans [Norvig \(1983\)](#).

Ces approches utilisent essentiellement des modèles probabilistes non-génératifs pour traduire une phrase dans sa représentation [L.R.S.](#) .

L'analyse syntaxique de la langue parlée est particulièrement difficile, surtout en présence d'erreurs de Reconnaissance Automatique de la Parole . Pour cette raison, les démarches évitant l'analyse syntaxique profonde ont été pleinement prises en considération. Elles génèrent des hypothèses de fragments sémantiques qui doivent être composées par un processus de composition successive d'un modèle sémantique. Plus de détails seront donnés dans les chapitres et sections suivants.



## Chapitre 2

# Interprétation basée sur l'analyse syntaxique complète

### Sommaire

---

<b>2.1</b>	<b>Grammaires à réseaux de transition</b> . . . . .	<b>32</b>
<b>2.2</b>	<b>Grammaires et analyseurs utilisés pour la Compréhension du Langage Parlé dans le projet Air Travel Information System</b> . . . . .	<b>35</b>
2.2.1	Le système DELPHI . . . . .	35
2.2.2	Le système Gemini et les système basés sur la "concordance de modèles" (Template Matcher) . . . . .	37
<b>2.3</b>	<b>Grammaires Lexicalisées</b> . . . . .	<b>41</b>
<b>2.4</b>	<b>Grammaires stochastiques pour l'interprétation</b> . . . . .	<b>49</b>
2.4.1	Le système TINA . . . . .	49
2.4.2	Le système HUM . . . . .	51
<b>2.5</b>	<b>Le modèle à vecteurs d'états cachés de Cambridge</b> . . . . .	<b>52</b>
<b>2.6</b>	<b>Conclusions</b> . . . . .	<b>54</b>

---



L'interprétation d'un texte écrit peut être vue comme un processus qui traduit une séquence de mots en langage naturel dans un ensemble de structures sémantiques décrit par un langage sémantique. Le langage sémantique a sa propre sémantique qui peut contenir des procédures. Il y a une possibilité d'intégrer ces procédures dans un analyseur syntaxique.

Dans [Waltz \(1981\)](#), il est dit qu'un analyseur sémantique travaille avec un analyseur syntaxique et produit des données acceptables dans un système de déductions logiques. Cette démarche est motivée par des arguments, par exemple dans [Jackendoff \(1990\)](#), que chacun des constituants syntaxiques majeurs d'une phrase correspond à un constituant conceptuel, mais l'inverse n'est pas vrai. Par exemple, en adaptant la notation [Jackendoff \(1990\)](#), une phrase de requête pour un restaurant près de la station de métro Montparnasse à Paris peut être représentée par ce qui suit, l'expression de la structure conceptuelle est entre crochets :

```
[
  Action REQUEST (
    [Thing RESTAURANT],
    [Path NEAR (
      [Place IN (
        [Thing MONTPARNASSE])
      ])
    ]
  )
]
```

Ce formalisme est basé sur un ensemble de catégories. Chaque catégorie, par exemple "Place" peut être élaborée en tant que fonction de lieu, par exemple "IN" et son argument.

L'expression peut être obtenue à partir d'une structure syntaxique comme ceci :

```
[S
  [VP
    [V give, PR me]
    NP[ART a, N restaurant]
    PP[PREP near, NP [N Montparnasse, N station]
  ]
]
```

### 2.1 Grammaires à réseaux de transition

Des efforts considérables dans la recherche sur la [C.L.P.](#) ont été faits avec un projet ARPA commencé en 1971. Le projet est examiné dans [Klatt \(1977\)](#), il a été principalement basé sur des approches de l'Intelligence Artificielle (IA). Comme le langage na-

turel est sensible au contexte, les réseaux de procédure pour l'analyse sous le contrôle des Grammaires de Réseaux de Transitions Augmentés ( **G.R.T.A.** ) ou "Augmented Transition Network grammaires" ( **A.T.N.G.** ) ont été proposées.

Les **G.R.T.A.** sont une extension des grammaires à réseau de transition ( **G.R.T.** ), qui peuvent être utilisés pour générer n'importe quel langage hors-contexte ( **L.H.C.** ). En outre, l'analyse des **L.H.C.** peut être formulée en termes de Réduction des Problèmes de Représentation Hall (1973).

Les **G.R.T.** sont des graphes avec des états et des arcs. La chaîne d'entrée est analysée de gauche à droite, un mot à la fois, supposant que ce soit dans un état actif. Le mot d'entrée et l'état actif déterminent l'arc suivi par l'analyseur. Les arcs ont des types, à savoir CAT (pour lire un symbole d'entrée), PUSH (pour transférer le contrôle à un sous-réseau) et POP (pour transférer le contrôle à partir d'un sous-réseau pour le réseau qui a exécuté le PUSH).

Pour analyser un énoncé, un système basé sur les **G.R.T.** a été proposé par Young et al. (1989); Ward (1990). Les phrases sémantiques sont reconnues par remplissage d'emplacements. Il n'y a pas de grammaire globale, chaque emplacement possible a une grammaire propre, représentée par un **G.R.T.** avec la possibilité d'activer un analyseur de graphe pour chacun d'eux.

C'est une approche intéressante dans laquelle des dispositifs distincts sont utilisés pour générer des hypothèses d'interprétation. Avec des approches de ce type, les processus de remplissage d'emplacements devraient être transférables d'une frame à l'autre et il devrait être possible d'utiliser des valeurs par défaut, pour proposer d'autres solutions lorsque des pannes se produisent, pour effectuer un remplissage partiel et la correction d'erreur.

Les **G.R.T.A.** sont des améliorations de **G.R.T.** dans lesquelles les actions de tests des conditions et le paramétrage d'enregistrements/registres sont associées à certains arcs. Les actions définissent le contenu des enregistrements avec les valeurs de caractéristique linguistique et peuvent également être utilisées pour construire des arbres syntaxiques. Les **G.R.T.A.** ont été introduites dans Woods (1970) pour l'analyse du langage naturel. Une nouvelle étiquette "TEST" d'arc a été introduite afin de tester les conditions sur le contenu des enregistrements ou des listes créées et remplies avec le formulaire LIST. Des règles pour définir les actions, les formulaires et les actes ont également été introduites.

Les **G.R.T.A.** sont exécutées, si elles sont représentées par un ensemble de descriptions d'arcs et d'états décrit comme suit :

```
(state
  (arc label/condition
    (action)
    (destination)
  )
)
```

L'état initial est le seul état actif au commencement. L'interprétation sémantique lève les états actifs à partir d'une liste et applique des règles du type : *Si l'état est actif alors envisager un mouvement d'arc à partir de cet état, vérifier si le mot peut être lu et la condition est satisfaite, le cas échéant, effectuer une action et pousser l'état de destination dans la liste des états actifs.*

La liste des états actifs peut être classée en fonction de priorités dépendant de probabilités et d'autres scores.

Dans de nombreux cas, en particulier dans des applications de domaine limité, quelques types d'informations représentent une très grande proportion d'énoncés et le processus de compréhension, peut être basé sur un modèle qui correspond à des modèles de mise en correspondance avec les données. Habituellement, les modèles d'emplacements sont remplis par des phrases courtes avec des scores associés. En combinant les scores de remplissage des emplacements, des modèles de scores peuvent être obtenus et l'instanciation avec le meilleur score est considérée comme l'interprétation de la phrase. Des systèmes de ce type sont SAM [Shank \(1975\)](#) et la *concordance de modèles* décrite dans [Jackson et al. \(1991\)](#). Les *G.R.T.A.* ont récemment été utilisés dans des rétro-analyseurs hybrides (RH) [Newman \(2007\)](#) et aussi pour la traduction automatique.

Une action "SETR SUBJ \*" est associée à l'arc étiqueté avec un nom propre N. L'action définit le registre SUBJ à la valeur du mot d'entrée correspondant au nom.

Les *G.R.T.A.* syntaxiques et sémantiques peuvent être utilisés en cascade. Il est également possible d'introduire des actions du type "BUILD" associées à un arc pour composer un arbre d'analyse ou pour générer des interprétations sémantiques.

Dans [Woods \(1975\)](#), des connaissances syntaxiques, sémantiques et pragmatiques sont intégrées dans des réseaux de sémantiques procédurales dans lesquelles les symboles des sous-réseaux peuvent correspondre à des entités syntaxiques ou sémantiques.

Dans [Walker \(1975\)](#) un analyseur de pertinence est utilisé. Ses résultats déclenchent des activations dans un réseau sémantique partitionné avec lequel des inférences et des prédictions sont réalisées par la propagation d'activation.

Réaliser l'analyse syntaxique d'abord, puis l'interprétation sémantique peut être très inefficace [McRoy et Hirst \(1990\)](#). Pour cette raison, d'autres stratégies d'interprétation fondée sur l'analyse ont été proposées.

## 2.2 Grammaires et analyseurs utilisés pour la Compréhension du Langage Parlé dans le projet Air Travel Information System

### 2.2.1 Le système DELPHI

Le système DELPHI [Bobrow et al. \(1990\)](#), développé par Bolt, Beranek and Newman (BBN), contient un certain nombre de niveaux, à savoir, syntaxiques (en utilisant des Grammaires à Clauses Définies ( [G.C.D.](#) ), (Definite Clause Grammar, [D.C.G.](#) )), la sémantique générale, la sémantique de domaine et l'action. Plusieurs traductions sont effectuées à l'aide des liens entre les représentations à différents niveaux. Les règles des [G.C.D.](#) ont des éléments, associés à un foncteur (leur catégorie principale) et zéro ou plus de fonctionnalités dans un ordre de position fixée. Les fonctionnalités sont les emplacements qui peuvent être remplis par des termes. Les termes peuvent être des variables ou des termes fonctionnels. La représentation sémantique est basée sur des frames. Une relation grammaticale a un composant qui déclenche une relation de traduction. La composition fonctionne sur l'interprétation sémantique des arguments pour produire l'interprétation sémantique d'une phrase nouvelle. De cette façon, des *fragments sémantiques* sont construits.

Les [G.C.D.](#) peuvent provenir d'une grammaire annotée indépendante du contexte. Les symboles atomiques de règles sont substitués par des structures complexes. L'analyseur utilise une approche appelée "tout chemin". Lorsque les règles sont appliquées, une factorisation et une unification sont réalisées. Les problèmes résolus par la sémantique de domaine sont les suivants : pluriels distributifs ou collectifs, désambiguïsation en utilisant le contexte, les noms relationnels, des contraintes de type sémantique. Les relations de contraintes peuvent faire : filtrage en cas d'analyse, calcul des valeurs d'une variable, sous-catégorisation.

DELPHI contient un analyseur linguistique qui génère les N-meilleures hypothèses en utilisant un algorithme simple et rapide, [Schwartz et al. \(1992\)](#); [Ostendorf et al. \(1991\)](#), puis à plusieurs reprises ré-évalue ces hypothèses par le biais d'algorithmes plus complexes et plus lents. De cette manière, plusieurs sources de connaissances différentes, peuvent contribuer au résultat final sans compliquer la structure de contrôle ou sans ralentir considérablement la dérivation du résultat final.

La première version de DELPHI utilisait un analyseur graphique basé sur l'unification [Austin et al. \(1991\)](#). Une caractéristique importante et utile de cet analyseur, qui a été retenue dans toutes les versions ultérieures, a été l'incorporation des probabilités pour les différents sens d'un mot et pour l'application des règles grammaticales. Ces probabilités sont estimées à partir des données et utilisées pour réduire l'espace de recherche pour l'analyse [Bates et al. \(1994, 1993\)](#).

Un module de secours robuste a été intégré dans les versions successives [Bates et al. \(1994\)](#). Le module de compréhension de repli dans DELPHI est appelé, si l'analyseur graphique d'unification échoue [Stallard et Bobrow \(1993\)](#); [Bates et al. \(1993\)](#). Plutôt que

d'employer le module sémantique et d'attribuer une note explicite en langage naturel à des hypothèses, DELPHI essaye d'analyser les premières  $N = 10$  hypothèses complètement, s'arrêtant quand une interprétation complète peut être générée. Si cela ne fonctionne pas, un autre passage à travers ces dix hypothèses est fait avec le module de secours, qui tente de produire une interprétation solide à partir de fragments analysés plus à gauche de la première analyse qui a échoué.

Le module de secours lui-même est composé de deux parties : le combineur syntaxique et le combineur de frames. Le combineur syntaxique utilise des règles grammaticales étendues qui sautent les éléments intervenants dans une tentative pour générer une analyse complète. Si la tentative a échoué, le combineur de frames essaye de combler les emplacements dans les frames d'une manière similaire à celle de la concordance de modèles du "SRI". Le combineur de frames utilise de nombreuses règles pragmatiques obtenues par l'étude des données d'entraînement qui ne pouvaient pas être supportées sur des motifs abstraits. Par exemple, les interprétations qui combinent l'information de vol et de transport terrestre sont exclues parce qu'elles ne sont jamais observées dans les données, même si une requête comme "Afficher les vols vers les aéroports avec un service de limousine" est théoriquement possible. Curieusement, le module de secours a fonctionné mieux si seulement le combineur de frames - mais pas le combineur syntaxique - était inclus.

Afin d'augmenter la robustesse et de réduire la dépendance sur le module de secours [Bates et al. \(1994\)](#), un graphe sémantique de structure de données a été introduit et la preuve syntaxique est considérée comme une seule façon de déterminer les liens sémantiques à partir desquels est construit le graphe [Bates et al. \(1994\)](#). Un graphe sémantique est un graphe orienté acyclique dans lequel les nœuds correspondent aux significations des mots de tête (par exemple l'arrivée, le vol, Boston) et les arcs sont des relations sémantiques binaires. L'opération d'analyse de base est de lier deux graphes déconnectés avec un nouvel arc.

Si l'analyseur graphique ne parvient pas à connecter ces graphes déconnectés, le relieur sémantique est invoqué. Les fragments sont les nœuds lexicaux, la combinaison est la complétion de graphes par une recherche, les probabilités de liaison sont issues de corpus. Ce composant peut ignorer l'ordre des fragments, sauter sur les éléments inanalysables, et même créer un nouveau nœud si c'est le seul moyen de lier des fragments. Par exemple, l'énoncé «Boston midi de première classe à Denver" provoque une hypothétisation d'un nœud FLIGHT dont les fragments sont suspendus.

La représentation sémantique est basée sur des frames comme suit :

```
(FLY
  (aircraft-of 747)
  (airline-of Delta)
  ( destination-of Denver)
)
```

Une relation grammaticale a un composant qui déclenche une relation de traduction :

```
(NP.....) -> HEAD (NP....) (PP-COMP (ON .....))  
(PP-COMP (ON .....)) -> FLY. airline-of
```

Un argument Phrase Prépositionnelle (PP) est lié à une phrase nominale (NP) pour faire un nouvel objet PP. La réalisation de la relation fonctionne sur l'interprétation sémantique des arguments pour produire l'interprétation sémantique d'une phrase nouvelle. De cette façon, des fragments sémantiques sont construits.

### 2.2.2 Le système Gemini et les système basés sur la "concordance de modèles" (Template Matcher)

Le système développé au Stanford Research Institute (SRI) se compose de deux modules sémantiques attelés ensemble : un module d'unification de grammaire appelé "Gemini", et la "concordance de modèles" ou "comparateur de modèles" qui agit comme un repli si Gemini ne peut pas produire une requête base de données acceptable [Hobbs et al. \(1996\)](#); [Appelt et al. \(1995\)](#).

Gemini est un analyseur du langage naturel basé sur l'unification qui combine des règles syntaxiques générales et des règles sémantiques pour l'anglais avec un lexique spécifique pour [A.T.I.S.](#) des restrictions de sélection et d'ordre.

Les analyseurs dirigés par la sémantique utilisent des modèles pour reconnaître les éléments qui correspondent avec les données. Les correspondances peuvent commencer avec les modèles lexico-sémantiques pour l'instanciation initiale des éléments lexicaux. Les interprétations sont construites par l'ajout d'éléments non-lexicaux inférés par un algorithme de recherche [Stallard et Bobrow \(1993\)](#).

Différents types de comparateurs peuvent être conçus à des fins différentes. Quand le but est uniquement de récupération, un vecteur de caractéristiques peut représenter adéquatement le contenu d'un message. Différentes structures sont nécessaires si l'objectif est d'obtenir une représentation conceptuelle pour être utilisé pour l'accès à une base de données ou pour un dialogue dont l'objectif est l'exécution d'une action. Les comparateurs de modèles d'états finis, les analyseurs lexicaux et comparateurs de modèles au niveau de la phrase sont discutés dans [Hobbs et Israel \(1994\)](#).

Les expressions imprévues et les constructions difficiles de la langaue parlée causent des problèmes pour une approche conventionnelle. Quelques types d'informations représentent une proportion très importante des énoncés. Un comparateur de modèle (TM) essaie de construire des modèles (quatre modèles de base instanciés de diverses manières).

Les modèles ont des emplacements remplis par la recherche de courtes phrases produites par la reconnaissance, même si les mots n'ont pas tous été mis en hypothèse correctement. Les scores d'instanciation sont essentiellement les pourcentages de mots corrects. Le modèle avec le meilleur score est utilisé pour construire la requête.

L'entrée du comparateur de modèles est la meilleure séquence d'hypothèses de mots générée par le composant de reconnaissance vocale, qui utilise un modèle de lan-

gage bi-gramme. Le TM essaie simplement de combler les emplacements de frames comme modèles. Une première version avait seulement 8 modèles traitant des vols, le transport terrestre, sens des codes et des rubriques, des avions, les villes, les compagnies aériennes et les aéroports. Les différents modèles sont en concurrence les uns avec les autres sur chaque énoncé, tous sont marqués, et le modèle avec le meilleur score génère la requête de base de données (à condition que son score soit supérieur à un certain seuil). Les emplacements sont remplis en regardant à travers l'énoncé de certaines phrases et des mots. Des systèmes similaires sont SAM (Shank (1975); Ge et Mooney (2005)).

Voici un exemple typique, adapté de Jackson et al. (1991). Pour l'énoncé : "Montrez-moi tous les vols Delta de Denver à Atlanta sans escale pour le douze du mois d'avril pour après dix heures du matin", le modèle de vol suivant sera généré :

```
[flight, [stops, nonstop],
         [airline, DL],
         [origin, DENVER],
         [destination, ATLANTA],
         [departing\_after,
          [1000]],
         [date,
          [april, 12, current\_year]]
]
```

Les mots dans l'énoncé peuvent contribuer à la sélection et au remplissage d'un modèle de diverses manières :

- ils peuvent aider à identifier le modèle - l'apparition du mot "downtown" est un bon indicateur du modèle de transport au sol
- ils peuvent remplir un emplacement - comme "Denver" et "Atlanta" dans l'exemple
- ils peuvent aider à indiquer quelle partie de phrase va dans quel emplacement - comme "de" ou "à" qui précède une phrase

Cependant, beaucoup de mots ne sont pas pertinents : « s'il vous plaît », « Montrez moi », « vous », et ainsi de suite. Le comparateur de modèles les saute tout simplement. Le score d'un modèle est fondamentalement le pourcentage de mots dans l'énoncé qui contribuent à combler le modèle. Cependant, certains mots clés qui sont fortement corrélés avec un modèle particulier auront un fort impact dans le score de ce modèle, si ils se produisent dans l'énoncé. Par exemple, l'apparition de *combien*, *tarif*, ou le *prix* augmente le score du modèle *tarif* ; l'apparition de *quel est*, *d'expliquer* ou de *définir* augmente le score du modèle *signification*. Toutefois, si le modèle n'a pas d'emplacements remplis, on lui attribue un score de zéro, ou si le système a essayé d'assigner deux ou plusieurs valeurs au même emplacement dans le modèle, le modèle est abandonné.

Si le meilleur score ne dépasse pas un certain seuil, le système répond avec *pas de réponse* plutôt qu'avec le modèle qui a donné le meilleur score. Rappelons que la notation A.T.I.S. sanctionne les mauvaises réponses assez durement, alors qu'*aucune réponse* est souvent une réponse préférable ; pour optimiser la performance de leur système sur les tests, les chercheurs du SRI ont fixé le seuil à la valeur qui a donné les meilleurs



résultats sur les données d'entraînement.

Dans une version ultérieure du système de C.L.P. , l'analyseur classique a été abandonné et le comparateur de modèles a été maintenu Appelt et Jackson (1992). Plusieurs améliorations ont été apportées dans le comparateur de modèles. La première version ne pouvait remplir les emplacements avec des mots ou des phrases fixes - il n'avait pas la capacité de traiter avec des catégories générales comme les dates et heures. Dans la deuxième version, des expressions relevant de ces catégories dans l'énoncé entrant ont été analysés par des grammaires particulières, et ont ensuite été mis dans les emplacements par le comparateur. Les emplacements ont été remplis par correspondance des expressions régulières - par exemple, *à partir de* suivi d'un nom de ville ou aéroport entraîne à mettre le nom de ville de départ dans l'emplacement du modèle de vol.

La différence majeure entre les versions du comparateur de modèles est que la dernière contient un mécanisme de traitement de contexte pour la classe "D" (dépendant du contexte) des requêtes, la première version a été uniquement conçue pour fonctionner avec la classe "A" (indépendant du contexte, acceptable ) des requêtes. Le mécanisme de traitement de contexte permet aux emplacements d'hériter des précédentes déclarations du même utilisateur.

Notez que si le premier énoncé de l'utilisateur est mal interprété, la suite d'énoncés dépendants du contexte peuvent tous être correctement reconnus mais mal compris.

Si les systèmes avaient été autorisés à demander aux utilisateurs la confirmation de temps à autre, ce problème serait relativement peu important. Toutefois, les évaluations A.T.I.S. ne permettaient pas le dialogue, ainsi plusieurs mécanismes ingénieux ont été conçus pour prévenir le problème. Par exemple, lorsque le système n'avait pas répondu à une requête, la suite des requêtes en fonction du contexte donnait "*aucune réponse*", jusqu'à ce qu'une requête qui définissait un contexte totalement nouveau arrive. En mode de collecte de données, d'autre part, le système génère une réponse, si possible, sur la base que les utilisateurs préfèrent des réponses légèrement inexacts plutôt que "Je ne comprends pas".

D'autres travaux n'ont pas insisté sur le comparateur de modèles, mettant l'accent sur la composante Gemini Cohen et Moore (1994); Moore et al. (1995). Néanmoins, le comparateur de modèles agit comme une sauvegarde de Gemini dans les versions ultérieures. Dans Moore et al. (1995), il est signalé que, dans la dernière version du système, le comparateur de modèles a abouti à l'amélioration d'environ 5-10% du taux d'erreur par rapport à Gemini seul.

Un exemple simplifié d'une règle syntaxique Gemini Dowding et al. (1993) :

```
syn (whq\_ynq\_slash\_np,  
    [s:  
      [sentence\_type=whq, form=tnsd, gapsin=G, gapsout=G],  
      np:[wh=ynq, pers\_num=N],  
      s:[sentence\_type=ynq,  
        form=tnsd,  
        gapsin=np:[pers\_num=N],
```



```

        gapsout=null]
    ]).

```

Cette règle, nommée “whq\_ynq\_slash\_np”, dit qu’une phrase (catégorie s) peut être composée d’un syntagme nominal (catégorie NP) suivi d’une phrase, où NP a la valeur ymq, pour ses caractéristiques “WH” et la valeur N fonctionnalité pers\_num. La phrase fille a une valeur de catégorie pour ses vides dans les caractéristiques (NP) dont la valeur pers\_num est la même que la valeur pers\_num de l’hypothèse de la phrase. La règle signifie que une phrase sans vides avec une phrase du type WHQ peut être construite par la recherche d’une phrase WH suivie d’une phrase nominale avec un vide en elle et qui a le même nombre de personnes que dans la phrase WH [Dowding et al. \(1993\)](#).

Les règles sémantiques sont similaires, sauf que chacun des constituants mentionnés dans le squelette de structuration de phrase est associé à une forme logique. La sémantique de la règle syntaxique ci-dessus est :

```

sem(whq\_ynq\_slash\_np,
    [ ([whq, S], s: []),
      (Np, np: []),
      (S, s: [gapsin=np: [gapsem=Np]])
    ]
).

```

Pour une règle syntaxique simple, un certain nombre de règles sémantiques peuvent être écrites. À la fin de 1992, Gemini contenait 243 règles syntaxiques et 315 règles sémantiques [Dowding et al. \(1993\)](#). Le lexique de Gemini spécifique au domaine [A.T.I.S.](#) à cette époque contenait 1.315 entrées, ce qui pourrait être étendu à 2.019 entrées par les règles morphologiques du système. Les restrictions de sélection et d’ordre de Gemini comprennent à la fois des informations prédicat-arguments très spécifiques à un domaine (par exemple, l’objet du verbe “sortir” doit être un aéroport ou une ville) et des restrictions très générales de prédicats (par exemple, l’adjectif “plus loin” s’applique à distances).

Un composant important de Gemini manipule des systèmes de réparations. Les réparations se produisent lorsque le locuteur a l’intention de supprimer un ou plusieurs mots. Pour trouver les constituants, Gemini utilise un algorithme de “tous les chemins”, un analyseur de graphe de bas en haut, qui identifie toutes les arêtes possibles recevables par la grammaire. L’analyseur est “en ligne”, ce qui signifie que toutes les arêtes finissant à la position  $i$  sont construites avant celles qui terminent à la position  $i + 1$ . Les traitements syntaxique et sémantique sont entrelacés : quand une règle syntaxique a été satisfaite sur l’espace d’un groupe de mots, des règles sémantiques et de contrôle d’ordre prennent place. L’arête syntaxique n’est ajoutée au graphe que si elle est déterminée à être sémantiquement bien formée. Une fois que tous les constituants ont été trouvés, Gemini utilise un analyseur de phrases (avec un ensemble différent de règles syntaxiques et sémantiques) pour construire une analyse de l’énoncé complet.

Dans la tentative d’utiliser des contraintes du langage naturel pour améliorer la

reconnaissance, un analyseur regarde les N-meilleures hypothèses de sortie par la reconnaissance, l'hypothèse privilégiée est la plus proche de la partie supérieure qui peut être analysée complètement. Le problème est que les grammaires modèlisent rarement la parole spontanée avec précision.

Gemini traite cette analyse, en séparant les constituants de ce qui est exprimé. Ainsi, Gemini peut souvent identifier les possibles constituants syntaxiques d'un énoncé, même si il n'a pas été en mesure d'analyser l'énoncé dans son ensemble. Ce fait a été exploité pour mettre au point une méthode heuristique pour la notation des N-meilleures hypothèses de sortie de la reconnaissance. Derrière cette heuristique, a été l'hypothèse selon laquelle un petit nombre de constituants est de meilleure qualité qu'un grand nombre. Ainsi, le score final pour une hypothèse combine le score de reconnaissance, le nombre de constituants, un bonus pour une analyse complète de l'énoncé, et des sanctions pour l'utilisation de règles de grammaire inhabituelle. Cette approche a abouti à une légère amélioration sur la sortie du logiciel de reconnaissance seul.

## 2.3 Grammaires Lexicalisées

Basé sur le travail de [Ajdukiewicz \(1935\)](#), [Bar-Hillel \(1953\)](#) a proposé un formalisme de grammaire dans lequel chaque dérivation syntaxique correspond à une interprétation sémantique.

En supposant que les langues naturelles sont sensibles à la nature même de l'analyse sémantique des langages de programmation, dans [Montague \(1974\)](#), il est suggéré que chaque règle syntaxique d'une grammaire générative de la langue naturelle est associée à une règle sémantique qui transforme la phrase dans une formule logique.

Suite à Montague, dans [Gazdar et al. \(1985\)](#), une fonction est introduite qui associe à une catégorie syntaxique  $x$ , l'ensemble des dénotations possibles associées à  $x$  dans le modèle sémantique d'une langue. Les détails et discussions au sujet de ces associations et leur utilisation peuvent être trouvés dans [Partee \(1976\)](#). La conciliation de la dépendance conceptuelle avec les grammaires de Montague est discuté dans [Jones et Warren \(1982\)](#). L'utilisation d'un lexique avec des grammaires de Montague est discutée dans [Dowty \(1979\)](#).

L'association des formules de construction sémantique avec l'analyse syntaxique a d'abord été proposée dans les grammaires catégorielles conçues pour obtenir une représentation sémantique de surface. Dans [Lambek \(1958\)](#), la syntaxe d'une langue est considérée comme une algèbre, les catégories grammaticales sont considérées comme des fonctions. Les représentations lexicales ont associé un modèle syntaxique qui suggère des continuations possibles de l'analyse syntaxique et sémantique de l'expression qui seront produits, comme indiqué dans le fragment suivant du lexique :

write	$(S \setminus NP) / NP$	$\lambda x \lambda y ((WRITE x) y)$
Mary	$S / (S \setminus NP)$	$\lambda f (f Mary)$
a	$NP / N$	$\lambda x (a n x)$
letter	$N$ .	

Les éléments sont associés à une catégorie syntaxique qui les identifie comme des fonctions et précise le type et la directivité de leurs arguments et le type de leurs résultats. Ainsi, dans l'exemple "Marie mange une orange", l'entrée lexicale  $\langle mange(S \setminus NP) / NP \rangle$  provoque le fait que lorsque "mange" dans les données apparaît avec l'entrée lexicale pour "manger", la fonction associée  $(S \setminus NP) / NP$  est appliquée. Le symbole / indique une application de fonction *en avant* qui cherche une correspondance avec une suite "NP" "mange" et exige l'évaluation de la fonction  $(S \setminus NP)$ . Le mot "un" est une entrée lexicale  $\langle NP / N \rangle$ . Cela provoque l'exécution d'une autre application de fonction en avant qui cherche un nom qui suit "une". Comme le nom est trouvé ( $\langle orange N \rangle$ ), la fonction sémantique  $x$  (un  $x$ ) est exécutée, renvoyant (une orange) qui est associée à l'affirmation de NP qui correspond désormais avec ce qu'on peut attendre de  $(S \setminus NP) / NP$  avec (une orange). Le  $x$  de  $\lambda x \lambda y ((MANGE x) y)$ , est lié à (une orange) conduisant à  $\lambda y ((MANGE une orange) y)$ . Maintenant, la fonction *en arrière*  $S \setminus NP$  doit être exécuté. Le symbole \ signifie que la fonction regarde en arrière pour faire correspondre une entrée lexicale avec le label NP qui est trouvé en effectuant l'exécution en avant de la fonction associée à l'entrée lexicale  $\langle SMarie / (S \setminus NP) \rangle$ . La fonction considère l'affirmation de S si ce qui suit est affirmé. Cela est vrai parce c'est la prévision *en arrière* du verbe et NP est une réécriture de Marie. En conséquence de l'appariement,  $y$  est lié à Marie, la production de la représentation sémantique  $((MANGE une orange) Marie)$  et provoque l'affirmation que le symbole de départ  $S$  avec l'analyse de la phrase à interpréter, est terminé avec succès.

Afin d'améliorer l'utilisation des grammaires hors-contexte ou légèrement sensibles au contexte, les règles de production ont été spécialisées par la lexicalisation. Cela consiste à inclure des têtes de mots dans la partie non terminale gauche de la règle de réécriture. Des règles simples peuvent identifier la tête d'une phrase.

Quelques exemples :

- La tête d'un VP est le verbe principal
- La tête NP est le nom principal
- La tête PP est la préposition principale
- La tête de phrase est à la tête de son VP

Les grammaires catégorielles combinatoires sont décrites en détail dans [Steedman \(2000\)](#). En grammaire catégorielle, des informations sur l'ordre des mots et de valence sont codées directement dans des catégories syntaxiques qui sont assignées à des mots. Ces catégories syntaxiques précisent le nombre d'arguments que peut prendre un mot, ainsi que la position relative des arguments à l'égard de la tête.

En grammaire catégorielle, les mots sont associés à des catégories très spécifiques qui définissent leur comportement syntaxique. Un ensemble de règles universelles définit comment les mots et d'autres constituants peuvent être combinés en fonction de

leurs catégories. Des variantes de la grammaire catégorielle diffèrent dans les règles qu'elles permettent, mais des catégories telles que définies ci-dessous sont les pierres angulaires de toute grammaire catégorielle. En général, l'ensemble des catégories syntaxiques  $C$  est défini récursivement comme suit :

**Catégories atomique** : la grammaire pour chaque langage est supposée définir un ensemble fini de catégories atomiques, habituellement :  $S, NP, N, PP... \in C$   
**Catégories complexes** : si  $X, Y \in C$ , alors  $X/Y, X \setminus Y \in C$

Les catégories complexes  $X/Y$  ou  $X \setminus Y$  sont des foncteurs avec un argument  $Y$  et un résultat  $X$ . Ici on suppose une variante de direction de la grammaire catégorielle, qui distingue les arguments à la droite du foncteur (indiqué par la barre oblique /) et des arguments à la gauche du foncteur (indiqué par la barre oblique inverse \).

Nous suivons la notation de Steedman pour représenter les catégories de foncteurs complexes. Dans cette notation, la catégorie précède toujours l'argument. D'où la catégorie  $(S \setminus NP)/NP$  des verbes transitifs en anglais encode l'information selon laquelle le verbe prend un syntagme nominal à sa droite, et un autre syntagme nominal à sa gauche pour former une phrase. Le lexique précise les catégories que les mots d'une langue peuvent prendre. Par exemple, un lexique catégoriel pour l'anglais peut contenir les mentions suivantes :

$$\begin{array}{l}
 \textit{John} \vdash NP \\
 \textit{shares} \vdash NP \\
 \textit{buys} \vdash (S \setminus NP)/NP \\
 \textit{sleeps} \vdash S \setminus NP \\
 \textit{well} \vdash (S \setminus NP) \setminus (S \setminus NP)
 \end{array}
 \tag{2.1}$$

Ici, *John* et *shares* sont des syntagmes nominaux. *Buys* est un verbe transitif. *Sleeps* est intransitif, comme il prend un seul argument NP. *Well* peut modifier *sleeps*, car il faut un verbe intransitif (ou un verbe) comme argument pour retourner un constituant de la même catégorie. Tel que défini ci-dessus, l'ensemble des catégories est infini. Cependant, il est tacitement admis que le lexique d'une langue spécifique, utilise uniquement un sous-ensemble fini de ces catégories, et dans la G.C.C. il y a une forte interaction entre les catégories lexicales de la langue et les règles combinatoires autorisées dans sa grammaire.

Chaque catégorie syntaxique a également une interprétation sémantique dont le type doit correspondre à celui de la catégorie syntaxique. Les interprétations sémantiques peuvent être des objets arbitrairement complexes (tant qu'il y a une correspondance à la catégorie syntaxique).

Le système défini par [Ajdukiewicz \(1935\)](#) et [Bar-Hillel \(1953\)](#) (donc : une grammaire AB catégorielle) constitue la base de la G.C.C. et de toutes les autres variantes de la grammaire catégorielle. Dans une grammaire AB catégorielle, les catégories peuvent se

combiner par l'application de fonctions. Dans des variantes de direction de la grammaire catégorielle, il existe deux versions de l'application de fonctions, en respectant la directionnalité de la barre oblique dans la catégorie syntaxique. Cependant, leur effet sur l'interprétation sémantique est le même :

- **Application en avant** :  $X/Y : f \quad Y : a \Rightarrow X : f(a)$ , pour tout  $X$  suivi par  $Y$  alors on génère la catégorie associée pour  $X$
- **Application en arrière** :  $Y : a \quad X \backslash Y : f \Rightarrow X : f(a)$ , pour tout  $X$  précédé par  $Y$  alors on génère la catégorie associée pour  $X$

La règle d'application en avant dit que si un constituant avec la catégorie  $X/Y$  est immédiatement suivi par un constituant avec la catégorie  $Y$ , elles peuvent se combiner pour former un constituant avec la catégorie  $X$ . De manière analogue, l'application en arrière permet à un constituant  $X \backslash Y$  qui est immédiatement précédé par un constituant avec la catégorie  $Y$  de se combiner avec ce constituant pour former un nouveau constituant  $X$ .

Une chaîne  $a$  est considérée grammaticale si chaque mot de la chaîne peut être classé dans une catégorie (tel que défini par le lexique) afin que les catégories lexicales des mots dans  $a$  puissent être combinées (selon les règles de la grammaire) pour former un constituant. Le processus de combinaison de constituants est appelé une dérivation.

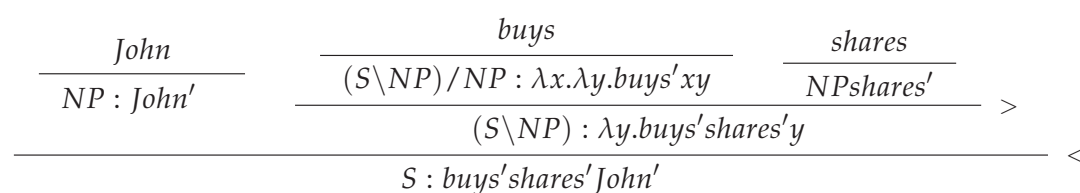


FIGURE 2.1 – Représentation des dérivations G.C.C. en arbres

On note que la structure du constituant induit par cette dérivation correspond à un arbre décrit dans la figure 2.2 :

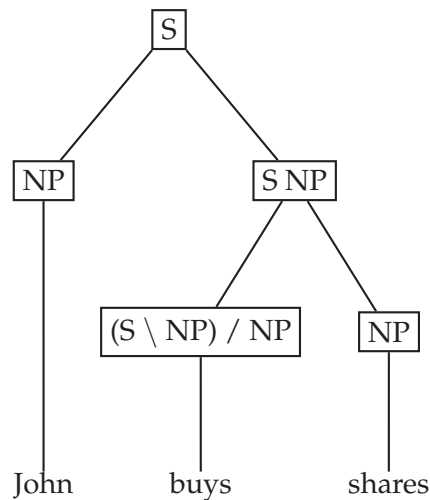


FIGURE 2.2 – Représentation des dérivations G.C.C. en arbres

Les G.C.C. étendent les grammaires catégorielles AB par un ensemble de schémas de règles basées sur les combinateurs de la logique combinatoire Curry et Feys (1958). Ces règles combinatoires permettent une analyse succincte des dépendances à long terme impliquées dans l'extraction et la construction de coordination. Syntactiquement, ils permettent des analyses de l'extraction et de coordonner les constructions qui utilisent les mêmes catégories lexicales pour les têtes de ces constructions comme dans les cas canoniques. Sémantiquement, ils garantissent que les dépendances non-locales remplissent les mêmes emplacements d'arguments que les dépendances locales.

**Le principe des règles combinatoires adjacentes** ne peut s'appliquer qu'à un nombre fini d'entités phonologiquement réalisées et d'entités de chaînes adjacentes. **Le principe de la cohérence de toutes les règles combinatoires syntaxiques** doit être compatible avec la directionnalité de la fonction principale. **Le principe d'héritage** : si la catégorie qui résulte de l'application d'une règle combinatoire est une catégorie de fonction, alors la barre oblique définit la directionnalité d'un argument donné dans cette catégorie qui sera la même que celle(s) définissant la directivité pour le(s) argument(s) correspondant(s) dans la/les fonction(s) d'entrée.

La composition permet à deux catégories de foncteurs de se combiner à partir d'un autre foncteur ; les règles unaires dites de « type-raising » inversent les rôles du foncteur et de l'argument en permettant à une catégorie argument  $X$  de se changer en catégorie foncteur  $T/(T \setminus X)$  ou  $T \setminus (T/X)$ , où  $T \setminus X$  peut être instancié par n'importe quelle catégorie de foncteur qui prend  $X$  en argument.

On observe quatre grands types de compositions :

- Composition en avant (Forward composition) :  
 $X/Y : f \quad Y/Z : g \Rightarrow_B X/Z : \lambda x.f(g(x))$
- Composition croisée en avant (Forward crossing composition) :  
 $X/Y : f \quad Y \setminus Z : g \Rightarrow_B X \setminus Z : \lambda x.f(g(x))$
- Composition en arrière (Backward composition) :





ont les mêmes catégories lexicales que lorsque leurs arguments sont en position canonique. En fait, dans les **G.C.C.**, toutes les dépendances bornées et non-bornées sont projetées à partir du lexique, dans un élément qui est exprimé par les deux principes suivants :

**Le principe de la gouvernance par la tête lexicale :** Les dépendances syntaxiques à la fois bornées et non-bornées sont spécifiées par le type syntaxique de leur tête. **Remarque 1:** Ces dépendances sont établies par l’humain ou apprises par des super-étiqueteurs.

Le principe d’unicité catégorielle de la tête : une seule catégorie lexicale non disjonctive pour la tête d’une construction donnée précise à la fois les dépendances bornées qui se présentent lorsque ses compléments sont en position canonique et les dépendances non bornées qui se présentent lorsque ces compléments sont déplacés dans la relativisation, la coordination, etc.

Comme indiquée ci-dessus, la composition n’est autorisée que dans les fonctions d’un argument. On note que c’est également le cas dans la construction de clusters d’arguments ci-dessus, puisque le deuxième foncteur n’a qu’un seul argument. Toutefois, la composition généralisée est nécessaire pour des phrases comme les suivantes :

$$\frac{I}{NP} \frac{\textit{offered}}{(S \setminus NP) / PP / NP} \textit{and} \frac{\textit{may}}{(S \setminus NP) / PP / NP} \frac{\textit{give}}{(S \setminus NP) / PP / NP} >_{B^2} \frac{\textit{a flower}}{NP} \frac{\textit{to a policeman}}{PP}$$

FIGURE 2.6 – Dérivation de la phrase “I offered and may give a flower to a policeman”

La convention avec le symbole \$ : pour une catégorie  $\alpha$ ,  $\alpha\$$  (respectivement  $\alpha/\$, \alpha \setminus \$$ ), désigne l’ensemble contenant  $\alpha$  et toutes les fonctions (respectivement les fonctions gauches, et les fonctions droites) dans une catégorie  $\alpha\$$  (respectivement  $\alpha/\$, \alpha \setminus \$$ ).

On observe quatre grands types de compositions généralisées :

- Composition généralisée en avant (Generalized forward composition) :  
 $X/Y : f \quad (Y/Z) / \$_1 : \dots \lambda z. gz \dots \Rightarrow_{>B^n} (X/Z) \$_1 : \dots \lambda z. f(g(z \dots))$
- Composition généralisée croisée en avant (Generalized forward crossing composition) :  
 $X/Y : f \quad (Y \setminus Z) / \$_1 : \dots \lambda z. gz \dots \Rightarrow_{>B_x^n} (X/Z) \$_1 : \dots \lambda z. f(g(z \dots))$
- Composition généralisée en arrière (Generalized backward composition) :  
 $(Y \setminus Z) \setminus \$_1 : \dots \lambda z. gz \dots X \setminus Y : f \Rightarrow_{<B^n} (X \setminus Z) \setminus \$_1 : \dots \lambda z. f(g(z \dots))$
- Composition généralisée croisée en arrière (Generalized backward crossing composition) :  
 $(Y \setminus Z) / \$_1 : \dots \lambda z. gz \dots X \setminus Y : f \Rightarrow_{<B^n} (X \setminus Z) \$_1 : \dots \lambda z. f(g(z \dots))$

Chacune de ces règles correspond à une famille de règles pour chaque arité  $n$  du foncteur secondaire. Sans aucune restriction sur l’arité du foncteur secondaire, toute la sensibilité au contexte serait obtenue. Toutefois, il n’y a pas encore eu la preuve que



cela est nécessaire pour capturer la syntaxe du langage naturel. Par conséquent, seuls les schémas jusqu'à une arité  $n$  bornée (Steedman suppose 4 pour l'anglais) sont admis dans la pratique. Les restrictions sur les règles de collecte de type indiqué ci-dessus jouent un rôle similaire. Ensemble, elles conservent une légère sensibilité au contexte. Une autre règle combinatoire, la substitution, est nécessaire pour des lacunes parasitaires, comme l'exemple suivant :

$$\frac{\frac{\frac{\text{articles}}{NP} \quad \frac{\text{that}}{(NP \setminus NP) / (S / NP)}}{NP \setminus NP} \quad \frac{\frac{I}{NP}}{S / (S \setminus NP)} \quad >T \quad \frac{\frac{\text{file}}{VP / NP} \quad \frac{\frac{\text{without}}{(VP \setminus VP) / VP[ng]} \quad \frac{\text{reading}}{VP[ng] / NP}}{(VP \setminus VP) / NP}}{VP \setminus VP} \quad >B \quad < S_x}{S / NP} \quad >B$$

FIGURE 2.7 – Dérivation de la phrase "articles that I file without reading"

Les substitutions sont définies comme suit :

- Substitution en avant (Forward substitution) :  
 $Y \setminus Z : g(X \setminus Y) \setminus Z : f \Rightarrow_S X \setminus Z : \lambda x. fx(g(x))$
- Substitution croisée en avant (Forward crossing substitution) :  
 $(X / Y) \setminus Z : fY \setminus Z : g \Rightarrow_S X \setminus Z : \lambda x. fx(g(x))$
- Substitution en arrière (Backward substitution) :  
 $(X / Y) / Z : fY \setminus Z : g \Rightarrow_S X / Z : \lambda x. fx(g(x))$
- Substitution croisée en arrière (Backward crossing substitution) :  
 $Y / Z : g(X \setminus Y) / Z : f \Rightarrow_S X / Z : \lambda x. fx(g(x))$

Les compositions et les règles de levée de type mènent à un nombre de dérivations alternatives syntaxiquement distinctes mais sémantiquement équivalentes, même pour les phrases ordinaires, les dérivations 2.2 et 2.8 en sont deux exemples :

$$\frac{\frac{\frac{John}{NP : John'} \quad \frac{buys}{(S \setminus NP) / NP : \lambda x. \lambda y. buys' .xy}}{S / NP : \lambda x. buys' x John'} \quad \frac{shares}{NP shares'}}{S : buys' shares' John'} \quad >B \quad >$$

FIGURE 2.8 – Dérivation de la phrase "John buys shares"

En outre, étant donné que chaque règle syntaxique a une opération correspondante sur l'interprétation sémantique, la G.C.C. peut tenir compte du fait que le traitement du langage humain et la compréhension est progressive et opère sur des phrases incomplètes et les constituants. Ces observations conduisent Steedman (2000) à l'hypothèse de la structure syntaxique qui en soi n'est pas un niveau de représentation pour la

théorie linguistique, au lieu des opérations syntaxiques qui fournissent seulement les moyens de générer des interprétations sémantiques. La dérivation syntaxique est alors simplement considérée comme une trace de l'algorithme qui construit des interprétations sémantiques. Steedman montre comment la multiplicité des dérivations peut être utilisée pour rendre compte non seulement pour la compréhension incrémentale de la langue, mais aussi l'interaction de la prosodie et la structure de l'information et des alternances de quantificateurs. Chacune de ces analyses repose sur l'hypothèse d'un type particulier de dérivation et d'un type particulier d'interprétation.

## 2.4 Grammaires stochastiques pour l'interprétation

Les grammaires stochastiques sont étudiées pour la reconnaissance de motifs structurels (voir Fu (1982) pour une revue plus complète). Une approche basée sur les grammaires stochastiques indépendantes du contexte peut être utilisée pour calculer les probabilités des étiquettes de concepts à partir d'arbre syntaxique. Des grammaires à historique Black et al. (1993) sont proposées pour cet objectif.

L'analyse syntaxique sous le contrôle des grammaires stochastiques indépendantes du contexte (S.C.F.G.) est une activité qui implique l'application d'un nombre fini de règles. Les séquences d'application de la règle peuvent être modélisées par une structure à états finis et l'historique des règles appliquées avant une règle donnée peut être résumé par un ensemble fini de paramètres. Des séquences d'applications de règles et leurs probabilités sont considérées dans les grammaires à historiques Black et al. (1993).

L'analyseur à grammaire à historiques proposé par Black et al. (1993), sépare des chaînes de mots sémantiquement pertinentes de celles qui ne le sont pas. La grammaire utilisée par l'analyseur a des règles de réécriture avec des probabilités associées. Prenons un exemple dans lequel  $S$  est le symbole de départ d'une grammaire,  $r_j$  est la  $j^{\text{ème}}$  règle de réécriture utilisée pour l'analyse d'une séquence de mots  $W$ . La probabilité que la grammaire génère une phrase  $W$  avec un arbre d'analyse  $\tau_g(W)$  en utilisant une séquence  $g$  de règles de réécriture est calculée comme suit :  $P(S \rightarrow \tau_g(W)) = \prod_{j=1}^J P_g(r_j | R_1^j)$

où  $R_1^j$  est l'historique de l'application de règles.  $R_1^j$  peut être approximé de différentes manières.

Il est possible d'intégrer les connaissances sémantiques et syntaxiques dans une grammaire hors-contexte (C.F.G.) où les étiquettes sémantiques et les structures sont représentées par des symboles non-terminaux.

### 2.4.1 Le système TINA

L'analyseur linguistique TINA, développé à l'Institut de Technologie du Massachusetts (MIT), suit la même évolution que d'autres analyseurs linguistiques : à l'origine il s'agissait d'un analyseur syntaxique global, mais, par la suite, un comparateur robuste

a été ajouté pour la sauvegarde lorsque l'analyseur global échoue [Zue et al. \(1992\)](#); [Seneff \(1992b\)](#). La grammaire TINA est écrite comme un ensemble de règles probabilistes de réécriture indépendantes du contexte, avec des contraintes, la grammaire convertie automatiquement au moment de l'exécution en une forme de réseau dans lequel chaque nœud représente une catégorie syntaxique ou sémantique [Seneff \(1989\)](#). Les probabilités associées aux règles sont calculées à partir des données d'entraînement, et servent à limiter la recherche au cours de la reconnaissance (sans elles, toutes les analyses possibles devraient être prises en compte). Le comparateur robuste a été obtenu en modifiant la grammaire pour permettre une analyse partielle [Seneff \(1992b,a\)](#). En mode robuste, l'analyseur procède de gauche à droite, comme d'habitude, mais un ensemble exhaustif d'analyses possibles est généré à partir de chaque mot de l'énoncé. Comme dans DELPHI, le comparateur remplit alors les emplacements dans les frames avec ces phrases, en fin de compte pour choisir la meilleure frame. L'aspect inhabituel de ce comparateur robuste, est qu'il exploite les caractéristiques du mécanisme de l'historique construit pour rendre le dialogue possible. Au cours du dialogue, ce mécanisme d'historique permet aux emplacements d'hériter de déclarations précédentes. De même, le comparateur robuste, "se souvient" des emplacements remplis précédemment dans le même énoncé. Un seul ajustement est nécessaire pour rendre le mécanisme approprié pour analyser l'historique de phrases internes : écraser des valeurs d'emplacements a été interdit (il se produit entre énoncés situés dans la version du dialogue).

Après la création du comparateur robuste, il y a eu peu de changements dans la structure de la composante en langage naturel. De nouvelles règles indépendantes du contexte ont été ajoutées pour augmenter la couverture, y compris les règles qui gèrent les faux départs [Zue et al. \(1993\)](#). Dans les versions les plus récentes du système, l'analyseur essaye les N-meilleures hypothèses dans l'ordre ; si aucune d'elles ne peut être analysée, le comparateur robuste est appliqué à l'hypothèse n°1 [Glass et al. \(1995\)](#); [Zue et al. \(1993\)](#). Une variante de la méthode emploie TINA pour recalculer les cinq premières hypothèses produites par le module de reconnaissance, mais cela n'a donné aucun avantage de performance [Glass et al. \(1995\)](#). La plupart des efforts récents sont portés en génie logiciel - l'ensemble du système a été réécrit pour le rendre plus modulaire, efficace et portable - et dans le développement d'un système de dialogue [Glass et al. \(1995\)](#). En fait, deux systèmes de dialogue ont été développés. Le premier permet aux utilisateurs de parler à un écran où s'affiche l'information pertinente pour le dialogue, par exemple, un utilisateur « commandant » un billet d'avion, peut voir que les champs du billet (origine, destination, date, etc) sont remplis par le système, et donc visualiser et corriger les malentendus qui se sont produits. Le second, une version sans affichage est conçu pour fonctionner par téléphone. C'est un système beaucoup plus difficile à construire. Par exemple, si 25 vols possibles satisfont une demande, il est raisonnable de les afficher sur un écran, mais déraisonnable pour le système de les lire. Ainsi, le système sans affichage doit jouer un rôle très actif dans la réduction de demandes des utilisateurs en posant des questions appropriées.

### 2.4.2 Le système HUM

Le modèle de compréhension caché (Hidden Understanding Model, HUM), mis au point chez BBN, est basé sur une approche totalement différente inspirée (mais non formellement équivalente) des Modèles de Markov Cachés [H.M.M. Miller et al. \(1994\)](#). Dans le système HUM, après qu'un arbre d'analyse est obtenu, les probabilités bigramme de chemin partiel vers la racine, sont utilisées étant donné un autre chemin partiel. L'interprétation est guidée par des instructions représentée par un arbre de décision stochastique. Soit  $M$  le sens d'un énoncé, et soit  $W$  la séquence de mots qui véhiculent ce sens. Par une règle Bayésienne, on peut écrire :

$$Pr(M|W) = \frac{Pr(W|M)Pr(M)}{Pr(W)}$$

Pour un  $W$  donné, nous voulons trouver le  $M$  qui maximise  $Pr(M|W)$ , ce qui peut être fait en maximisant  $Pr(W|M)Pr(M)$ , puisque  $Pr(W)$  est fixé.  $Pr(M)$  peut être estimée à partir d'un modèle de langage sémantique qui précise comment le sens des expressions est généré stochastiquement ;  $Pr(W|M)$  peut être estimée à partir d'un modèle de réalisation lexicale qui spécifie comment les mots sont générés, et donnent un sens. Le modèle sémantique de langage emploie des représentations du sens en arborescences structurées : les concepts sont représentés comme des nœuds dans un arbre, avec des sous-concepts représentés comme des nœuds enfants. Chaque nœud terminal est le parent d'un mot ou d'une séquence de mots. On note aussi que, HUM permet l'imbrication arbitraire de concepts.

Par exemple, le concept «FLIGHT» a les sous-concepts possibles suivants AIRLINE, FLIGHT\_NUMBER, ORIGIN, et DESTINATION. ORIGIN et DESTINATION ont comme enfants possibles, les nœuds terminaux (respectivement) ORIGIN\_IND et CITY, et DEST\_IND et CITY. Dans cet arbre de représentation structurée, l'expression «vol United 203 de Dallas à Atlanta» peut être analysé comme :

```
FLIGHT [AIRLINE[United]
  FLIGHT_IND[flight]
  FLT_NUM[203]
  ORIGIN[
    ORIGIN_IND[from]
    CITY[Dallas]]
  DESTINATION[
    DEST_IND[to]
    CITY[Atlanta]]
  ]
]
```

Ceci est une version légèrement modifiée d'un exemple [Miller et al. \(1994\)](#). Les probabilités dans le modèle sémantique de langage sont de la forme  $Pr(state_i|state_{i-1}, context)$ , où le contexte est le concept parent, et dont  $Pr$  est estimé à partir des données. Par exemple, puisque dans le corpus [A.T.I.S.](#), les personnes mentionnent généralement une origine avant une destination, elles précisent presque toujours la destination si elles

mentionnent une ou l'autre, mais parfois omettent l'origine : la probabilité  $Pr(DESTINATION|ORIGIN, FLIGHT)$  sera plus grande que  $Pr(ORIGIN|DESTINATION, FLIGHT)$ .

Le modèle de réalisation lexicale est un modèle de langage bi-gramme augmenté avec des informations sur le concept de parent en cours :  $Pr(word_i|word_{i-1}, concept)$ .

Deux pseudo-mots \*start et \*end marquent le début et la fin d'une phrase, où une phrase est définie comme une séquence de mots. Ainsi,  $Pr(leaving|*begin*, ORIGIN\_IND)$  est la probabilité que « leaving (quitter) » soit le premier mot d'une phrase ORIGIN\_IND. On a vu que trouver le sens  $M$  pour une chaîne de mots  $W$  est équivalente à trouver le  $M$  qui maximise  $Pr(W|M)Pr(M)$ . Dans la pratique, cela se fait en trouvant le chemin de probabilité maximum à travers le réseau constitué de deux modèles combinés (modèle sémantique de langage et le modèle de réalisation lexicale), qui en théorie pourrait demander un temps exponentiel en fonction de la durée des phrases. Heureusement, comme pour le problème de recherche dans la R.A.P. , l'algorithme de Viterbi peut être combiné avec un élagage judicieux pour maintenir les exigences de calcul dans des limites raisonnables Miller et al. (1994, 1995). Dans une première version, HUM a été entraîné sur 6.000 phrases dont la représentation sémantique a été produite par le système DELPHI, qui traitait également le discours et le traitement d'arrière-plan. Dans une version supérieure, 10.000 phrases d'entraînement ont été annotées à la main par quatre étudiants de premier cycle avec une formation préalable Miller et al. (1995). Chaque phrase a été annotée par au moins deux annotateurs, pour assurer l'exactitude et la cohérence. La composante du discours était codée à la main.

L'intégration des scores syntaxiques et sémantiques se fait avec un analyseur isolé dans un analyseur dirigé pour détecter les concepts, Nagai et al. (1994) L'utilisation de grammaire stochastique pour représenter la syntaxe et la sémantique est proposée Stahl et Müller (1995); Schwartz et al. (1997) avec un modèle d'entropie maximum pour l'attachement propositionnel de phrase.

## 2.5 Le modèle à vecteurs d'états cachés de Cambridge

He et Young (2003b) fournissent les considérations suivantes. L'utilisation de règles manuscrites de grammaire sémantique et d'analyseurs a un coût de développement élevé et est fragile parce que l'utilisateur généralement ne sait pas quelles constructions grammaticales sont prises en charge par le système.

Afin de faire correspondre directement des chaînes de mots dans les structures destinées au sens, les modèles statistiques de la production de phrases ont été proposés. Les langages réguliers à états finis, malgré leur puissance générative, sont incapables de représenter des informations structurelles intégrées et en conséquence ne peuvent pas faire des distinctions importantes, par exemple entre le départ et la destination. Les modèles hiérarchiques récursifs avec la puissance des langages hors-contexte comme le HUM, ne sont pas facilement entraînés à partir de données non étiquetées.

Le modèle à vecteurs d'états cachés est basé sur les vecteurs représentant un état.

Chaque vecteur est le résultat d'une action de "pousser" les balises sémantiques à partir du symbole de la racine et se terminant avec le symbole préterminal comme dans la figure 2.9 ci-après :

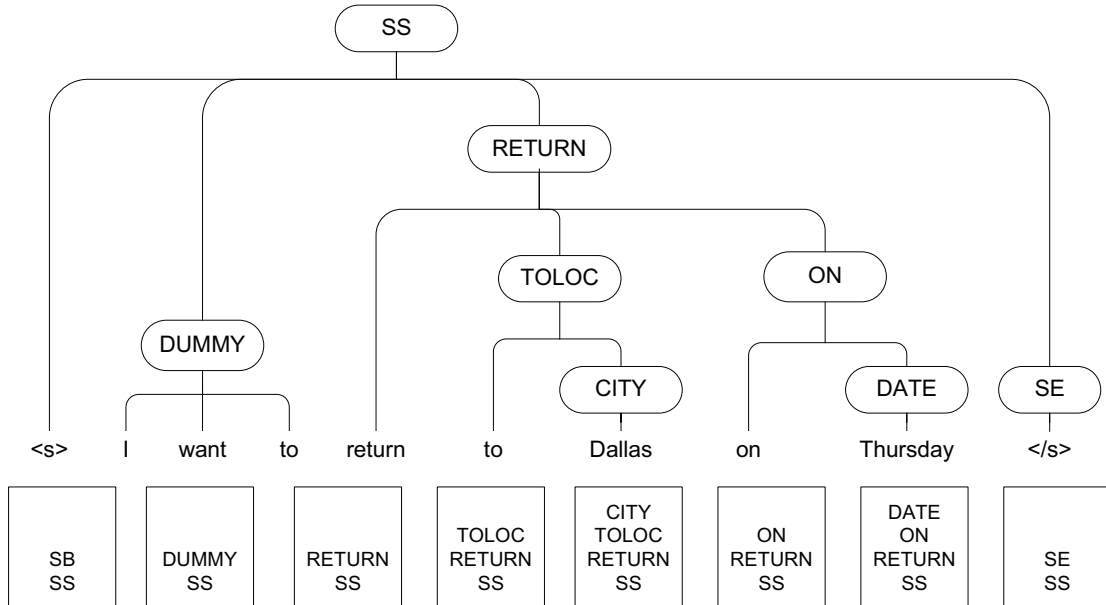


FIGURE 2.9 – Exemple de phrase analysée avec un modèle à vecteurs d'états cachés

Chaque vecteur d'état est vu comme une variable cachée et représente l'état d'un automate de type "push-down". Les transitions entre les états peuvent être prises en compte dans une pile de déplacement suivie par une simple poussée sur un ou plusieurs concepts sémantiques liés à l'entrée suivante.

Chaque opération peut être contrainte et le modèle prend en charge l'intégration, mais tous les arbres d'analyse sont strictement des branchements droits. Il existe trois types de mouvements de probabilités :

1. "pop" : sortir des balises sémantiques de la pile ;
2. "push" : pousser une balise pré-terminale sémantique dans la pile ;
3. générer le mot suivant.

Si  $N$  est une séquence d'opérations "pop" sur la pile,  $C$  est la séquence de vecteurs de balises conceptuelles et  $W$  la séquence de mots, le modèle calcule  $P(W, C, N)$  avec un certain nombre d'hypothèses.

$$P(W, C, N) = \prod_{t=1}^T P(n_t | c_{t-1} []) P(c_t [1] | c_t [2...D]) P(w_t | c_t []) \quad (2.2)$$

où  $c_t$  est le  $t^{ième}$  vecteur d'étiquettes sémantiques obtenu par poussée et déplacement successifs des balises sémantiques.

L'entraînement est réalisé en indiquant des mots-clés dans les classes d'étiquettes sémantiques, par l'utilisation des réseaux bayésiens et en donnant des exemples d'énoncés qui aboutiraient à chaque type de schéma sémantique. En outre, les classes lexicales des domaines spécifiques et l'annotation sémantique abstraite pour chaque énoncé sont fournies.

### 2.6 Conclusions

En conclusion, nous voyons que les grammaires non stochastiques ont une couverture incomplète des énoncés. Les grammaires stochastiques indépendantes du contexte couvrent totalement les énoncés issus de la parole, mais en contrepartie sont limitées dans la sensibilité à décrire le contexte et dans les erreurs d'analyse des phrases parlées souvent agrammaticales.

Des systèmes complexes incluant des grammaires sont décrits dans le chapitre 4.

Une question importante est : avons-nous besoin de l'entière dérivation syntaxique pour obtenir une interprétation sémantique ? La seule chose que les humains perçoivent sont des mots qu'ils comprennent et cela avant d'avoir étudié les méthodes de grammaire formelle.

## Chapitre 3

# Interprétation basée sur l'analyse syntaxique partielle

### Sommaire

---

<b>3.1</b>	<b>Les analyseurs partiels</b> . . . . .	<b>56</b>
<b>3.2</b>	<b>Les modèles génératifs</b> . . . . .	<b>58</b>
3.2.1	Les modèles à états finis . . . . .	58
3.2.2	Interprétation avec des transducteurs à états finis stochastiques . . . . .	60
3.2.3	Les Réseaux Bayésiens Dynamiques . . . . .	65
<b>3.3</b>	<b>Les modèles basés sur un apprentissage discriminant</b> . . . . .	<b>68</b>
3.3.1	Les arbres de classification sémantique . . . . .	68
3.3.2	Les Machines à Vecteurs de Support . . . . .	71
3.3.3	Utilisation de la connaissance préalable . . . . .	72
<b>3.4</b>	<b>Modélisation log-linéaire</b> . . . . .	<b>75</b>
3.4.1	Fonctions caractéristiques . . . . .	75
3.4.2	Les "Conditional Random Fields ( C.R.F. )" . . . . .	76
3.4.3	Normalisation du niveau de position . . . . .	77
3.4.4	L'entraînement . . . . .	77
3.4.5	Processus de décision . . . . .	78
<b>3.5</b>	<b>Utilisation des Conditional Random Fields , pour l'analyse superficielle</b> . . . . .	<b>78</b>
3.5.1	Segmentation avec les Conditional Random Fields . . . . .	80
<b>3.6</b>	<b>Conclusions</b> . . . . .	<b>81</b>

---



### 3.1 Les analyseurs partiels

La génération d'hypothèses sémantiques à partir du texte ou de la parole reste une tâche difficile pour laquelle différentes approches ont été proposées depuis les premières tentatives dans les premiers projets.

Il y a de nombreuses raisons pour lesquelles l'analyse complète n'est pas possible pour de nombreuses phrases prononcées en particulier si les mots sont émis en hypothèse par un système *R.A.P.*. En outre, le but ultime de l'interprétation sémantique automatique est de mettre à jour l'état de la connaissance du système sur le monde extérieur. A cet effet, l'interprétation peut être effectuée sans une analyse syntaxique complète.

Les premières expériences de compréhension du langage parlé ont montré que de nombreuses phrases prononcées n'appartiennent pas à un langage engendré par une grammaire formelle et peuvent contenir des disfluences telles que les répétitions et les auto-corrrections. En outre, le processus d'interprétation utilise des connaissances qui sont incomplètes ou imprécises avec des hypothèses de mots qui peuvent ne pas être correctes. Afin de surmonter en partie les problèmes découlant de l'analyse de l'imprécision et d'erreurs, l'analyse partielle et d'autres méthodes ont été proposées pour relier les fragments de structures sémantiques avec les modèles des mots et des abstractions syntaxiques.

Dans *Woods et al. (1976)*, les analyseurs à grammaires à réseaux de transitions augmentés (*G.R.T.A.*) partiels ont été proposés pour réaliser l'analyse de la transcription automatique d'une phrase. Une spécification du champ d'exploration est associée à des actions de grammaire. L'analyse peut procéder vers la gauche ou vers la droite du mot d'entrée. La spécification du champ indique un ensemble d'états que l'analyseur doit avoir traversé avant que l'action puisse être exécutée en toute sécurité. Si ce n'est pas le cas, alors l'action est retardée.

Pour certaines applications du domaine, les connaissances sémantiques et syntaxiques peuvent être intégrées dans une grammaire de base unique, dans laquelle les constituants sémantiques sont des symboles non-terminaux. Chacun d'eux identifie un sous-réseau qui est juste une grammaire à réseaux de transitions (*G.R.T.*) avec seulement des symboles syntaxiques non-terminaux et des mots. Dans le système Phoenix *Young et al. (1989)*; *Ward (1990)*, les *G.R.T.* de ce type sont proposées et utilisées dans les attachements de procédure pour les emplacements de frames. Chaque *G.R.T.* peut être activée sous le contrôle de la stratégie d'interprétation pour la formulation d'hypothèses des fragments d'analyse sémantique partielle.

La principale raison pour laquelle les probabilités sont utilisées dans l'analyse syntaxique est la possibilité de donner des priorités aux différents arbres d'analyse possibles d'une phrase. Le calcul des probabilités d'interprétations sémantiques dans la *C.L.P.* a un objectif différent. En fait, ces probabilités devraient évaluer la plausibilité que les structures hypothèses conceptuelles aient été exprimées par les phrases analysées. Pour ces raisons, il est intéressant de voir l'interprétation comme l'analyse des

morceaux extraits d'une phrase.

Les tâches d'analyse de séquences dans la langue et dans la biologie sont souvent décrites comme des correspondances à partir de séquences d'entrée à des séquences d'étiquettes encodant l'analyse. Dans le traitement du langage, les exemples de ces tâches comprennent le balisage des parties du discours, la reconnaissance d'une entité nommée, et l'analyse superficielle. L'analyse superficielle identifie les noyaux non-récursifs de différents types de phrases dans le texte, peut-être comme précurseur à l'analyse complète ou à l'extraction d'information [Abney \(1991\)](#). Des grammaires simples sont utilisées pour détecter d'éventuelles clauses, puis l'analyse basée sur la classification complète l'analyse avec inférence [Kasper et Hovy \(1990\)](#).

Les analyseurs sémantiquement dirigés utilisent la correspondance de modèle, pour assortir les supports pour hypothèses sémantiques avec les données. La correspondance peut commencer avec des modèles lexico-sémantiques pour instancier des éléments lexicaux initiaux. Les interprétations sont construites en ajoutant des éléments non lexicaux inférés par un algorithme de recherche [Stallard et Bobrow \(1993\)](#). Différents types de comparateurs peuvent être conçus à des fins différentes. Les comparateurs de modèles à états finis, les analyseurs lexicaux et comparateurs de modèles au niveau phrase sont discutés dans [Hobbs et Israel \(1994\)](#).

Le système développé par le Stanford Research Institute (SRI), se composait de deux modules sémantiques reliés ensemble : une unification de grammaire à base de module appelé "Gemini", et le comparateur de motifs ou Template Matcher (TM), qui agit comme un repli si "Gemini" ne peut pas produire une interprétation acceptable pour exécuter une requête sur une base de données [Jackson et al. \(1991\)](#).

L'entrée du TM est la séquence d'hypothèse de mot la plus probable générée par le composant [R.A.P.](#) . Le TM essaye simplement de combler les emplacements en cherchant des phrases courtes dans les modèles de simili-frames . Les différents modèles sont en concurrence les uns avec les autres sur chaque énoncé, tous sont évalués, et le modèle avec le meilleur score génère l'interprétation. Les scores d'instanciation sont les pourcentages de mots corrects.

Comme les bons analyseurs pour les phrases nominales sémantiquement importantes peuvent être codés à la main relativement rapidement, il est possible de dériver manuellement des motifs pour la génération avec un concept TM, des hypothèses à partir de phrases nominales.

Comme la correspondance de motifs est généralement effectuée en utilisant des motifs de mots qui peuvent être représentés par des expressions régulières, il est apparu intéressant d'étudier des modèles génératifs pour les langues exprimant des constituants sémantiques ou un étiquetage de concepts en utilisant des modèles statistiques à états-finis.

Afin de réduire le risque d'erreurs dans la représentation [L.R.S.](#) d'une phrase parlée, l'interprétation machine est conçue comme un processus de décision guidée par un critère d'optimalité. En plus de l'analyse partielle et de back-off, dans le projet "Air Travel Information System" ([A.T.I.S.](#)), réalisé dans les années 1990 et examiné au chapitre

14 de [De Mori \(1998\)](#), il a été jugé utile pour caractériser l'imprécision du modèle avec des probabilités. Il a également été jugé utile d'obtenir des paramètres de distribution de probabilités par un processus piloté par les données d'apprentissage automatique, utilisant des corpus manuellement annotés. Avec ces modèles, une démarche d'interprétation calcule la probabilité d'un résultat fondé sur des hypothèses incertaines et sur l'imprécision de la connaissance.

Par souci de simplicité, nous supposons qu'une séquence d'hypothèses de concepts  $C_1^K = c_1, \dots, c_k, \dots, c_K$  est exprimée par un locuteur dans une séquence  $W$  de mots qui est hypothétisée par le système de [R.A.P.](#) basé sur des paramètres acoustiques  $Y$ . Cela peut être représenté comme suit :  $Y \rightarrow_e W \rightarrow_e C_1^K$ . Le symbole  $\rightarrow_e$  indique une relation de preuve qui signifie que si une séquence d'observation acoustique  $Y$  est observée, alors il y a des preuves de  $W$  et, de ce fait, il existe des preuves de  $C_1^K$ .

Différentes séquences de concepts peuvent être exprimées par une séquence  $W$ . La séquence  $Y$ , peut générer d'autres hypothèses  $W'$  qui peuvent exprimer  $C_1^K$  comme d'autres séquences de concepts. La présence de  $C_1^K$  dans le message parlé, décrit par des paramètres acoustiques  $Y$ , peut seulement être affirmée avec une probabilité calculée de la manière suivante :

$$\begin{aligned}
 P(C_1^K|Y) &= \frac{1}{P(Y)} \left[ \sum_j P(Y|C_1^K W_j) P(C_1^K W_j) \right] \\
 &\approx \frac{1}{P(Y)} \left[ \sum_j P(Y|W_j) P(C_1^K W_j) \right]
 \end{aligned}
 \tag{3.1}$$

Un treillis d'hypothèses de mots peut être obtenu avec un système de [R.A.P.](#) à partir duquel un treillis d'hypothèses de concepts, contenant des structures de concepts, peuvent être générées comme décrit dans [Raymond et al. \(2006\)](#). Les séquences d'hypothèses de concepts pour une phrase ou un tour du dialogue peuvent être extraites du treillis de concepts. Soit  $C$  indiquant une séquence de concepts hypothèses correspondant à la séquence de mots  $W$ .

$P(Y|W)$  est le score acoustique de  $W$ ,  $P(W|C)$  est obtenu avec un modèle de langage conceptuel ([M.L.C.](#)) qui est un modèle génératif de mots exprimant des concepts dans  $C$  et  $P(C)$  est obtenu avec un modèle de concept.

La liste des N-meilleures séquences  $W$  de mots d'hypothèses générées par la [R.A.P.](#) peut être utilisé à cette fin.

## 3.2 Les modèles génératifs

### 3.2.1 Les modèles à états finis

Les motifs utilisés pour détecter des constituants sémantiques dans la même phrase peuvent partager des composants et peuvent capturer les dépendances du contexte,

mais elles sont de longueur finie parce que les phrases, surtout parlées, contiennent un ensemble fini de mots et souvent en petit nombre. Les modèles à états finis sont donc appropriés pour représenter de façon concise ces modèles.

Dans [De Mori et al. \(1991\)](#), il est constaté que les probabilités des modèles de langage à états finis peuvent être adaptés aux attentes par l'historique des paroles déjà prononcées ou des états du dialogue parlé.

Un dialogue est une séquence de tours, alternant la prise de parole des protagonistes. Supposons que, au tour  $\tau$ , le dialogue est dans un état  $S_\tau$ .  $S_\tau$  est déterminé par les valeurs des variables logiques qui sont liées au tour  $\tau$ . Les modèles de langage dépendant des états du dialogue sont considérés. Ils sont obtenus par l'adaptation d'un modèle de langage générique à une situation spécifique dans laquelle certains concepts ont une probabilité plus élevée d'apparaître dans un futur état  $S_{\tau+1}$ .

Soit  $P_S(W_i|W_{i-n+1}^{i-1})$ , un n-gramme de distribution de probabilités pour un Modèle de Langage ( M.L. ) dépendant de l'état  $S = S_\tau$ . Soit  $P_j(W_i|W_{i-n+1}^{i-1})$  la distribution de probabilités d'un M.L.C. correspondant au constituant sémantique  $C_j$ . Soit  $P_S(C_j)$  la probabilité que ce concept  $C_j$  soit exprimé par un sujet parlant au système dans l'état  $S$ . Puis :

$$P_S(W_i|W_{i-n+1}^{i-1}) = \sum_{j=1}^J P_S(C_j)P_j(W_i|W_{i-n+1}^{i-1}) \quad (3.2)$$

Le calcul peut être étendu au cas dans lequel un dialogue est dans différents états avec une distribution de probabilités au cours de ces états, à un tour donné.

Les automates stochastiques à N-grammes variables et leur utilisation pour l'hypothétisation de constituants sémantiques sont proposés dans [Nasr et al. \(1999\)](#). Il propose des "transducteurs stochastiques à états finis" ou "stochastic finite state transducers" ( S.F.S.T. ) composés d'un nombre variable de modèles de langage et d'un modèle de langage général. Chaque modèle local est utilisé pour calculer la probabilité d'une séquence de mots sachant la classe d'une phrase. Formellement, un modèle local est représenté comme un automate à états finis pondéré (WFSA) dont les arêtes sont étiquetées avec les mots. Un parcours dans le WFSA représente une expression de la classe.

Les états spécialisés du dialogue [De Mori et al. \(1991\)](#) dépendant des modèles de langages sont construits pour les constituants sémantiques pour les fonctions et les arguments ou structures [Kuhn et De Mori \(1995\)](#). Un exemple de M.L. basé sur des automates à états finis stochastiques peut être trouvé dans [Prieto et al. \(1994\)](#). Les phrases appartenant à ces langages sont acceptées par des S.F.S.T. qui renvoient des étiquettes pour les constituants sémantiques [Riccardi et al. \(1997\)](#); [Kawahara et al. \(1999\)](#); [Nasr et al. \(1999\)](#).

D'autres modèles génératifs sont basés sur des transducteurs stochastiques à états finis. Les topologies des S.F.S.T. peuvent être conçues compte tenu des approximations de grammaires finies hors contexte telles que proposés dans [Pereira \(1990\)](#), où des approximations de grammaires d'arbres adjoints "Tree Adjoining Grammars" (TAG) comme décrit dans [Rambow et al. \(2002\)](#). L'estimation des distributions de probabilités

des S.F.S.T. utilisée pour la compréhension du langage naturel est décrite dans Prieto et al. (1994).

### 3.2.2 Interprétation avec des transducteurs à états finis stochastiques

Les modèles de langage spécialisés ( M.L. ) sont construits pour les constituants sémantiques ou des fonctions et des arguments ou des structures. On les appelle des modèles de langage conceptuel ( M.L.C. ).

Les phrases appartenant à ces langages sont acceptées par des transducteurs stochastiques à états finis (Stochastic Finite State Transducers ( S.F.S.T. )), qui produit des étiquettes pour les constituants sémantiques. Une utilisation concurrente de modèles de langage générique et de modèles de langage dépendant de concepts est possible. Les résultats de l'application de différents modèles peuvent être comparés et utilisés pour définir les situations de fiabilité sur la base de laquelle les actions de dialogue peuvent être décidées. Ces situations peuvent également être définies sur la base des descriptions des mesures de confiance acoustique, linguistique et sémantique.

Un M.L.C. peut être considéré comme un langage accepté par un S.F.S.T. qui est une approximation à états finis d'une grammaire du langage naturel.

Le S.F.S.T. peut contenir des séquences très courtes de mots (un seul mot) qui peuvent indiquer la présence de plus d'une structure conceptuelle. Ces chemins ambigus peuvent être inclus dans des chaînes de mots plus complexes ce qui exprime de façon non ambiguë une seule structure conceptuelle. Selon le type de séquence de mots  $W$  qui fait partie des hypothèses générées par un système R.A.P. , une ou plusieurs structures conceptuelles  $T$  sont générées avec une probabilité  $P(T|W)$ .

L'inférence de composition et la mise en hypothèses de constituants sont des opérations très différentes. En particulier, il n'est pas évident de savoir comment les concepts sont liés aux mots. En outre, la connaissance pour l'interprétation doit contenir des modèles de séquences de mots pour chaque composant conceptuel. Les modèles utilisés pour détecter des constituants différents dans la même phrase peuvent partager des composants et peuvent capturer des dépendances de contexte, mais les phrases sont de longueur finie parce que les phrases, surtout parlée, contiennent un nombre fini de mots, souvent petit. Les modèles à états finis sont donc appropriés pour représenter de manière concise ces modèles.

Au contraire, les relations sémantiques peuvent utiliser l'hypothèse émise sur les composants dans différentes phrases et générer des structures qui peuvent appartenir à un langage sensible au contexte. Des compositions des constituants conceptuels peuvent avoir à satisfaire des contraintes qui diffèrent des contraintes imposées sur les mots exprimant un composant conceptuel. En outre, les relations sémantiques sont indépendantes de la langue, alors que les relations entre les constituants conceptuels et les mots sont dépendantes de la langue. La composition est intégrée à la stratégie du dialogue, parce que les actions de dialogue peuvent avoir pour objectif d'obtenir tous les éléments d'une demande d'un service. La première activité de compréhension à être

effectuée à chaque tour de dialogue est de détecter tous les composants conceptuels qui sont codés dans l'énoncé en cours d'analyse.

Suivant les considérations ci-dessus, si  $c_1$  et  $c_2$  sont deux constituants basiques conceptuels ou d'étiquettes de concepts, les probabilités  $P(c_1|Y)$  et  $P(c_2|Y)$  peuvent être difficiles à comparer, car elles sont émises en hypothèses avec différents supports d'hypothèses de mots. Soient  $W_1$  le support pour  $c_1$ , et  $W_2$  le support pour  $c_2$ . Pour estimer la plausibilité que les concepts sont codés dans le signal de parole décrit par les caractéristiques  $Y$ , une mesure de confiance générale  $\Psi(c_j)$  peut être dérivée pour chaque étiquette de concepts  $c_j$ . Les décisions sur la présence de  $c_j$  dans le message oral, peuvent être prises en se basant sur la probabilité  $P(c_j|\Psi(c_j))$ . La mesure de confiance  $\Psi(c_j)$  peut être calculée à partir des probabilités  $P(Y|W_j)$ ,  $P(W_j)$ ,  $P(c_j|W_j)$ .

Un système de dialogue a une croyance qui génère des espérances  $B$  sur les structures conceptuelles. Soit  $CTS : \{c_1, \dots, c_j, \dots, c_J\}$  un ensemble d'étiquettes de concepts. Un transducteur stochastique à états finis **S.F.S.T.** est conçu pour chaque concept  $c_j$ , on l'identifie par  $SFST_j$ . La définition de ces constituants doit satisfaire deux contraintes majeures, on doit pouvoir inférer un modèle de langage pour chaque constituant  $c_j$ , et il doit être possible de composer ou d'inférer n'importe quelle structure de connaissance sémantique à partir de l'ensemble des constituants.

Un **S.F.S.T.** peut être une concaténation de modèles de phrase (**M.P.**) et d'outils de remplissage. Ces outils implémentent des modèles de langages génériques alors que les **M.P.** acceptent des phrases utiles pour l'interprétation.

Les modèles de langages pour les **S.F.S.T.** peuvent être construits avec des procédures partielles ou totalement automatiques. Dans de tels cas, ils peuvent être réutilisés ou adaptés à de nouveaux domaines. Différents automates, représentant différents concepts, peuvent partager des phrases, permettant aux interprétations sémantiques de partager des parties de mots transportés par le signal de parole, sans requérir comme d'autres approches populaires, à une analyse sémantique de surface, et que les segments sémantiques ne chevauchent pas [Hacioglu \(2004\)](#); [Pradhan et al. \(2004\)](#).

En outre, les hypothèses de concepts basées sur le même mot dans une phrase peuvent coexister et il n'est pas nécessaire d'intégrer les symboles d'hypothèses sémantiques avec des étiquettes et des mots en une séquence unique pour le calcul des probabilités n-grammes de symboles. Une activité initiale **R.A.P.** utilise des modèles de langage génériques, indiqué comme **GEN.L.M.**, pour générer un graphe  $WG$  d'hypothèses de mots. Un automate  $AWG$  est dérivé de ce graphe. Une séquence  $W$  d'hypothèses de mots est calculée par la vraisemblance  $L(Y, W)$ , où  $Y$  représente les observations acoustiques.

Un **S.F.S.T.** spécial, indiqué comme  $SFST_0$ , implémente un modèle de langage qui est obtenu en soustrayant tous les modèles de langages conceptuels du **M.L.** générique. Il y a un **S.F.S.T.** ( $SLM_c$ ) pour chaque concept élémentaire  $c$ . La définition de ces constituants doit satisfaire deux contraintes majeures. Il doit être possible d'inférer un modèle de langage pour chaque constituant et il doit être possible d'inférer n'importe quelle structure de connaissance sémantique à partir d'un ensemble de constituants.



Une source de connaissances est construite en connectant tous les *S.F.S.T.* en parallèle. Un *S.F.S.T.* indiqué comme  $SLM_0$  est introduit pour représenter toutes les séquences de mots qui ne contiennent pas les expressions des composants conceptuels.

Une telle source de connaissances est composée d'un graphe d'hypothèses mot WG conduisant à un graphe SEMG, l'opérateur  $\circ$  indique la composition :

$$\begin{aligned} SEMG &= WG \circ \left( \bigcup_{c=0}^C SLM_c \right) \\ SLM_0 &= GEN.L.M. \circ \left( \bigcup_{c=1}^C SLM_c \right) \end{aligned} \quad (3.3)$$

Les arcs des SEMG sont étiquetés avec des paires de symboles. Le premier élément de la paire est un mot  $w$  qui est associé avec sa vraisemblance  $\log(P(Y(w), w))$ , où  $Y(w)$  est la séquence de vecteurs de caractéristiques acoustiques associée à  $w$  dans WG. Le second symbole de la paire peut être soit un symbole vide, soit le début ou la fin d'une étiquette sémantique. Une étiquette sémantique représente n'importe quel constituant ou structure sémantique pour laquelle une relation avec un motif de mot a été identifiée.

Le support du concept  $c_j$ ,  $sup(c_j)$  est l'union de tous les chemins partant de la source jusqu'au puits du WG et qui incluent la séquence de mots correspondant aux PMs dans le *S.F.S.T.* d'étiquettes de concepts. Des phrases pour différents concepts peuvent se superposer. En fait, les mots exprimant le sens d'une fonction peuvent être essentiels ou très utiles pour évaluer qu'un nom propre qui indique sans ambiguïté "un carré" plutôt qu'un autre type d'entité, par exemple. Un certain niveau de redondance peut aussi être très utile pour compenser les erreurs de la *R.A.P.* .

Une structure des données est créée avec des arcs de début et de fin d'une phrase supportant une interprétation et sa structure sémantique correspondante. Les arcs dans les graphes composés pointent vers les éléments de cette structure.

Bien que les SEMG puissent être utilisés directement par le moteur d'un dispositif de reconnaissance ASR en un seul passage, la deuxième étape proposée ici, a un impact négligeable sur la vitesse globale parce que les opérations sur les automates et transducteurs sont très rapides.

Si plus d'un concept est émis en hypothèse dans la même phrase et les concepts hypothétiques peuvent être composés pour partie d'une structure sémantique, alors l'intersection de leurs supports est effectuée. Si l'intersection n'est pas vide, alors la composition est validée.

Chaque structure sémantique  $\Gamma$  obtenue par composition est calculée avec la probabilité postérieure suivante :

$$s_w(\Gamma) = P(W(\Gamma)|Y) = \sum_{W' \in support(\Gamma)} P(W'|Y) \quad (3.4)$$

Afin d'obtenir les hypothèses conceptuelles qui sont les plus susceptibles d'être exprimées par l'énoncé analysé, SEMG est projetée sur ses résultats (les deuxièmes éléments des paires associées aux transitions) conduisant à une machine à états finis pondérée (F.S.M.), avec seulement des indicateurs de début et fin de mots balises sémantiques. Le F.S.M. qui en résulte est ensuite déterminisé et minimisé conduisant à une F.S.M. SWG donnée par :  $SWG = OUTPROJ(SEMG)$  où  $OUTPROJ$  représente les opérations de projection des sorties. La probabilité  $P(w_1^N, c_1^N)$  calculée est :

$$P(w_1^N, c_1^N) = \prod_{n=1}^N P(w_n c_n | h_n) \quad (3.5)$$

avec  $h_n = \{w_{n-1}c_{n-1}, w_{n-2}c_{n-2}\}$  (3-gram)

Une projection de sortie représentant une hypothèse sémantique  $G$  est calculée avec  $s_w(\Gamma)$ . Les hypothèses sémantiques peuvent alors être classées et seulement la N-meilleure liste est conservée pour un processus ultérieur. En conséquence, l'ensemble  $NB(WG)$  de la N-meilleure séquence de mots pour chacune des hypothèses conceptuelles est obtenu. On note que la somme de 3.4 tend à mettre sur le haut de la liste, les interprétations qui ont un support dans le  $WG$ , qui peut contenir différentes séquences de mots portant la même interprétation.

Les phrases de  $NB(WG)$  peuvent être recalculées en utilisant un langage adapté et des modèles sémantiques. Il est possible, pour le recalcul, d'utiliser différents types de modèles de langue. Si le recalcul commence à partir d'un treillis de phonèmes, alors il est également possible de mettre à jour le lexique en fonction de ce qui est dans  $NB(WG)$ .

En général le score d'une structure conceptuelle  $\Gamma$  est calculé en considérant la meilleure séquence de mots  $W$  dans  $WG$  qui contient un support pour  $\Gamma$ .

$$s(\Gamma, Y) = \log(P(Y|W)) + \log(P(\Gamma|WB)) + \log(P(W|B)) + \log(P(B)) \quad (3.6)$$

Voici un exemple court tiré du papier [Raymond et al. \(2006\)](#) qui décrit l'utilisation des transducteurs pour l'interprétation sémantique. Lors du traitement d'un énoncé, le module [R.A.P.](#) du système de dialogue sort un treillis de mots codé comme un automate  $G_w$ . Une fonction de pondération  $w(\pi)$  est utilisée pour donner un poids sur chaque chemin  $\pi$ ,  $w(\pi) = \log(P(Y|W)P(W))$ , où  $Y$  est la séquence des observations acoustiques,  $W$  la chaîne de mots représentant le chemin  $\pi$ ,  $P(Y|W)$  la probabilité donnée à un modèle de langage bi-gramme. Un exemple est donné par la figure 3.1. Les scores de vraisemblance sont attachés à chaque transition.

$G_w$  est composé avec un transducteur  $T_{concept}$  pour obtenir un transducteur mot-concept  $T_{WC} : T_{WC} = G_w \circ T_{concept}$  illustré par la figure 3.2.

Un chemin  $path(I, x, y, F)$  dans  $T_{wc}$  (avec  $I$  l'état initial et  $F$  un état final de  $T_{wc}$ ) est soit une chaîne de mots si l'on considère les symboles d'entrée  $x$  soit une chaîne de concepts si l'on considère les symboles de sortie  $y$ . Afin d'obtenir toutes les interprétations possibles contenues dans  $G_w$ , il faut projeter  $T_{wc}$  sur les symboles de sortie, puis



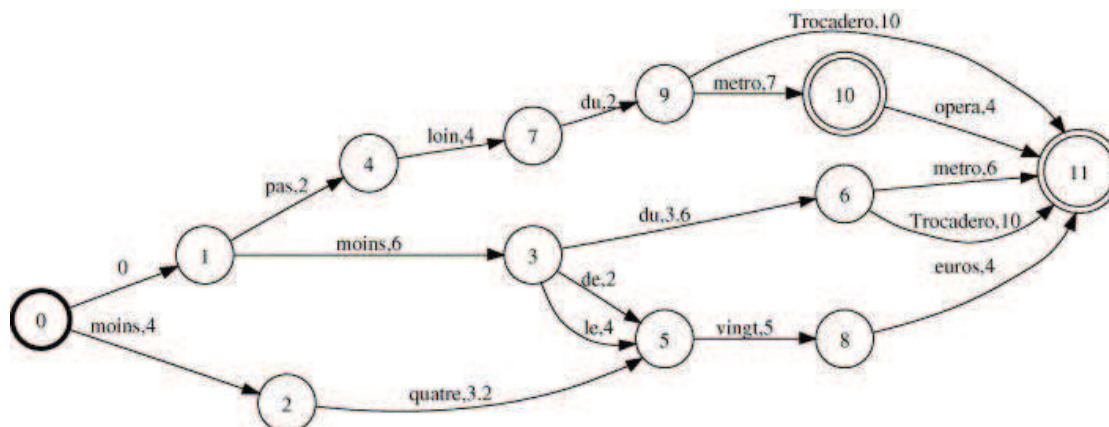


FIGURE 3.1 – Exemple de treillis de mots  $G_W$  issu d'un module de reconnaissance de la parole

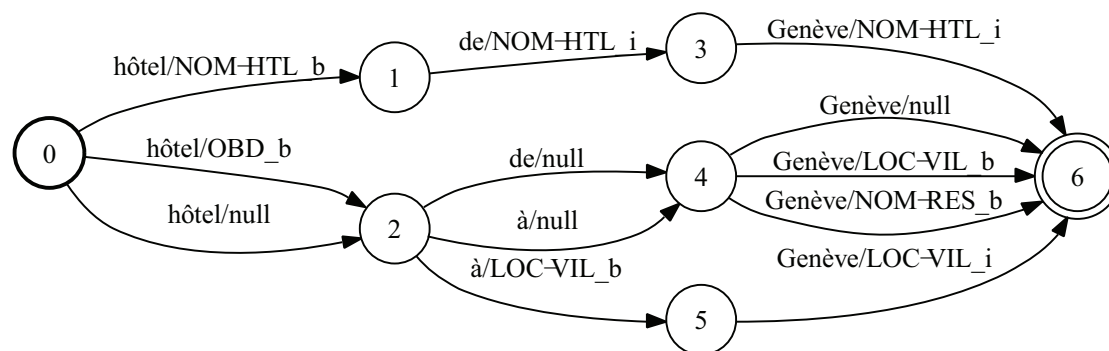


FIGURE 3.2 – Exemple de transducteur mot-concept  $T_{WC}$  correspondant à la composition de l'automate  $G_w$  avec le transducteur  $T_{concept}$

déterminer et minimiser l'automate à états finis qui en résulte. L'automate obtenu est appelé  $G_C$ .

La liste des  $N$  meilleures interprétations conceptuelles  $I_1, I_2, \dots, I_n$ , est obtenue à partir de  $G_C$ , on énumère les  $N$  meilleurs chemins  $S_1, S_2, \dots, S_n$  dans  $G_C$ . Chaque interprétation  $I_i$  est une série d'étiquettes  $y$  représentée par un automate  $S_i$ . Pour chaque  $I_i$  est également joint un automate  $G_{w_i}$  qui est l'ensemble des chemins de  $T_{WC}$  qui produit la chaîne  $I_i$  :  $[G_{w_i}] = [T_{wc} \circ S_i]$ . Ceci est représenté dans la figure 3.3.

Une fois que les constituants conceptuels ont été émis en hypothèse pour une phrase prononcée, les relations sémantiques qui leur sont applicables, doivent être identifiées et utilisées pour instancier les structures sémantiques composites et effectuer des inférences.

Un modèle de langage **M.L.** spécifique, appelé **M.L. conceptuel**, peut être conçu pour chaque balise concept. Les **M.L. conceptuels** peuvent être combinés avec un **M.L. générique** et utilisés directement dans le processus de décodage de la **R.A.P.** . Cette méthode est particulièrement utile lorsque les messages du discours sont analysés et contiennent des segments hors de domaine. Ces segments sont émis en hypothèse lors

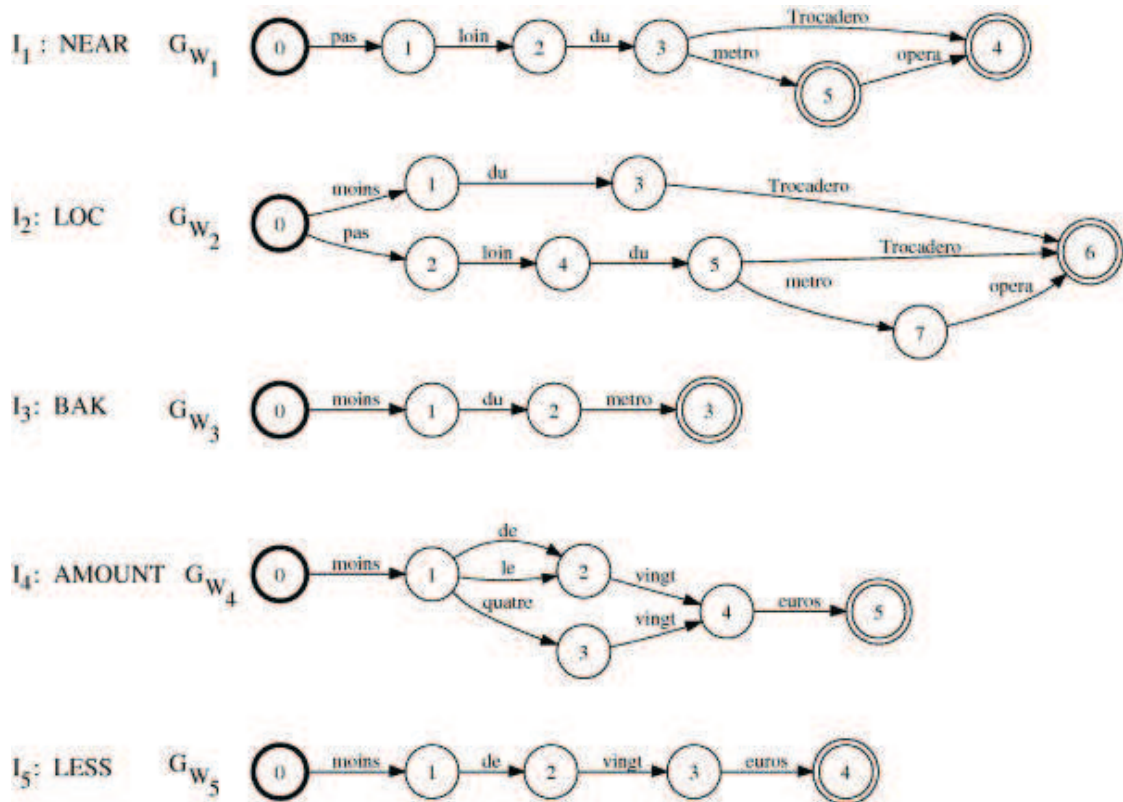


FIGURE 3.3 – Liste des  $N$  meilleures interprétations conceptuelles basiques  $I_i$  avec leur automate  $G_{w_i}$  correspondant

du décodage et ignorées dans le traitement ultérieur. Les détails peuvent être trouvés dans [Camelin et al. \(2006\)](#); [Damnati et al. \(2007\)](#).

Dans [Camelin et al. \(2010\)](#) la preuve est apportée que des améliorations ont été obtenues dans l'extraction d'opinion de systèmes de sondage téléphonique en utilisant des modèles de langage conceptuels probabilistes dans le processus de décodage R.A.P. . Des modèles génératifs basés sur les réseaux bayésiens dynamiques ( D.B.N. ) ont également été proposés. Les détails de ces modèles, leur formation et leur utilisation peuvent être trouvés dans [Lefèvre \(2007\)](#).

### 3.2.3 Les Réseaux Bayésiens Dynamiques

Au cours des dernières années, les réseaux bayésiens dynamiques (Dynamic Bayesian Networks ( D.B.N. )) ont été étudiés pour de nombreuses tâches séquentielles de modélisation des données, telles que la reconnaissance automatique de la parole, parties du discours (Part of Speech ( P.O.S. )) ou l'étiquetage des actes de dialogue [Ji et Bilmes \(2006\)](#), l'analyse de séquences d'ADN. Les D.B.N. ont montré qu'ils fournissent une grande souplesse pour la représentation du système stochastique complexe avec de bonnes performances par rapport aux autres techniques utilisées plus largement

comme indiqué dans Lefèvre (2006, 2007).

Un système de compréhension automatique de la parole basé sur les D.B.N. exécute le processus de compréhension comme une procédure purement stochastique avec l'introduction d'une phase de normalisation de valeur stochastique qui était basée auparavant sur des règles Lefèvre (2006). La mise en hypothèse des concepts  $C$  et des valeurs  $V$  est effectuée comme suit :

$$\hat{C} = \arg \max_C \sum_V P(W|C, V)P(V|C)P(C) \quad (3.7)$$

$$\hat{V} = \arg \max_V P(W|\hat{C}, V)P(V|\hat{C})P(\hat{C}) \quad (3.8)$$

Les D.B.N. sont utilisés ici dans un procédé de décodage multi-étapes. D'abord les concepts sont hypothésés basés uniquement sur la séquence de mots, ensuite leurs valeurs sont estimées compte tenu de la séquence de mots et de la séquence de concepts hypothésés.

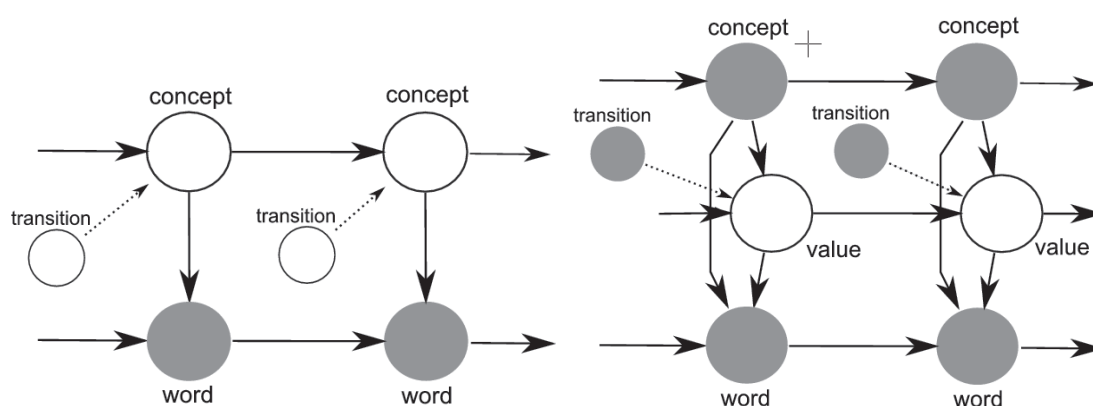


FIGURE 3.4 – Exemple de modèles de D.B.N.

La figure 3.4 montre deux modèles de génération D.B.N. utilisés dans le système de Compréhension Automatique de la Langue Parlée (C.L.P.). Par souci de simplicité, certains sommets supplémentaires (variables) et les arêtes (la dépendance conditionnelle) de la D.B.N. réelle ne sont pas représentés. Dans la figure, seuls deux tranches de temps (correspondant à deux mots) sont représentées. Dans la pratique, le modèle régulier (chunk-bloc) est répété de manière à s'adapter à la séquence de mots étudiée. Les nœuds remplis sont des variables observées tandis que les vides sont cachés. Les lignes simples représentent les dépendances conditionnelles entre les variables; les lignes pointillées indiquent les parents de commutation (variables qui influent sur la relation). Un exemple d'un parent de commutation est donné par le nœud de transition qui influe sur le concept de nœud : quand la transition est nulle, le concept est une simple copie de la notion précédente, mais quand il est mis à 1 la valeur du nouveau concept est déterminée en conséquence par  $P(c|c - 1)$ .

Dans notre contexte, toutes les variables sont observées durant l'entraînement, donc pas d'itérations nécessaires pour estimer les paramètres. Les tables de probabilité conditionnelle des arêtes peuvent être dérivées directement à partir des données d'observation. Toutefois, afin d'améliorer leurs estimations, les modèles de langage factorisés (Factorised Language Models **F.L.M.**) ont été utilisés avec un "backoff parallèle généralisé" [Bilmes et Kirchhoff \(2003\)](#). Les **F.L.M.** sont une extension de la norme des modèles de langage **M.L.** où la prévision est basée sur un ensemble de fonctionnalités (et pas seulement sur les occurrences précédentes de la variable à prédire). Pour compléter ce nouveau cadre, le "backoff parallèle généralisé" permet d'étendre les procédures backoff standard au cas où des types d'entités hétérogènes sont examinés et aucun ordre temporel évident n'existe (contrairement au classique **M.L.**, les caractéristiques de **F.L.M.** peuvent se produire au moment de la prédiction). Plusieurs implémentations **F.L.M.** sont utilisées dans les modèles de **C.L.P.**, chacun d'eux correspondant à une flèche dans les représentations graphiques de **D.B.N.** (voir Figure 3.4) et calcule les probabilités suivantes :

- $P(C) \simeq \prod P(c|c_h)$  : Séquence de concepts
- $P(V|C) \simeq \prod P(v|c)$  : Valeurs conditionnées par les concepts
- $P(W|C) \simeq \prod P(w|w_h, c)$  : Séquence de mots conditionnée par les concepts.
- $P(W|C, V) \simeq \prod P(w|w_h, v, c)$  : Séquence de mots conditionnée par les concepts et les valeurs.

où  $h$  représente un historique qui peut varier en fonction de la longueur du modèle utilisé (-1 pour les 2-grammes, -1,-2 pour les 3-grammes etc.)

Les modèles **D.B.N.** réels utilisés dans le système ?, par exemple, sont illustrés dans la figure 3.4. Le concept et la valeur de décodage des mesures sont découplés et correspondent à des graphes supérieurs et inférieurs dans la figure 3.4. Le processus de décodage conceptuel génère des concepts et des séquences de transitions qui deviennent des variables observées pour le décodage de la valeur. Les probabilités conditionnelles sont des **F.L.M.** 2 ou 3-grammes.

Traditionnellement, après qu'une séquence de concepts a été décodé, les sous-chaînes segmentées de mots sont converties dans une forme normalisée, définies dans le dictionnaire sémantique. Dans la phrase "Donnez-moi le prix car je voudrais une chambre équipée pas plus de cinquante euros", le module de normalisation traduit la séquence "pas plus de" affectée à l'attribut "comparative-paiement-chambre" à la forme normalisée "moins de". Beaucoup de séquences lexicales peuvent correspondre à la même valeur normalisée. Les valeurs d'attributs sont des unités numériques, les noms propres ou des catégories sémantiques regroupant des unités lexicales qui sont des synonymes pour la tâche. L'ensemble des valeurs normalisées associées à chaque attribut est défini dans le dictionnaire sémantique avec 3 configurations différentes possibles : une liste de valeurs (par exemple "comparative" avec les valeurs possibles "autour", "moins-que", "maximum", "a minima" et "plus-que"), les valeurs des expressions régulières (comme pour les dates) ou ouvert (pas de restrictions, comme pour les noms des clients). Cette étape de normalisation est généralement fondée sur des règles déterministes, mais peuvent être introduites dans le modèle global stochastique grâce à un niveau de plus.

## 3.3 Les modèles basés sur un apprentissage discriminant

### 3.3.1 Les arbres de classification sémantique

La plupart des travaux ont utilisé deux grandes approches de l'apprentissage machine pour l'étiquetage de séquences.

La première approche s'appuie sur des *S.F.S.T.* et des modèles probabilistes *k*-génératifs pour des séquences jumelées d'entrées et des séquences d'étiquettes, par exemple les modèles de Markov cachés (*H.M.M.*) [McCallum et al. \(2000\)](#); [Kupiec \(1992\)](#) ou modèles de Markov multi-niveaux [Bikel et al. \(1999\)](#).

La deuxième approche considère le problème de l'étiquetage de séquences comme une suite de problèmes de classification, une pour chacune des étiquettes dans la séquence. Le résultat de classification à chaque position peut dépendre de l'entrée dans son ensemble et sur les *k*-classifications précédentes.

On considère qu'une approche générative, fournit un entraînement de phrases bien comprises et des algorithmes de décodage pour les *H.M.M.* et plus généralement des modèles graphiques. Toutefois, les modèles génératifs efficaces ont besoin d'hypothèses d'indépendances conditionnelles strictes. Par exemple, il n'est pas pratique de mettre une étiquette à une position donnée dépendante d'une fenêtre de la séquence d'entrée aussi bien que pour les étiquettes environnantes, puisque le problème de l'inférence pour le modèle graphique correspondant serait intraitable.

Les éléments non indépendants des entrées, tels que la capitalisation, les suffixes et les mots environnants, sont importants dans le traitement des mots invisibles pour l'entraînement, mais ils sont difficiles à représenter dans les modèles génératifs.

En outre, les caractéristiques représentées par des schémas impliquant des dépendances complexes à distance ne sont pas capturées par d'autres modèles génératifs légèrement sensibles au contexte.

L'interprétation peut être considérée comme une classification des groupes de signes dans les classes, chaque classe étant identifiée par une étiquette concept ou l'instance d'un type de structure sémantique. Avec le développement des méthodes discriminatives pour l'entraînement des classificateurs à partir d'exemples, il apparaît intéressant d'apprendre des modèles de mots utiles pour différencier des phrases exprimant un concept, des phrases n'exprimant pas cette notion.

Le système CHANEL, [Kuhn et De Mori \(1995\)](#) apprend les règles d'interprétation sémantique au moyen d'une forêt d'arbres de décision spécialisés, appelée "Semantic Classification Trees" (*S.C.T.*). Chaque arbre de la forêt est obtenu par apprentissage discriminant à partir d'exemples d'un corpus annoté sémantiquement. L'annotation nécessaire consiste seulement à ajouter un sens à la transcription d'une phrase.

Il y a un *S.C.T.* pour chaque concept élémentaire. Un *S.C.T.* est un arbre binaire avec une question associée à chaque nœud. Les questions sont générées et sélectionnées automatiquement. Chaque nœud a deux successeurs, l'un est atteint si la réponse à la

question de nœud est OUI, l'autre nœud est atteint si la réponse est NON. Les questions portent sur des modèles de phrases faites de mots et de symboles génériques (+). Si le motif du nœud correspond à la phrase qui doit être interprétée, alors la réponse à la question du nœud est OUI et le nœud pointé par le successeur de l'arc option OUI est considéré, sinon la réponse est NON et le nœud correspondant est considéré comme successeur. La figure 3.5 montre un exemple d'un arbre pour le concept "tarif". Un exemple de question est le motif "<+ M(tarif) +>" qui correspond à une phrase contenant n'importe quel mot membre de l'ensemble M(tarif) des mots exprimant la même signification que tarif.

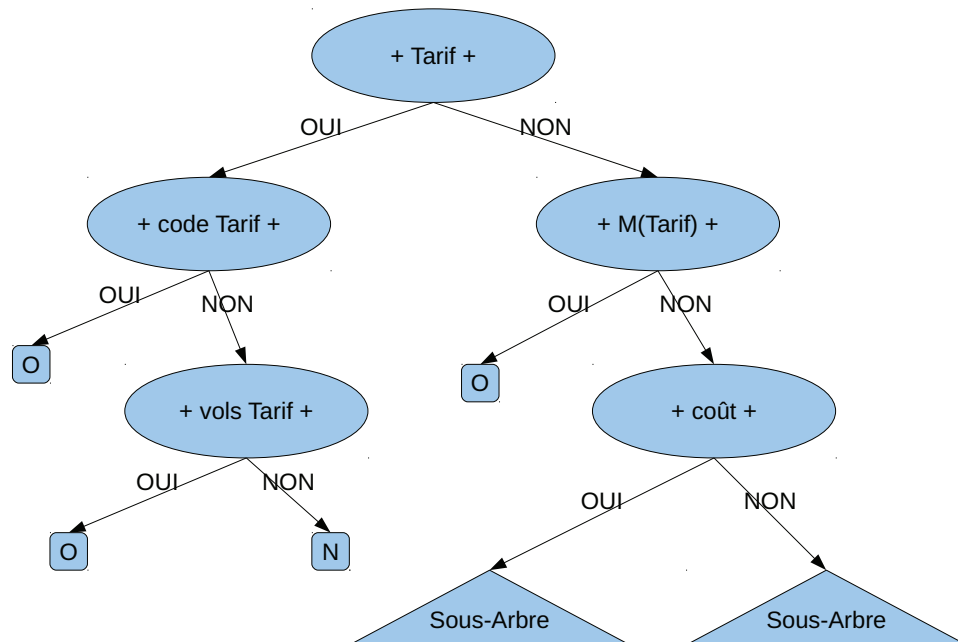


FIGURE 3.5 – Exemple d'un arbre de classification sémantique S.C.T.

La nature des questions posées dans le S.C.T. est telle que les connaissances acquises sont robustes à des erreurs grammaticales et lexicales dans l'entrée de la R.A.P. . En fait, ces questions sont générées d'une manière qui tend à minimiser le nombre de mots qui doit être correct pour la compréhension. La génération des questions comporte des "lacunes" : les mots et groupes de mots qui peuvent être ignorés. Ainsi, chaque feuille d'un S.C.T. correspond à une expression régulière contenant des lacunes, des mots et unités syntaxiques (par exemple, heures, dates, types d'avion). La plupart des S.C.T. de CHANEL décident si un concept donné est présent ou absent de la représentation sémantique de l'énonciation, pour de tels S.C.T. , l'étiquette OUI ou NON dans une feuille, dénote la présence ou l'absence du concept correspondant.



Les **S.C.T.** construisent une partie de la représentation sémantique et examinent l'énoncé entier. Cette situation permet à un mot ou une phrase de contribuer à plus d'un concept, et permet aussi à des mots ou des phrases qui sont éloignés les uns des autres, de participer à la même notion. Les détails sur l'entraînement du **S.C.T.** se trouve dans [Kuhn et De Mori \(1995\)](#).

Dans le système CHANEL, les hypothèses d'interprétation sont marquées par les probabilités suivantes :  $P(C|W)P(W|Y)$ .  $P(C|W)$  est obtenu à partir du nombre de fois que la feuille liée au motif (qui correspondait avec W) est atteinte.

D'autres classificateurs ont ensuite été développés pour l'analyse superficielle. L'analyse sémantique peu profonde, avec l'objectif de créer une signification indépendante du domaine de représentation basée sur une structure prédicat - argument, est décrite dans [Gildea et Jurafsky \(2002\)](#). La majorité des approches utilisent des fonctionnalités d'analyse superficielle pour effectuer la classification. Dans une classification constituant par constituant (C/C) [Hacioglu et Ward \(2003\)](#), la représentation de l'arbre syntaxique d'une phrase est linéarisée en une séquence de constituants non terminaux syntaxiques. Chaque constituant est classé en fonction de plusieurs arguments ou des rôles sémantiques en utilisant les caractéristiques découlant de leur contexte respectif. Dans la méthode au mot par mot, les caractéristiques sont obtenues avec un processus de bas en haut (bottom-up) pour chaque mot après séparation d'une phrase dans les phrases. En supposant que les parties du discours (**P.O.S.**) sont également disponibles, les mots sont étiquetés à l'intérieur d'une phrase (I), en dehors d'une expression (O) ou au début d'une phrase (B). C'est ce qu'on appelle la représentation IOB [Ramshaw et Marcus \(1995\)](#).

Le problème de l'analyse superficielle est la segmentation NP (qui trouve les noyaux non récursifs des syntagmes nominaux appelé NP de base). Le travail de pionnier de [Ramshaw et Marcus \(1995\)](#) introduit la segmentation NP comme un problème d'apprentissage machine, avec des ensembles de données standards et des paramètres d'évaluation. La tâche a été étendue à des types supplémentaires de phrases pour la tâche partagée CoNLL-2000 [Sang et Buchholz \(2000\)](#), qui est maintenant la tâche d'évaluation standard pour l'analyse superficielle. [Ramshaw et Marcus \(1995\)](#) utilisent l'apprentissage basé sur la transformation [Brill \(1995\)](#), qui peut être considérée comme une méthode de classification.

L'approche de classification séquentielle peut gérer de nombreuses fonctionnalités corrélées, comme l'ont démontré les travaux sur l'entropie maximum de [McCallum et al. \(2000\)](#); [Ratnaparkhi \(1996\)](#) et une variété d'autres classifieurs linéaires, y compris SNoW [Punyakanok et Roth \(2000\)](#), AdaBoost [Abney et al. \(1999\)](#), et les machines à vecteurs de supports [Kudo et Matsumoto \(2001\)](#). En outre, les classifieurs sont entraînés afin de minimiser une fonction liée à l'erreur d'étiquetage, conduisant à une plus petite erreur dans la pratique, si les données d'entraînement sont suffisamment disponibles. En revanche, les modèles génératifs sont entraînés afin de maximiser la probabilité conjointe des données d'entraînement, qui n'est pas aussi étroitement liée à la précision des paramètres d'intérêt, si les données réelles n'ont pas été générées par le modèle, comme c'est toujours le cas dans la pratique.

### 3.3.2 Les Machines à Vecteurs de Support

Les Machines à Vecteurs de Support (Support Vector Machines ( *S.V.M.* )) réalisent une approche standard de classification basée sur le balisage de concept. Cette approche est basée sur le classificateur local car le *S.V.M.* traite le problème d'étiquetage comme une séquence de problèmes de classification, une pour chacune des étiquettes dans la séquence. L'algorithme manipule des caractéristiques corrélées et non locales, mais à la différence des modèles génératifs, il ne peut pas prendre de décisions à différentes positions qui vont à l'encontre des autres décisions. Un système basé sur cette approche dans le travail partagé est CoNLL2000, "Chunking" et "BaseNP Chunking task". Il utilise des combinaisons heuristiques des classificateurs *S.V.M.* séquentiels à déplacement avant et à déplacement arrière, prenant comme paramètres les décisions précédemment faites. Puisque les *S.V.M.* sont des classificateurs binaires, le système s'étend à des classifieurs *S.V.M.* K-classe en utilisant la classification par paires. Par conséquent, les  $K(K - 1)/2$  classificateurs sont construits considérant toutes les paires de classes. La décision finale est donnée par leur vote pondéré.

Dans Pradhan et al. (2004) , le problème de l'analyse est formulé comme un problème de classification multi-classe comme initié dans Gildea et Jurafsky (2002) et utilise un classifieur à vecteurs de support ( *S.V.M.* ).

Un analyseur fondé sur les actions d'étiquetage produisant des structures d'arbres peu profondes et non chevauchantes, est décrit dans Hacioglu (2004). Une phrase est considérée comme une séquence de phrases de base à un niveau syntaxique. Au niveau sémantique, les arguments prédicats ont associé des séquences de phrases de base. La détection d'un prédicat est représentée comme un arbre lexicalisé. Les niveaux de l'arbre contiennent des représentations des prédicats-arguments lexicalisés avec les mots clés, les morceaux syntaxiques engendrés par chaque argument, les parties du discours ( *P.O.S.* ) et des mots. La stratégie d'analyse est fondée sur des mesures de marquage et se compose de trois éléments qui sont successivement appliqués au texte d'entrée pour un prédicat choisi afin de déterminer ses arguments. Ces composants sont de l'étiquetage de *P.O.S.* , la phrase de base et des marqueurs sémantiques. Les classifieurs *S.V.M.* sont utilisés à tous les niveaux.

Les résultats dans Pradhan et al. (2007) ont montré une baisse significative des performances lors de l'entraînement et des tests sur des corpus différents. Les hypothèses d'étiquettes de concepts peuvent être générées en utilisant des classifieurs et des combinaisons de classifieurs. Créer des systèmes robustes pour le langage naturel est fortement tributaire de la disponibilité des données pour l'entraînement, la reconnaissance et la compréhension.

Dans Hacioglu et Ward (2003), les Machines à Vecteurs de Supports ( *S.V.M.* ), sont utilisés par exemple pour classer N rôles mots commençants, mots internes, mots cibles et les mots extérieurs après l'apprentissage sur FrameNet .

Le modèle d'application est décidé par une liste de décision, Rivest (1987), c'est à dire, en appliquant toujours la plus forte fonctionnalité. Dans une liste de décisions, tous les éléments sont triés par ordre décroissant de la confiance. Quand un nouveau



concept cible est classé, le classifieur compare les fonctionnalités contre les contextes de la notion cible dans la liste. La première caractéristique correspondante est appliquée pour trouver une valeur d'emplacement et vérifier la cohérence parmi les emplacements / valeurs identifiés. On note que les classifieurs dépendant du thème et correspondant à des concepts différents sont entraînés et exécutés de manière indépendante.

### 3.3.3 Utilisation de la connaissance préalable

Parmi les exigences de dialogues parlés pour des domaines d'application tels que ceux pour les services à la clientèle automatisés et services d'assistance, il convient de mentionner :

- préciser la reconnaissance avec un large vocabulaire pour une large variété de requêtes en entrée,
- identifier et interpréter les segments de parole en exprimant des demandes des utilisateurs,
- soutenir les stratégies mixtes d'initiative.

Dans [Shapire et al. \(2005\)](#), les auteurs explorent l'utilisation de la connaissance humaine, conçue pour compenser le manque de données dans la construction de classificateurs robustes utilisant l'algorithme de AdaBoost [Freund et Schapire \(1999, 1996\)](#). Dans sa forme standard, le "boosting" ne permet pas l'incorporation directe de la connaissance préalable. Une modification du "boosting" est nécessaire. Les connaissances préalables peuvent être acquises auprès de plusieurs sources, par exemple, le jugement humain, la connaissance du monde, les manuels, et les sites internet liés au domaine. En fait, tout en développant un système de dialogues parlés, les concepteurs ont accès à une ou plusieurs de ces sources de connaissances.

Les concepteurs utilisent ces sources de connaissances pour en déduire des informations cruciales pour le développement du système de dialogue, c'est à dire, les fonctionnalités à l'appui, et une compréhension de base de la façon dont les utilisateurs peuvent interagir avec l'application. Il serait donc prudent, par conséquent, d'utiliser également ces sources de connaissances pour l'amorçage du module de catégorisation de textes, nécessaire à la compréhension du langage naturel, en particulier lorsque les données sont limitées. L'idée de base de l'approche consiste à modifier la fonction de perte utilisée dans le "boosting" ainsi que l'algorithme qui évalue deux termes, l'un mesurant l'ajustement aux données d'entraînement, et l'autre mesurant l'ajustement aux données du modèle construit par l'humain. La modification algorithmique actuelle s'avère être très simple, ne nécessitant que l'ajout de pseudo-exemples pondérés pour l'ensemble d'entraînement. À cet égard, la méthode s'avère être similaire à celle proposée par [Pazzani et Billsus \(1997\)](#) pour modifier l'algorithme Bayésien naïf pour intégrer les connaissances préalables.

### Boosting et incorporation de connaissances préalables

Le but d'un algorithme d'apprentissage est d'utiliser les données d'entraînement, pour dériver un ensemble de règles qui prédit précisément la classe des nouvelles instances de  $x$ , une telle prédiction est un classifieur. L'approche de "boosting" est basée sur [Freund et Schapire \(1997\)](#), [Schapire \(2002\)](#). L'idée du boosting est de construire un classifieur de très haute précision, en combinant plusieurs classifieurs basiques, chacun d'eux étant faiblement précis. La collection des classifieurs de bases est construite par tour. A chaque tour un classifieur est généré. Le boosting fournit un ensemble de poids non négatifs pour les exemples d'entraînements. Ces poids permettent d'avoir une évaluation sur l'acuité de la classification à correctement étiqueter chaque exemple. Généralement les exemples qui sont le plus souvent mal classés, par les classifieurs de base, auront un poids plus élevé pour forcer l'apprentissage, sur les exemples les plus durs. Une mesure de confiance est utilisée, elle est basée sur la publication [Schapire et Singer \(1999\)](#).

Une régression logistique est utilisée pour la prédiction de la probabilité de survenance d'un événement par ajustement de données à une courbe logistique de fonction logit. Il s'agit d'un modèle linéaire généralisé utilisé pour la régression binomiale. Comme de nombreuses formes de l'analyse de régression, le modèle fait usage de plusieurs variables explicatives qui peuvent être numériques ou catégorielles.

Soient  $X$  et  $Y$  les espaces respectivement de tous les exemples et de toutes les étiquettes. Si l'on considère un classifieur binaire alors  $Y = -1, +1$ . L'entraînement et le test se font sur des exemples  $x, y$  sélectionnés indépendamment sur la distribution  $D$  de  $X \times Y$ . La confiance  $h$  sort une valeur réelle  $h(x)$ , dont le signe ( $-1$  ou  $+1$ ) est interprété comme une prédiction, et la valeur absolue  $|h(x)|$  est la mesure elle-même. Avec les données d'entraînement, on souhaite construire une règle qui estime la probabilité conditionnelle telle que  $y = +1$  étant donné  $x$  quand le test d'exemple  $(x, y)$  est pris dans la distribution  $D$ . Cette étape est effectuée par régression logistique en créant une fonction réelle  $f : X \rightarrow \mathbb{R}$  en estimant la probabilité par  $\sigma(f(x))$  où

$$\sigma(z) = \frac{1}{1 + \exp -z} \quad (3.9)$$

Pour trouver  $f$ , il faut maximiser la vraisemblance conditionnelle des données ou minimiser le log négatif de la vraisemblance de la manière suivante :

$$\sum_i \ln(1 + \exp(-y_i f(x_i))) \quad (3.10)$$

Une connection entre le "boosting" et la régression logistique a été en premier suggérée par [Friedman et al. \(2000\)](#) et explorée par la suite, par [Duffy et Helmbold \(1999\)](#).

Une modification du "boosting" a été introduite pour prendre en compte les connaissances au préalable. Un expert humain doit commencer par construire une règle  $\pi$  faisant correspondre chaque exemple  $x$  à une distribution de probabilité conditionnelle estimée  $\pi(y|x)$  sur les possibles valeurs d'étiquettes  $-1, +1$ .

En intégrant les connaissances au préalable, la prédiction a ainsi deux objectifs qui rentrent en conflit : s'adapter aux données, ou s'adapter aux connaissances du modèle humain. Pour mesurer l'adaptation au modèle a-priori, pour chaque exemple  $x_i$ , l'entropie relative est employée (appelée aussi divergence de Kullback-Leibler) entre la distribution du modèle a-priori  $\pi(\cdot|x_i)$  et la distribution sur les étiquettes associées avec le modèle logistique construit  $\sigma(f(x))$ . Plus précisément avec  $\pi_+(x) = \pi(y = +1|x)$ , l'adaptation au modèle a-priori est mesuré par :

$$\sum_i RE(\pi_+(x)||\sigma(f(x))) \quad (3.11)$$

avec  $RE(p||q)$  est une entropie relative binaire :

$$RE(p||q) = p \ln\left(\frac{p}{q}\right) + (1-p) \ln\left(\frac{1-p}{1-q}\right) \quad (3.12)$$

L'importance relative des deux termes est contrôlée par le paramètre  $\eta$  La fonction objective suivante est obtenue :

$$\sum_i [\ln(1 + \exp(-y_i f(x_i))) + \eta RE(\pi_+(x)||\sigma(f(x)))] \quad (3.13)$$

Ces premières descriptions d'approches du "boosting", supposent une prédiction binaire, c'est à dire deux classes sont mises en hypothèses  $Y = -1, +1$ . Les méthodes introduites par [Schapire et Singer \(1999, 2000\)](#), permettent la prédiction de plus de deux classes, et dans lesquelles un exemple peut appartenir à plusieurs classes. L'idée intuitive est de réduire à des questions binaires. Ainsi on demande au système si chaque exemple est ou n'est pas dans chacune des classes.

En particulier, on suppose qu'on a  $k$  classes  $Y = 1, 2, \dots, k$ . Chaque étiquette  $y_i$  est un vecteur  $\{-1, +1\}^k$ , où le  $\ell^{ième}$  composant indique, si l'exemple appartient ou non à la classe  $\ell$ . Le but est de trouver la fonction  $f : X \times Y \rightarrow \mathfrak{R}$  et  $\sigma(f(x, \ell))$  est la probabilité estimée que l'exemple  $x$  soit dans la classe  $\ell$ . La fonction objective 3.10 devient :

$$\sum_i \sum_{\ell} \ln(1 + \exp(-y_{i,\ell} f(x_i, \ell))) \quad (3.14)$$

Le framework permet une connaissance préalable de n'importe quelle nature, tant qu'elle fournit des estimations de la probabilité d'un exemple appartenant à une classe. Une technique possible pour la création d'un tel modèle est maintenant décrit. Pour chaque ensemble de données, un des auteurs, avec un accès à la liste des catégories (mais pas les données elles-mêmes), peut penser à quelques mots-clés pour chaque catégorie. Ces mots-clés sont produits par un processus tout à fait subjectif de libre association avec la connaissance générale de ce que les catégories représentent (et aussi la période pendant laquelle les données ont été connectées), mais sans autre information ou accès à des données. Bien que cette étape nécessite l'implication directe de l'homme, le reste du processus de génération d'un modèle a priori est entièrement automatique. Ces mots-clés sont utilisés pour construire un modèle très simple et naïf.

Une autre possibilité est d'utiliser les connaissances humaines pour extraire de nouvelles caractéristiques pour les classifieurs [Camelin et al. \(2006, 2010\)](#).

Toutefois, puisque les classifieurs sont entraînés pour prendre la meilleure décision locale, contrairement aux modèles génératifs ils ne peuvent pas compromettre les décisions à différentes positions les uns contre les autres. En d'autres termes, les classifieurs séquentiels sont myopes sur l'impact de leur décision actuelle sur les décisions ultérieures [Bottou \(1991a\)](#); [Lafferty et al. \(2001\)](#). Ceci a obligé les meilleurs systèmes de classifieurs séquentiels à recourir à des combinaisons heuristiques de classifieurs séquentiels en avant et en arrière [Kudo et Matsumoto \(2001\)](#).

Une revue de l'analyse superficielle pour la segmentation peut être trouvée dans [Molina et Pla \(2002\)](#)

Des nouveaux modèles de combinaisons stochastiques log-linéaires ont été proposés pour obtenir et évaluer l'interprétation d'un énoncé entier.

### 3.4 Modélisation log-linéaire

Deux types de modèles log-linéaires ont été proposées. Elles diffèrent uniquement par le terme de normalisation. Le premier est normalisé au niveau de la phrase (Champs aléatoires de conditions (Conditional Random Fields , C.R.F. )) [Lafferty et al. \(2001\)](#), le deuxième au niveau positionnel (abrégé sous la forme log-pos). La représentation générale de ces modèles est décrite par l'équation 3.15 comme la probabilité conditionnelle d'une séquence de concepts  $c_1^N = c_1, \dots, c_N$  étant donné une séquence de mots  $w_1^N = w_1, \dots, w_N$  :

$$p(c_1^N | w_1^N) = \frac{1}{Z} \prod_{n=1}^N \exp \left( \sum_{m=1}^M \lambda_m h_m(c_{n-1}, c_n, w_{n-\delta}^{n+\delta}) \right) \quad (3.15)$$

Les modèles log-linéaires sont basés sur des caractéristiques de fonctions  $h_m(c_{n-1}, c_n, w_{n-\delta}^{n+\delta})$  où  $w_{n-\delta}^{n+\delta}$  est le sous-ensemble  $w_{n-\delta}, \dots, w_{n+\delta}$  de  $w_1^N$  avec un paramètre prédéfini  $\delta \in N_0$ , représentant l'information extraite de la phrase donnée, les paramètres correspondants  $\lambda_m$  qui sont estimés par un procédure d'entraînement et une normalisation de terme Z.

#### 3.4.1 Fonctions caractéristiques

Dans les expérimentations, on utilise des fonctions caractéristiques binaires  $h_m(c_{n-1}, c_n, w_{n-\delta}^{n+\delta})$ , c'est-à-dire elles retournent comme valeur soit "0", soit "1". La valeur "1" est retournée ou "levée" si une combinaison prédéfinie de valeur  $c_{n-1}, c_n, w_{n-\delta}, \dots, w_{n+\delta}$  est trouvée dans la date par exemple, autrement "0" est renvoyée. Par exemple : une fonction caractéristique peut être employée si et seulement si le mot prédécesseur  $w_{n-1}$  est "le" et le concept  $c_n$  est "nom". Un autre exemple où une fonction caractéristique

pourrait être levée si et seulement si le concept prédécesseur  $c_{n-1}$  est "nombre" et le concept  $c_n$  est "monnaie". On appelle ces fonctions caractéristiques des "caractéristiques lexicales" pour les cas basés sur des mots prédécesseurs et successeurs. On appelle également ces fonctions caractéristiques, des "caractéristiques de bigram de concepts", les cas basés sur les concepts prédécesseurs et successeurs.

Par souci de clarté, on abrège le numérateur de l'équation 3.15 par

$$H(c_{n-1}, c_n, w_{n-\delta}^{n+\delta}) = \exp \left( \sum_{m=1}^M \lambda_m h_m(c_{n-1}, c_n, w_{n-\delta}^{n+\delta}) \right) \quad (3.16)$$

ce qui équivaut à

$$p(c_1^N | w_1^N) = \frac{1}{Z} \prod_{n=1}^N H(c_{n-1}, c_n, w_{n-\delta}^{n+\delta}) \quad (3.17)$$

Avant que les paramètres  $\lambda_m$  des modèles soient estimés, une liste, contenant toutes les fonctions caractéristiques qui ont pu être rencontrées dans le corpus d'entraînement au moins une fois, est générée.

### 3.4.2 Les "Conditional Random Fields ( C.R.F. )"

Les chaînes linéaires de C.R.F. comme définies dans Lafferty et al. (2001) peuvent être représentées avec l'équation 3.17 et une normalisation  $Z$  au niveau de la phrase

$$Z = \sum_{\tilde{c}_1^N} \prod_{n=1}^N H(\tilde{c}_{n-1}, \tilde{c}_n, w_{n-\delta}^{n+\delta}) \quad (3.18)$$

avec comme probabilité :

$$p(c_1^N | w_1^N) = \frac{\prod_{n=1}^N H(c_{n-1}, c_n, w_{n-\delta}^{n+\delta})}{\sum_{\tilde{c}_1^N} \prod_{n=1}^N H(\tilde{c}_{n-1}, \tilde{c}_n, w_{n-\delta}^{n+\delta})} \quad (3.19)$$

Les C.R.F. peuvent être utilisés de la même manière que des automates à états finis, où un chemin correspond à une possible séquence de concept  $c_1^N$ . Cette approche permet de facilement implémenter les conditions sur une séquence de concepts.

Comme décrit plus haut, la séquence  $c_1^N$  est une séquence d'étiquettes construite. Dans le projet LUNA , l'information est transmise en utilisant un treillis de concepts. Pour évaluer la complexité d'un modèle de langage approprié pour ce type d'applications, l'influence de la longueur de l'historique ou de la recherche sur des éléments futurs (représentée par  $\delta$  dans l'équation 3.15) sur le taux d'erreurs de concepts a été étudiée dans le projet LUNA .

### 3.4.3 Normalisation du niveau de position

Une autre normalisation possible de l'équation 3.17 est en rapport avec le niveau de position.

$$Z = \prod_{n=1}^N \sum_{\tilde{c}} H(c_{n-1}, \tilde{c}_n, w_{n-\delta}^{n+\delta}) \quad (3.20)$$

avec comme probabilité :

$$p(c_1^N | w_1^N) = \prod_{n=1}^N \frac{H(c_{n-1}, c_n, w_{n-\delta}^{n+\delta})}{\prod_{n=1}^N \sum_{\tilde{c}} H(c_{n-1}, \tilde{c}_n, w_{n-\delta}^{n+\delta})} \quad (3.21)$$

Cette modélisation fait généralement référence à l'approche d'Entropie Maximum, on peut citer par exemple [Bender et al. \(2003\)](#). C'est la première approche étudiée dans le projet LUNA, qui a été remplacée par la suite par les [C.R.F.](#) Elle est encore d'un certain intérêt puisque la normalisation est beaucoup moins coûteuse en temps de calculs.

### 3.4.4 L'entraînement

En utilisant l'équation 3.17 avec les normalisations 3.18 ou 3.20 et un ensemble de données d'entraînement  $\left\{ \{c_1^N\}_q | \{w_1^N\}_q \right\}_{q=1}^Q$ , avec Q paires de mots/concepts, le critère pour l'entraînement est donné par le critère de probabilité postérieure de classe :

$$\hat{\lambda}_1^M = \arg_{\lambda_1^M} \max \left\{ \sum_{q=1}^Q \log p_{\lambda_1^M} \left( \{c_1^N\}_q | \{w_1^N\}_q \right) \right\} \quad (3.22)$$

Pour éviter le surapprentissage sur les données d'entraînement, des Gaussiennes sont utilisées comme méthode de lissage ainsi décrite dans [Chen et Rosenfeld \(1999\)](#)

$$\hat{\lambda}_1^M = \arg_{\lambda_1^M} \max \left\{ \log p(\lambda_1^M) + \sum_{q=1}^Q \log p_{\lambda_1^M} \left( \{c_1^N\}_q | \{w_1^N\}_q \right) \right\} \quad (3.23)$$

avec

$$p(\lambda_1^M) = \prod_{m=1}^M \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\lambda_m^2}{2\sigma^2} \right) \quad (3.24)$$

Le paramètre  $\lambda_1^M$  est calculé par un processus itératif appliquant l'algorithme de propagation dit "resilient back" [Riedmiller \(1994\)](#) dans le cas de [C.R.F.](#) et l'algorithme de mise à l'échelle itératif généralisé [Darroch et Ratcliff \(1972\)](#) dans le cas de la normalisation sur le niveau de position.

### 3.4.5 Processus de décision

Pour une séquence de mots  $w_1^N$ , la meilleure séquence d'étiquettes de concepts est produite en utilisant les modèles définis précédemment. Il est nécessaire de résoudre le problème de maximisation suivant :

$$\hat{c}_1^N(w_1^N) = \arg \max_{\lambda_1^M} \{p(c_1^N | w_1^N)\} \quad (3.25)$$

Les **C.R.F.** dans cette approche sont implémentés avec des automates, où le processus de décision est équivalent à sélectionner le premier meilleur concept, alors que la décision prise dans le modèle de normalisation par niveau de position est implémentée par programmation dynamique.

## 3.5 Utilisation des Conditional Random Fields , pour l'analyse superficielle

Les champs conditionnels aléatoires (Conditional Random Fields (**C.R.F.** )) réunissent le meilleur des modèles de génération et de classification. Comme les modèles de classification, ils peuvent accueillir de nombreuses fonctionnalités statistiquement corrélées aux entrées, et ils sont entraînés de manière discriminative. Mais comme les modèles génératifs, ils peuvent trouver un compromis dans les décisions à différentes positions des séquences pour obtenir un étiquetage globalement optimal. [Lafferty et al. \(2001\)](#) a montré que les **C.R.F.** dépassent les modèles de classification tels que les **H.M.M.** sur des données synthétiques et sur une tâche d'étiquetage des parties du discours (Part-of-Speech).

Dans [Sha et Pereira \(2003\)](#), les auteurs montrent que les **C.R.F.** dépassent tous les résultats reportés, sur la segmentation des modèles NP sur les ensembles de données de tests standards, et sont statistiquement indifférenciables des meilleurs analyseurs, le système [Kudo et Matsumoto \(2001\)](#) basé sur les **C.R.F.** propose un vote de 24 classificateurs à vecteurs de support regardant en avant et en arrière. Pour obtenir ces résultats, il faut abandonner l'algorithme original itératif d'apprentissage des **C.R.F.** pour les algorithmes d'optimisation convexe avec de meilleures propriétés de convergence.

Le perceptron généralisé proposé par [Collins \(2002\)](#) est étroitement liée aux **C.R.F.** , mais les meilleures méthodes d'entraînements des **C.R.F.** semblent avoir un léger avantage sur le perceptron généralisé.

Un **C.R.F.** sur  $(X,Y)$  est spécifié par un vecteur  $f$  de fonctionnalités locales et leur poids correspondant  $\lambda$  Chaque caractéristique locale est soit une caractéristique d'état  $s(y, x, i)$ , soit une caractéristique de transition  $t(y, y', x, i)$ , où  $y, y'$  sont des étiquettes,  $x$ , est une séquence d'entrée et  $i$  une position d'entrée. Pour rendre la notation plus



uniforme, on écrit également, pour toute caractéristique d'état  $s$ , et de transition  $t$  :

$$\begin{aligned} s(y, y', x, i) &= s(y', x, i) \\ s(y, x, i) &= s(y_i, x, i) \\ t(y, x, i) &= \begin{cases} t(y_{i-1}, y_i, x, i) & i > 1 \\ 0 & i = 1 \end{cases} \end{aligned} \quad (3.26)$$

Les caractéristiques dépendent des entrées environnantes à la position donnée, bien qu'elles puissent dépendre aussi des propriétés globales des entrées, ou être non nulles seulement à certaines positions. Par exemple, les caractéristique qui distinguent les premières ou les dernières étiquettes.

Le vecteur de caractéristiques globales des **C.R.F.** pour la séquence d'entrée  $x$  et la séquence d'étiquette  $y$  est donnée par :  $F(y, x) = \sum_i f(y, x, i)$ , où  $i$  s'étend sur les positions d'entrées. La distribution de probabilité conditionnelle définie par les **C.R.F.** est alors :

$$P_\lambda(Y|X) = \frac{\exp \lambda.F(Y, X)}{Z_\lambda(X)} \quad (3.27)$$

avec

$$Z_\lambda(X) = \sum_y \exp \lambda.F(y, x) \quad (3.28)$$

Toute distribution conditionnelle positive  $P(Y|X)$  qui obéit aux propriétés de Markov  $P(Y_i | \{Y_j\}_{j \neq i}, X) = P(Y_i | Y_{i-1}, Y_{i+1}, X)$  peut être écrite sous la forme décrite dans l'équation 3.27, pour un choix approprié de fonctions caractéristiques et de vecteur de poids [Hammersley et Clifford \(1971\)](#). La séquence d'étiquettes la plus probable pour la séquence d'entrée  $x$  est :

$$\hat{y} = \operatorname{argmax}_y P_\lambda(y|x) = \operatorname{argmax}_y \lambda.F(y, x) \quad (3.29)$$

parce que  $Z_\lambda(x)$  ne dépend pas de  $y$ .  $F(y, x)$  se décompose en une somme de termes pour les paires consécutives d'étiquettes.

On entraîne un **C.R.F.** en maximisant la log-vraisemblance de l'ensemble d'entraînement  $\{(x_k, y_k)\}_{k=1}^N$ , qui est supposé fixé pour le reste de la section :

$$\begin{aligned} \ell &= \sum_k \log P_\lambda(y_k | x_k) \\ &= \sum_k [\lambda.F(y_k, x_k) - \log Z_\lambda(x_k)] \end{aligned} \quad (3.30)$$

Pour effectuer cette optimisation, il faut chercher le zéro du gradient :

$$\nabla \ell = \sum_k \left[ \lambda.F(y_k, x_k) - E_{P_\lambda(Y|x_k)} F(Y, x_k) \right] \quad (3.31)$$

Le maximum de la vraisemblance des données d'entraînement est atteint quand la moyenne empirique du vecteur de caractéristique globale équivaut à l'espérance du



modèle. L'espérance  $E_{P_\lambda(Y|x_k)} F(Y, x_k)$  peut être calculée efficacement en une variante de l'algorithme d'avant-arrière. Pour un  $x$  donné, on définit la matrice de transition pour la position  $i$  comme suit :

$$M_i[y, y'] = \exp \lambda \cdot f(y, y', x, i) \quad (3.32)$$

Soit  $f$  n'importe quelle caractéristique locale,  $f_i[y, y'] = f(y, y', x, i)$ ,  $F(y, x) = \sum_i f(y_{i-1}, y_i, x, i)$ , et soit  $*$  désigne le produit de matrice, alors :

$$\begin{aligned} E_{P_\lambda(Y|x_k)} F(Y, x_k) &= \sum_y P_\lambda(y|x) F(y, x) \\ &= \sum_i \frac{\alpha_{i-1} (f_i * M_i \beta_i^\top)}{Z_\lambda(x)} \\ Z_\lambda(x) &= \alpha_n \cdot \mathbf{1}^\top \end{aligned} \quad (3.33)$$

où  $\alpha_i$  et  $\beta_i$  les vecteurs de coût d'état avant-arrière définis par :

$$\begin{aligned} \alpha_i &= \begin{cases} \alpha_{i-1} M_i & 0 < i \leq n \\ 1 & i = 0 \end{cases} \\ \beta_i^\top &= \begin{cases} M_{i+1} \beta_{i+1}^\top & 1 < i \leq n \\ 1 & i = n \end{cases} \end{aligned} \quad (3.34)$$

Ainsi on peut utiliser une passe en avant pour calculer  $\alpha_i$  et une passe en arrière pour calculer  $\beta_i$  et accumuler les espérances des caractéristiques. Pour éviter le surapprentissage, les auteurs pénalisent la vraisemblance par un poids gaussien sphérique a priori [Chen et Rosenfeld \(1999\)](#).

### 3.5.1 Segmentation avec les Conditional Random Fields

Dans [Sha et Pereira \(2003\)](#), les segmentations en **C.R.F.** ont une dépendance markovienne de second ordre, entre les étiquettes de segments. Ceci peut facilement être encodé en créant des paires d'étiquettes de **C.R.F.** des étiquettes consécutives de segments. L'étiquette à la position  $i$  est  $y_i = c_{i-1}c_i$  où  $c_i$  est l'étiquette segment du mot  $i$  parmi "O" (pour Sortie (Output)), "B" (pour Commencement (Begin)) et "I" (pour Intermédiaire (Inter)). B est utilisé pour commencer un segment, les étiquettes O et I sont impossibles. les étiquettes successives sont contraintes de la manière suivante :  $y_{i-1} = c_{i-2}c_{i-1}$ ,  $y_i = c_{i-1}c_i$  et  $c_0 = O$ . Ces contraintes topologiques du modèle sont renforcées en donnant aux caractéristiques appropriées un poids de  $-\infty$ , forçant toutes les étiquettes interdites d'avoir une probabilité à zéro.

La décision est contrainte par la puissance de l'ordinateur. Le choix de caractéristique a été pour les auteurs, gouverné par la puissance de l'ordinateur, puisqu'il n'y a pas de sélection des caractéristiques et elles sont toutes utilisées à l'entraînement et la phase de test. Les auteurs utilisent la représentation factorisée suivante, pour les

caractéristiques :  $f(y_{i-1}, y_i, x, i) = p(x, i) \cdot q(y_{i-1}, y_i)$ , où  $p(x, i)$  est un prédicat sur les séquences  $x$  d'entrées, la position courante  $i$ ,  $q(y_{i-1}, y_i)$  est un prédicat sur les paires d'étiquettes. La factorisation de  $f(y_{i-1}, y_i, x, i)$  est possible car l'ensemble d'étiquettes est fini. et permet à chaque prédicat d'entrée d'être évalué une seule fois pour toutes les caractéristiques qui l'utilisent. Cela permet de travailler avec des millions de caractéristiques sur des grands ensembles d'entraînement. Par exemple,  $p(x, i)$  peut être un prédicat : "le mot à la position  $i$  est "the" ", ou "les étiquettes à la position  $i - 1, i$  sont DT,NN".

Les paramètres d'évaluation standard pour un segmenteur sont la précision  $P$  (fraction de la production des segments qui correspondent exactement aux segments de référence), le rappel  $R$  (fraction des segments de référence retournée par le segmenteur), et leur moyenne harmonique, le F1-score  $F1 = \frac{2 * P * R}{P + R}$  (que nous appelons F-score par la suite). Les relations entre le F-score et l'erreur d'étiquetage ou de log-vraisemblance ne sont pas directes, de sorte qu'on note à la fois le F-score et les autres paramètres pour les modèles testés. Pour faire des comparaisons avec d'autres résultats, on utilise le F-score.

## 3.6 Conclusions

Plusieurs exemples de modèles décrits dans ce chapitre ont été implémentés et évalués dans le projet LUNA . Les résultats ont montré qu'aucun modèle n'est parfait et qu'il est préférable d'utiliser les modèles pour calculer des probabilités d'hypothèses à utiliser pour prendre des décisions après avoir effectué une composition sémantique.

Nous allons voir dans la suite, quelques systèmes avancés.



# Chapitre 4

## Systemes avances

### Sommaire

---

<b>4.1</b>	<b>Methodes recentes</b>	<b>84</b>
4.1.1	Approche de traduction automatique	84
4.1.2	Quelques applications recentes	87
<b>4.2</b>	<b>Architectures avances</b>	<b>90</b>
4.2.1	Le systeme de Microsoft	90
4.2.2	Le systeme d'AT & T	94
4.2.3	Le systeme d'IBM	97
4.2.4	Le projet Europeen LUNA	100

---

## 4.1 Méthodes récentes

### 4.1.1 Approche de traduction automatique

Les descriptions sémantiques peuvent être considérées comme des phrases dans un Langage de Représentation Sémantique ( **L.R.S.** ) généré par une grammaire. Il est possible de voir l'interprétation sémantique comme une traduction de phrases en langage naturel dans les phrases **L.R.S.** . Un schéma de traduction dirigé par la syntaxe ( **S.T.D.S.** ) est un penta-tuple  $T : [VN, VT_1, VT_2, R, S]$ , où  $S$  est le symbole de départ,  $VN$  est l'ensemble des symboles non-terminaux,  $VT_1$  est le jeu de mots d'entrée,  $VT_2$  est l'ensemble des primitives sémantiques,  $R$  est l'ensemble des règles, pour la réécriture des symboles non-terminaux, du type  $A \rightarrow \alpha\beta$ , où  $\alpha \in (VN \cup VT_1)$ ,  $\beta \in (VN \cup VT_2)$  et  $A \in VN$ .

Avec ces règles, les phrases et les descriptions sémantiques correspondantes peuvent être générées. Il est également possible de générer des descriptions sémantiques lors de l'analyse des phrases d'entrée. Les traductions peuvent être marquées par des probabilités dans un **S.T.D.S.** stochastique.

Dans ce cas, la syntaxe et la sémantique des règles de production sont associées à des probabilités avec lesquelles il est possible de calculer la probabilité. C'est la probabilité que les règles sémantiques génèrent l'arborescence abstraite d'une interprétation étant donné l'arbre abstrait de l'analyse syntaxique d'une phrase.

Les règles pour les **S.T.D.S.** peuvent être construites manuellement, ou elles peuvent être apprises avec leurs probabilités par des techniques d'inférence grammaticale à partir d'un corpus. Comme les règles ne permettent pas de générer toutes les phrases possibles observables, la couverture de ces grammaires n'est que partielle. L'interprétation des phrases qui ne peuvent être générées par les règles est rendue possible par l'exécution de correction d'erreurs d'analyse. Une bonne introduction du formalisme et son application à la reconnaissance des motifs peut être trouvée dans [Fu \(1982\)](#); une application à la compréhension de la parole d'un **S.T.D.S.** stochastique avec des capacités d'apprentissage est décrite dans [Vidal et al. \(1993\)](#). Un transducteur subsequentiel d'apprentissage est décrit dans [Castellanos et al. \(1998\)](#).

Basé sur des recherches approfondies sur la traduction automatique statistique [Brown et al. \(1993\)](#), un système pour la compréhension dépendant du contexte **A.T.I.S.** a été développé chez IBM [Koppelman et al. \(1995\)](#); [Epstein et al. \(1996\)](#) basé sur les concepts suivants. Soit  $E$  la suite des mots de langue anglaise dans un énoncé **A.T.I.S.**, et soit  $F$  le contenu sémantique de l'énonciation représentée dans un langage formel approprié. Les auteurs s'intéressent à la distribution conjointe  $Pr(F, E) = Pr(F)Pr(E|F)$ . En particulier, pour un  $E$  donné, il faut trouver sa traduction la plus probable :

$$\arg \max_F (Pr(F|E)) = \arg \max_F (Pr(F, E)) = \arg \max_F (Pr(F)Pr(E|F)) \quad (4.1)$$

Ainsi, on a besoin d'un modèle de langage  $Pr(F)$  pour le contenu sémantique des requêtes **A.T.I.S.**, et un modèle de traduction  $Pr(E|F)$ . L'idée centrale du modèle de

traduction est que l'énoncé en anglais peut être regroupé dans des phrases et chaque regroupement est généré par un mot de **L.R.S.** . L'alignement entre les regroupements de  $E$  et les unités en  $F$  est caché, ainsi que les limites de chaque regroupement.

Soit  $E = e_1, \dots, e_i, \dots, e_{LE}$  un énoncé en anglais fait de  $LE$  mots, et  $F = f_1, \dots, f_j, \dots, f_{LF}$  la séquence correspondante de  $LF$  symboles descriptifs de regroupement. Il peut y avoir plusieurs regroupements pour le même symbole. Un alignement  $A$  est un tuple LC décrivant, pour chaque élément  $f_j$ , le mot  $e_i$  dans un ordre continu, lui correspondant. Soit  $a_i$  un élément de  $A$ , contenant  $e_i$  et  $f(a_i)$  l'élément correspondant dans  $F$ . on peut écrire :

$$P(E|F) = \sum_A P(E, A|F) = P(LE|LF) \prod_{i=1}^{LE} P(e_i|f(a_i))P(\cdot_i|F) \quad (4.2)$$

Ces probabilités sont fournies par un modèle de "fertilité".

- Modèle 1 : Le modèle 1 suppose  $P(a_i, F) = \frac{1}{LF}$  signifiant que chaque mot anglais peut être aligné à n'importe quelle position  $c$  dans un langage formel équiprobable. Il suppose aussi que  $e_i$  est généré indépendamment et est seulement conditionné sur le  $f$  duquel il est aligné. Ainsi aucun regroupement n'est supposé. Les formules sont fournies dans [Brown et al. \(1993\)](#)
- Modèle de regroupement A : Le modèle A suppose qu'un élément de  $A$  correspond à un regroupement  $c$ . Ainsi il y a  $LF$  éléments dans  $A$ . De plus plusieurs regroupements peuvent correspondre à un même  $f$ , mais tous les mots d'un regroupement doivent correspondre au même  $f$ .

Ainsi on peut écrire les formules suivantes :

$$\begin{aligned} P(E|nc, F) &= \sum_{nc=1}^{LE} P(nc|F)P(E|nc, F) \\ P(E|nc, F) &= \sum_{C:|C|=nc} P(E, C|F) \\ P(E, C|F) &= \sum_A P(E, C, A|F) \\ P(E, C, A|F) &= \frac{1}{LF^{|C|}} \prod_{k=1}^{|C|} P(c_k|f(a_k)) \\ P(C_k|f(a_k)) &= \prod_{i=b_k}^{b_k+|c_k|} P(e_i|c_k) \end{aligned} \quad (4.3)$$

Avec ces hypothèses, il est possible d'échanger la somme et le produit conduisant à :

$$\begin{aligned} P(E, C|F) &= \frac{1}{LF^{|C|}} \prod_{k=1}^{|C|} q(c_k) \\ q(c_k) &= \sum_j P(c_k|f_j) \end{aligned} \quad (4.4)$$

Dans un travail plus récent [Della Pietra et al. \(1997\)](#), un modèle général de fertilité est

examiné dans lequel le calcul suivant est effectué :

$$\begin{aligned}
 P(E, C, A|F) &= \frac{1}{LF!} \prod_{j=1}^{LF} P(n_j|f_j) n_j! \prod_{k=1}^{|C|} P(c_k|f(a_k)) \\
 P(c_k|f(a_k)) &= P(|c_k||f(a_k)) \prod_{i=b_k}^{b_k+|c_k|} P(e_i|c_k)
 \end{aligned} \tag{4.5}$$

où  $L$  est le nombre total de regroupements,  $n$  est le nombre de regroupements par symbole  $f$  et  $|L|$  représente le nombre total des arrangements de  $L$  regroupements. Si l'on suppose que le nombre  $n$  de regroupements alignés à un symbole  $f$  suit une distribution de Poisson, alors :

$$P(n|f) = \frac{e^{-\lambda_f} \lambda_f^n}{n!} \tag{4.6}$$

De petites améliorations sur la tâche [A.T.I.S.](#) ont été obtenues avec le modèle général et un modèle de langage de mot d'en-têtes basées sur la probabilité suivante :

$$P(c|f) = \frac{P(|c||f)}{|c|} \sum_{i=1}^{|c|} P_{head}(e_i|f_c) \prod_j P_{nonhead}(e_j|f_c) \tag{4.7}$$

$P(|c||f)$  et  $P(e|f)$  sont calculés avec suppression des interpolations et lissés avec les uni-grammes non conditionnels correspondants.

Toutes ces approches ne prennent pas en compte le fait que certains mots n'ont aucune importance pour l'interprétation et les phrases  $F$  peuvent avoir des lacunes de regroupement peu important pour la compréhension. Au contraire, les probabilités d'association entre les regroupements lointains peuvent être importantes.

Dans [Papineni et al. \(1998\)](#) des modèles de traduction statistiques sont utilisés pour traduire une phrase source  $S$  dans une langue cible  $T$  artificielle en maximisant la probabilité suivante :

$$P(T|S) = \frac{P(S|T)P(T)}{P(S)} \tag{4.8}$$

La tâche centrale dans l'entraînement est de déterminer les corrélations entre le groupe de mots dans une langue et des groupes de mots dans l'autre. L'analyse du signal a échoué à capturer ces corrélations, un modèle direct a été conçu pour calculer directement la probabilité a posteriori  $P(T|S)$ .

Les caractéristiques binaires importantes sont utilisées pour caractériser les co-occurrences d'une séquence  $s$  et d'une séquence  $t$  dans les deux langues. Les caractéristiques vérifient la présence ou l'absence de  $n$ -grammes, bi-grammes sur de longues distances, des ensembles de mots à la fois dans les langues source et cible.

La probabilité a postérieure est calculée comme suit :

$$P(T|S) = \frac{P_0(T|S) \prod \alpha_i^{\Phi_i(S,T)}}{P(S)} \tag{4.9}$$

où  $\alpha_i = e^{\lambda_i}$  est un vote multiplicatif assigné à la caractéristique  $\Phi_i(S, T)$ . C'est la solution optimale pour n'importe quelle distribution de probabilité qui satisfait cette contrainte et est proche de celle a priori.

La qualité d'un modèle peut être mesurée compte tenu des paires d'événements (l'historique, l'avenir),  $(h, f)$ . Une mesure de la qualité d'un modèle  $P$  est la discrimination mesurée comme :

$$D(P) = \sum_{i=1}^T \log \left( \frac{P(f_i|h_i)}{P(\tilde{f}_P(h_i)|h_i)} \right) \quad (4.10)$$

où  $\tilde{f}_P(h_i)$  est le futur prédit par le modèle étant donné l'historique.  $\tilde{f}_P(h_i)$  est le meilleur modèle d'estimation basé sur  $\arg \max_c P(f|h)$ . Une autre mesure de qualité est la vraisemblance des données d'entraînement impliquant les distributions empiriques :

$$\sum_{h,f} \tilde{P}(h, f) \log P(f|h) \quad (4.11)$$

Etant donnée un ensemble de caractéristiques, la discrimination est calculée avec l'ensemble  $\alpha_i = e^{\lambda_i}$  qui minimise la discrimination  $D$ . En démarrant à partir d'un petit ensemble de caractéristiques, de nouvelles sont ajoutées, une à la fois, et ordonnées suivant leur capacité à réduire  $D$ .

D'intéressantes solutions pour l'interprétation sémantique utilisant l'approche de machine de traduction peuvent être trouvés dans [Macherey et al. \(2001\)](#). [Sudoh et Tsukada \(2005\)](#) propose un modèle de compréhension de la langue naturelle qui peut être entraîné en utilisant le relâchement de la correspondance entre les paires de séquence de mots et un ensemble de concepts associés au niveau de la phrase. Les concepts sont représentés comme des paires attribut / valeur.

En outre, dans les systèmes de dialogue parlé, il est nécessaire de rejeter les hypothèses de concepts erronées aussi bien que de trouver les plus susceptibles. Pour le rejet, une mesure de confiance des concepts est nécessaire. Une méthode est proposée en utilisant un modèle de traduction statistique des mots vers les concepts. Les modèles de traduction peuvent être entraînés en utilisant le corpus avec une annotation en concept simplifiée, où chaque phrase est alignée à un ensemble de concepts, mais chaque concept n'est pas aligné explicitement aux mots correspondants dans la phrase. Ici, les alignements entre les mots et les concepts peuvent être automatiquement obtenus à partir des co-occurrences. Le modèle est un modèle de probabilité basé sur N-grammes joints qui peuvent être facilement intégrés aux moteurs de reconnaissance de la parole basés sur les N-grammes. La confiance des hypothèses de compréhension de la parole peut être obtenue comme une probabilité a posteriori à l'aide des contextes de niveau concept sous la forme d'un N-grammes.

#### 4.1.2 Quelques applications récentes

Dans la dernière décennie, une expérience fondée sur la difficulté et la complexité passée et présente de la [C.L.P.](#) , une collection de méthodes générales ou spécifiques



à la C.L.P. a été proposée. Des exemples de problèmes majeurs et les solutions sont brièvement présentés dans la suite.

Le routage d'appels est un exemple important et concret de la catégorisation de messages parlés. Dans les applications de ce type, un acte de dialogue exprimé par une phrase est classé pour générer une action sémantique primitive appartenant à un ensemble bien défini. Une solution à la catégorisation de messages parlés est proposé dans [Gorin \(1995\)](#). Une représentation distribuée des connaissances est représentée par un réseau utilisé pour la correspondance des mots ou des phrases avec des actions. Un système déployé pour le routage d'appels, intitulé "How May I Help You" (HMIHY) est décrit dans [Gorin et al. \(1997\)](#).

Dans [Li et Chou \(2002\)](#) l'indexation sémantique latente dans l'acheminement des appels est proposé pour identifier l'objectif de l'énoncé. Les réseaux bayésiens des croyances sont proposés pour en déduire une intention communicative.

Diverses techniques ont été proposées pour la modélisation et la détection des actes de dialogue (DA). Parmi eux, il convient de mentionner les arbres de classification sémantique ([Mast et al. \(1996\)](#)), les arbres de décision ([Stolcke et al. \(1998\)](#); [Ang et al. \(2005\)](#)), les modèles de Markov cachés ( H.M.M. ) ([Stolcke et al. \(1998\)](#)), les modèles de Markov à fragments flous ([Wu et al. \(2002\)](#)), les réseaux neuronaux ([Stolcke et al. \(1998\)](#); [Wang et al. \(1999\)](#)), modèles d'entropie maximale ([Stolcke et al. \(1998\)](#); [Ang et al. \(2005\)](#)). L'étiquetage en Acte de Dialogue en utilisant l'apprentissage basé sur la transformation est proposée dans [Samuel et al. \(1998\)](#).

Les systèmes récents tentent de faire correspondre directement les paroles en actions [Potamianos et al. \(2005\)](#). Des méthodes pour relier les mots aux actes de dialogue ou les étiquettes sémantiques locales aux objectifs de la communication avec les réseaux de croyances sont proposées dans [Meng et al. \(1999\)](#).

Des modèles graphiques et D.B.N. pour la détection des Actes de Dialogues dans les systèmes de C.L.P. sont proposés dans [Ji et Bilmes \(2005\)](#). Pour la segmentation et la classification commune des actes de dialogues, une technique basée sur les modèles de langages à événements cachés "Hidden-Event Language Model" (HELM) est décrite dans [Zimmermann \(2005\)](#).

La détection exacte des événements n'est obtenue que si les frontières des phrases sont identifiées dans des messages oraux contenant plus d'une phrase. On considère souvent un événement par phrase. Les approches pour cette tâche ont utilisé des H.M.M. ([Shriberg et al. \(2000\)](#)) et les C.R.F. ([Liu et al. \(2005\)](#)). Dans [Roark et al. \(2006\)](#), un modèle de référence est utilisé pour produire N-meilleures listes de segmentations possibles.

Les modèles de langage basés sur l'analyse sémantique latente (Latent Semantic Analysis, L.S.A. ) capturent certaines relations sémantiques entre les mots. Le L.S.A. fait correspondre les mots et leur histoire dans un espace sémantique en utilisant la technique de décomposition en valeurs singulières (SVD) [Bellegarda \(2000\)](#). Les similitudes de mots sont mesurées avec des paramètres de distance tels que le produit scalaire entre les vecteurs. Une technique similaire a été proposée pour la formulation

d'hypothèses de composants sémantiques dans une phrase [Chu-Carroll et Carpenter \(1999\)](#). Dans [Zhang et Rudnicky \(2002\)](#), le L.S.A. est proposé pour intégrer les connaissances à de multiples niveaux. Dans le même ordre, une solution avec laquelle les améliorations pertinentes ont été observées dans les expériences de gros corpus, est proposée dans [Wang et al. \(2002\)](#). Les "Super Abstract Rôle Values" (SuperARV) sont introduits pour encoder les sources de connaissances multiples dans une représentation uniforme qui est beaucoup plus fine que des parties du discours ( P.O.S. ).

Le projet DARPA Communicator pour des dialogues parlés, comprenant des composants C.L.P. a été réalisé au début du 21<sup>ème</sup> siècle aux Etats-Unis. Des architectures intéressantes et des paradigmes ont été développés<sup>1</sup>, dont les évaluations sont rapportées dans [Walker et al. \(2002\)](#).

Dans [Thomae et al. \(2005\)](#), une hiérarchie des modèles de langages est proposée pour l'interprétation. L'introduction de l'information sémantique dans les modèles de langage est proposée dans [Erdogan et al. \(2005\)](#). Les modèles à états finis sont utilisés pour interpoler les scores obtenus avec un M.L. de concepts et un n-gramme général du M.L. . En plus de cela, l'information sémantique analysée est combinée avec un n-gramme en utilisant un modèle à deux niveaux statistiques. Les paramètres sont également utilisés pour le calcul de la probabilité conjointe d'une phrase et son analyse avec un modèle d'entropie maximale unique ( E.M. ).

L'interprétation peut bénéficier des collections de l'information linguistique. Dans l'étiquetage des rôles sémantiques, un lexique est utilisé, qui énumère les rôles possibles pour chaque argument syntaxique de chaque attribut. Un lexique des prédicats a été développé dans le projet "FrameNet" [Lowe et al. \(1997\)](#), et un lexique des verbes a été développé dans le projet "PropBank" [Palmer et al. \(2005\)](#). Dans le projet "VerbNet" [Kipper et al. \(2000\)](#), un lexique de verbes hiérarchisé conçu manuellement a été développé et basé sur le classement proposé dans le verbe [Levin \(1993\)](#). Pour chaque classe de verbes, VerbNet spécifie les cadres syntaxiques ainsi que le rôle sémantique assigné à chaque emplacement d'une frame. Diverses méthodes basées sur les paramètres ont été proposées pour identifier et classifier des prédicats et des arguments et pour l'extraction des relations. Dans le projet "WordNet" [Miller \(1995\)](#), un mot est représenté par un ensemble de sens synonymes appartenant à un alphabet de "synsets". La désambiguïsation du sens des mots peut être effectuée en utilisant le "WordNet" [Miller et al. \(1990\)](#) ou divers types de probabilités. [Zechner \(1998\)](#) propose une combinaison de l'analyse partielle, également appelé regroupement en blocs "chunks", avec la correspondance des arguments du verbe avec des frames sous-catégorisées qui peuvent être extraites automatiquement, par exemple, à partir de WordNet.

"MindNet" [Richardson et al. \(1998\)](#) produit une structure hiérarchique des relations sémantiques ("semrels") à partir d'une phrase à l'aide des mots dans un dictionnaire lisible par la machine. Ces structures sont inversées et reliées à chaque mot figurant dans les dictionnaires, permettant ainsi d'effectuer les correspondances et de calculer des similarités par propagation de l'activation. Les méthodes à noyau et les modèles d'entropies maximales ont été proposés à cet effet par [Moschitti \(2004\)](#). Une approche

1. <http://fofoca.mitre.org/> Galaxy Communicator

fondée sur la mémoire pour l'apprentissage des structures linguistiques utiles pour l'analyse syntaxique de surface est décrite dans [Argamon et al. \(1998\)](#).

Les résultats dans [Pradhan et al. \(2007\)](#) avec des classificateurs *S.V.M.*, ont montré qu'il y a une baisse significative des performances lors de l'entraînement et des tests sur des corpus différents.

L'utilisation combinées de *S.C.T.*, Adaboost et des classificateurs *S.V.M.* est proposé dans [Raymond \(2005\)](#) pour augmenter la robustesse de la classification en utilisant le consensus classifieur comme un indicateur de confiance.

## 4.2 Architectures avancées

### 4.2.1 Le système de Microsoft

Dans [Wang et Acero \(2001, 2002\)](#); [Wang et al. \(2002\)](#), les structures sémantiques sont définies par des schémas. Chaque schéma est un objet. Les structures d'objets sont définies par un schéma XML. Étant donné un schéma sémantique, une grammaire sémantique indépendante du contexte est calculée à partir des modèles. Les détails des schémas sont appris automatiquement. Une entité est l'élément de base d'un schéma qui définit les relations entre les entités. Une entité se compose d'une tête, des modificateurs optionnels et des propriétés optionnelles définies récursivement de sorte qu'ils intègrent une séquence différente d'emplacements du schéma. Chaque emplacement est délimité par un préambule optionnel et un postambule qui sont à l'origine, des porteurs d'emplacements. Certains porteurs d'emplacements sont spécifiés par des motifs de *P.O.S.*

Comme les grammaires sont mises en place à partir de schémas par des motifs, la tâche d'apprendre la grammaire devient l'apprentissage d'expressions des pré-terminaux comme les têtes, commandes, préambules etc. La sémantique de surface représente les intentions des utilisateurs, alors que la sémantique discursive représente les croyances système. La sémantique de surface d'une phrase est représentée par un arbre obtenu par une activité d'analyse. Les nœuds sont des objets sémantiques qui sont des incarnations linguistiques des entités de domaine, les fonctions de domaine, des entités telles que le numéro, les devises, l'heure, la date. La tâche d'évaluation sémantique est de découvrir ce que chaque objet sémantique est censé représenter. L'algorithme parcourt l'arbre en profondeur d'abord en fonction du type de l'objet qu'un expert a invoqué et qui renvoie à l'exécution d'une fonction ou une instance d'objet. Des objets et des entités sémantiques de domaine peuvent être générés avec une interface graphique. Une phrase donnée peut générer une annotation qui, avec la *C.F.G.* génère un arbre d'analyse partielle lorsque les bornes correspondent à des pré-terminaux.

Dans [Wang et Acero \(2002\)](#), les grammaires sémantiques (portabilité, mais couverture limitée) sont combinées avec des classificateurs pour la reconnaissance des concepts et en particulier les 6 tâches dans *A.T.I.S.*

Les classifieurs sont les suivants :

- Classifieur Bayésien Naïf « Naive Bayes » (entrée : un vecteur à un élément pour chacun des mots : 1 si le mot est présent, 0 si absent).
- Machine à vecteur de support ( *S.V.M.* ), un par concept
- n-grammes
- leur combinaison avec ROVER (on prend l’hypothèse qui remporte la majorité des voies)

Les *S.V.M.* seuls semblent être meilleurs même si ROVER est appliqué. D’importantes améliorations ont été apportées en remplaçant certains mots avec leur catégorie sémantique trouvée par l’analyseur. Utiliser des concepts détectés de cette manière pour filtrer les règles de grammaires sémantiques qui seront appliquées pour trouver les emplacements, a donné lieu à une importante amélioration. Dans [Wang et Acero \(2003\)](#) d’autres améliorations sur le système Microsoft sont présentées.

Après l’alignement du texte pré terminal, l’appariement est tenté, si des ambiguïtés apparaissent, le concepteur humain est invité à les résoudre. Ensuite, un outil dérive une grammaire générative de production des phrases pour ces concepts. La grammaire est appelée *grammaire sémantique*.

Essentiellement un pré-terminal est généré pour le nom d’une frame et de chaque emplacement. Pour chaque nom d’emplacement, un préambule et un postambule pré-terminaux sont générés.

Par exemple : Soit la structure de frame  $A[a_1, a_2]$ , associé une grammaire sémantique :

$$A \rightarrow ACmd[A_{properties}]$$

$$A_{properties} \rightarrow A_{property}[A_{properties}]$$

$$A_{property} \rightarrow a_{1_{property}} | a_{2_{property}}$$

$$a_{1_{property}} \rightarrow Pre_{a_1} Term_{a_1} Post_{a_1}$$

$$a_{2_{property}} \rightarrow Pre_{a_2} Term_{a_2} Post_{a_2}$$

$ACmd, Pre_{a_1}, Term_{a_1}, Post_{a_1}, Pre_{a_2}, Term_{a_2}, Post_{a_2}$  sont des pré-terminaux pour lesquels des règles de réécriture doit être inférées pour décider quelle séquence de mots, elles doivent réécrire.

Un corpus d’entraînement est fourni avec des phrases et des séquences de pré-terminaux. La segmentation est un processus impliquant des états cachés qui peut être exercé par l’algorithme de Viterbi, une fois que les probabilités de réécriture des pré-terminaux en séquences de mots, sont connues. Ces probabilités peuvent être obtenues avec un algorithme d’entropie maximale.

[Wang et Acero \(2006\)](#) examine la relation entre ces modèles, compare leur performance en termes de précision, vitesse d’entraînement et la robustesse à faible densité de données.

Les auteurs convertissent le problème SLU dans un problème séquentiel d’étiquetage qui attribuent une étiquette de domaine liée à chaque mot dans un énoncé.

Formellement le problème peut être vu comme l’assignation d’une séquence d’états  $s_1^T$  à une observation  $o$  avec  $\tau$  mots, à l’aide d’une analyse avec grammaire indépen-

dante du contexte (C.F.G.), qui identifie toutes les correspondances possibles des règles C.F.G. pour le remplissage des emplacements. La séquence d'états désirée  $s_1^T$  devrait avoir la probabilité postérieure maximale  $P(s_1^T|o; \lambda)$  suivant le modèle  $\lambda$ . Dans ce cas, des modèles de graphes conditionnels non-dirigés sont utilisés pour les postérieurs. Ils suivent la forme suivante :

$$P(s_1^T|o; \lambda) = \frac{1}{z(o; \lambda)} \exp(\lambda \cdot f(s_1^T, o)) \quad (4.12)$$

où  $f(s_1^T, o)$  est un vecteur des caractéristiques qui sont des fonctions de séquences d'états et d'observation ;  $\lambda$  est un vecteur de paramètres qui sont des poids pour les caractéristiques ;  $z(o; \lambda) = \sum_{s_1^T} \exp(\lambda \cdot f(s_1^T, o))$  normalise la distribution sur toutes les séquences d'états possibles.

Les caractéristiques utilisées pour la Compréhension du Langage Parlé, dans le modèle conditionnel, incluent :

1. Les caractéristiques "commande prioritaire" capturent la vraisemblance qu'une commande est passée par un utilisateur :

$$f_{ShowFlight}^{PR}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{if } t = 0 \wedge Prefix(s^{(t)}) = SF \\ 0 & \text{sinon} \end{cases} \quad (4.13)$$

2. Les caractéristiques de "transition d'état" capturent l'ordre des différents emplacements dans la commande :

$$f_{SF.DCity, SF.ACity}^{TR}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } s^{(t-1)} \in \{SF.DCity.start, SF.DCity.cont\} \\ & \wedge s^{(t)} \in \{SF.ACity.pre, SF.ACity.start\} \\ 0 & \text{sinon} \end{cases} \quad (4.14)$$

3. Les caractéristiques uni-grammes (UG) et bi-grammes (BG) capturent les co-occurrences de mots avec préambule/état de post-commandes :

$$f_{SF.DCity.pre, from}^{UG}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } s^{(t)} = SF.DCity.pre \wedge o^t = from \\ 0 & \text{sinon} \end{cases}$$

$$f_{SF.DCity.pre, departing, from}^{UG}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } s^{(t-1)} = SF.DCity.pre \\ & \wedge o^{t-1} = departing \wedge o^t = from \\ 0 & \text{sinon} \end{cases} \quad (4.15)$$

4. Les caractéristiques de "contexte des emplacements précédents" avec une taille de fenêtre  $k$ , capturent les dépendances des interprétations des emplacements dans le préambule du précédent emplacement.

$$f_{SF.ACity, SF.DTime, w}^{PC}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } s^{(t-1)} \in \{SF.ACity.start, SF.ACity.cont\} \\ & \wedge s^t \in \{SF.DTime.start, SF.DTime.pre\} \\ & \wedge w \in \Theta(SF.ACity, o, t-1, k) \\ 0 & \text{sinon} \end{cases} \quad (4.16)$$

Ici  $\Theta(SF.AC_{ity}, o, t - 1, k)$  représente un ensemble de mots qui contient jusqu'à  $k$  mots, qui ne sont pas couverts par les règles des C.F.G. et qui apparaissent en avant des plus longues phrases couvertes par la règle C.F.G. , pour l'emplacement SF.AC<sub>ity</sub>.

5. Les caractéristiques de couvertures par morceaux des C.F.G. , pour les mots en préambule/ de post commandes, capturent la vraisemblance qu'un mot couvert par une règle de C.F.G. peut ne pas être une partie d'un remplisseur d'emplacements.

$$f_{ShowFlight, City}^{CC1}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } Prefix(s^{(t)}) = SF \\ & \wedge covers(City, o^t) \wedge \neg IsFiller(s^{(t)}) \\ 0 & \text{sinon} \end{cases} \quad (4.17)$$

Ici  $IsFiller(s)$  indique que l'état  $s$  est soit un état remplisseur d'emplacement, soit un état d'entrée, soit un état de continuation. La caractéristique est active quand le "non-remplisseur"  $o^t$  est couvert par la règle "City".

6. Les caractéristiques de couvertures par morceaux des C.F.G. , pour les limites d'emplacements préviennent des erreurs comme la segmentation de "Washington D.C." en deux emplacements distincts. Elles sont activées quand une limite d'un emplacement est couverte par une règle C.F.G. .

$$f_{ShowFlight, City}^{CC2}(s^{(t-1)}, s^{(t)}, o, t) = \begin{cases} 1 & \text{si } Prefix(s^{(t)}) = SF \wedge covers(City, o_{t-1}^t) \\ & IsFiller(s^{(t-1)}) \wedge IsFilleStart(s^{(t)}) \\ 0 & \text{sinon} \end{cases} \quad (4.18)$$

Chaque caractéristique  $f_k$  est associée à un poids  $\lambda_k$ . Les paramètres du modèle peuvent être optimisés en tenant compte des différentes fonctions objectifs des méthodes d'apprentissage discriminant.

Les auteurs ont introduit différents critères discriminants pour les modèles d'apprentissage pour le C.L.P. , et ont comparé les C.R.F. , les perceptrons, les systèmes à erreurs de classification minimum ( M.C.E. ) et les modèles à larges marges ( M.L.M. ) en termes de précision, vitesse d'entraînement et la robustesse à la dispersion des données. Tous les modèles discriminants ont accompli des précisions similaires, sauf pour le perceptron, qui a un taux d'erreurs beaucoup plus élevé. Les C.R.F. sont aussi plus robustes lorsque moins de données d'apprentissage sont disponibles. Toutefois, les Perceptrons et perceptrons initialisés avec M.C.E. et les modèles M.L.M. sont beaucoup plus rapides à entraîner, et l'écart de précision devient plus petit lorsque plus de données sont disponibles. Il est un bon compromis pour utiliser perceptron initialisée avec M.C.E. ou M.L.M. si la vitesse d'entraînement est essentielle pour une application, où une grande quantité de données d'entraînement est disponible.

La possibilité de concevoir des modèles de langages structurés (SLM) basés sur les C.F.G. structurées a été étudiée dans Chelba et Jelinek (2000). Le modèle utilise des clusters  $\Phi(W_1^{i-1})$  d'historiques d'applications de règles trouvées par un analyseur qui



calcule la probabilité suivante :  $P(w_i|\Phi(W_1^{i-1})) = \sum_T P(w_i T_i | W_1^{i-1})$  où les  $T_i$  sont les structures trouvées par l'analyseur.

Des analyseurs statistiques ont été proposés pour découvrir des structures syntaxique aussi bien que pour la prédiction de chaîne de mots, [Roark \(2001\)](#); [Charniak \(2001\)](#); [Collins \(2003\)](#). Avec ces analyseurs la probabilité qu'une grammaire génère toute la séquence de mots peut être calculée, permettant ainsi une couverture complète (mais pas toujours un résultat d'analyse correct) de chaque message en langage naturel.

A l'Université de Cambridge [He et Young \(2003b\)](#), des modèles statistiques de production de phrases ont été proposés qui ne nécessitent pas l'entière annotation des données pour l'entraînement. En conséquence, l'état exact, où la lecture de l'en-tête est sur une position précise de mot, n'est pas connu, fait de l'état une variable cachée. Plus de détails sur cette approche peuvent être trouvés dans le chapitre [Wang \(2003\)](#).

Le re-calcule du treillis de mots avec un analyseur statistique est proposé dans [Roark \(2002\)](#). L'analyse dirigée par les en-têtes pour les treillis de mots est proposée dans [Collins et al. \(2004\)](#).

La question de combiner la grammaire basée sur les données et les approches axées sur les données a été examinée dans [Rayner et Hockey \(2003\)](#); [Wang et Acero \(2003\)](#).

### 4.2.2 Le système d'AT & T

Dans [Bangalore et Johnston \(2004\)](#), une grammaire indépendante du contexte, est utilisée pour introduire des interprétations à partir des treillis de mots. Un modèle de langage mélangé pour une application multimodale est décrit avec un composant entraîné avec des données du domaine et un autre avec les données générées par une grammaire. La compréhension est la reconnaissance de la séquence de prédicats / arguments, étiquettes de concepts qui maximise  $P(C|W)$ . Un rapprochement est fait en tenant compte de bi-grammes et tri-grammes de concepts de Dix prédicats, tels que *l'aide*, *l'assertion*, *la demande d'information*, ont été considérés avec vingt types d'arguments comme *la cuisine*, *le prix*, *la localisation*. Ils sont hypothésés en utilisant des classificateurs Adaboost. Les entrées N-grammes figurant dans l'énoncé multimodal sont utilisées comme des classificateurs faibles pour les besoins du classement des prédicats.

Les approches à base de règles souffrent de leur fragilité et le coût élevé de création et de maintien des ensembles de règles complexes. Les approches pilotées par les données sont robustes et fournissent un simple processus de développement d'applications compte tenu des données du domaine d'application. Toutefois, le recours à des données spécifiques à un domaine est aussi une des limites importantes des approches pilotées par les données. Le développement d'un système de conversation en utilisant des approches pilotées par les données ne peut pas se faire tant que les données relatives au domaine d'application ne sont pas disponibles.

Afin de surmonter le décalage possible entre l'entrée de l'utilisateur et le langage codé dans la grammaire multimodale, une distance d'édition est utilisée, basée sur un

algorithme de filtrageLa chaîne, avec le moins de modifications qui peuvent être assignées d'une interprétation par la grammaire, est sélectionnée.

Une grammaire déclarative multimodale et indépendante du contexte est utilisée dans laquelle chaque terminal est un triplet  $W : G : M$ , composé d'expression (mots,  $W$ ), le geste (symboles gestuels,  $G$ ), et le sens (signification des symboles,  $M$ ). La grammaire multimodale n'encode pas seulement des modèles d'intégration multimodale, mais aussi la syntaxe de la parole et du geste, et l'attribution de sens. Le sens est représenté en XML, ce qui facilite l'analyse et l'exploitation des arbres par les autres composants du système. Le symbole *SEM* est utilisé pour abstraire sur les contenus trop spécifiques, comme l'ensemble des points délimitant une zone ou les identificateurs d'objets sélectionnés. Le sens est constitué par la concaténation des symboles de sens et remplaçant les symboles *SEM* avec le contenu spécifique approprié.

Il y a eu un certain nombre d'implémentations de calcul de large couverture, indépendant du domaine, des grammaires syntaxiques pour l'anglais dans différents formalismes [Joshi \(2001\)](#). Une telle mise en œuvre est le formalisme de grammaire lexicalisée d'arbres adjoints (Lexicalised Tree Adjoining Grammar [L.T.A.G.](#) ). Une [L.T.A.G.](#) se compose d'un ensemble d'arbres élémentaires (Supertags) [Bangalore et Joshi \(1999\)](#) chacun associés à un élément lexical. L'ensemble des phrases générées par un [L.T.A.G.](#) peut être obtenu en combinant des supertags utilisant la substitution et les opérations d'adjonction.

Dans des travaux connexes ([Rambow et al. \(2002\)](#)), il a été démontré que, pour une version restreinte de [L.T.A.G.](#) , les combinaisons d'un ensemble de supertags peuvent être représentées comme une machine à états finis ([F.S.M.](#) ). Ce [F.S.M.](#) code et compacte l'ensemble des phrases générées par une grammaire [L.T.A.G.](#) . Un corpus de domaine spécifique est dérivé par la construction d'un lexique composé de paires de mots avec leurs supertags qui sont pertinents pour ce domaine. La grammaire est ensuite compilée pour construire un [F.S.M.](#) de toutes les condamnations à une longueur donnée. Ce [F.S.M.](#) est ensuite échantillonné et un modèle de langage est construit. Compte tenu de déclarations non transcrites à partir d'un domaine spécifique, il est également possible d'adapter le modèle de langage.

Un modèle de langage mélangé est obtenu avec le modèle de langage formé par les données de domaine et celle obtenues avec les données générées.

La compréhension est la reconnaissance de la séquence de balises principales argument qui maximise  $P(T | S)$  où  $T$  est la séquence de balises et  $S$  de la phrase.

Un rapprochement est fait en tenant compte de bi-grammes et tri-grammes de balises, et un classificateur AdaBoost qui donne :

$$P(t|x) = \frac{1}{1 + e^{-2\sum(w_i f_i(x))}} \quad (4.19)$$

où  $w$  sont les poids  $f$  est la contribution de chaque classificateur faible.

Le problème de l'identification et de l'extraction des arguments d'une entrée multimodale est considérée comme un problème d'associer à chaque signe de l'entrée avec



un tag spécifique qui encode l'étiquette de l'argument et la durée de l'argument. Ces balises sont tirées d'un ensemble d'étiquettes qui est construit par l'extension de l'étiquette de chaque argument par trois symboles additionnels suivants : intérieur, extérieur, cas limite [Ramshaw et Marcus \(1995\)](#). Ces symboles correspondent aux cas où un symbole est l'intérieur d'un segment, en dehors d'un segment ou à la limite de deux segments. Une autre tentative d'approche pour surmonter l'inadéquation possible entre l'entrée de l'utilisateur et la langue codée dans la grammaire multimodale est étudiée. Une distance d'édition est utilisée, basée sur un algorithme de filtrage pour contraindre l'ensemble des chaînes codées dans le réseau résultant de la [R.A.P.](#) pour correspondre à l'une des chaînes qui peut être assignée à une interprétation. Les opérations d'édition (insertion, suppression, substitution) peuvent être basées sur les mots. Les opérations d'édition sont codées comme un transducteur et peuvent s'appliquer à l'une des meilleures sorties du réseau de la [R.A.P.](#) . La chaîne avec le moins de modifications qui peut être affectée d'une interprétation par la grammaire est sélectionnée.

La performance du modèle de langage entraîné sur des données échantillonnées à partir de la grammaire est nettement améliorée par rapport aux performances d'une grammaire conçue à la main.

L'approche par reconnaissance de modèle (appelée aussi l'approche par filtrage), basée sur une approche de compréhension robuste interpose l'inadéquation entre les chaînes de caractères de la sortie de la [R.A.P.](#) et les chaînes affectées à l'interprétation. Des expériences sur la meilleure sortie du système de [R.A.P.](#) , ont été réalisées avec la reconnaissance de modèles basée sur les mots, ainsi que la reconnaissance de modèles téléphonique. L'approche de filtrage de compréhension robuste améliore la précision sur le concept de base de façon significative.

En outre, la méthode reconnaissance de modèle téléphonique a une performance similaire à la méthode basée sur le mot correspondant.

Les uni-grammes, bi-grammes et tri-grammes figurant dans l'énoncé multimodal sont utilisés comme classifieurs faibles pour les fins de la classification principale. Afin de prédire l'étiquette d'un mot pour l'extraction de l'argument, le contexte tri-gramme gauche et à droite et les étiquettes pour les deux précédents jetons comme classifieurs faibles sont utilisés.

Les deux approches à la compréhension robuste surpassent le modèle de base de façon significative.

Toutefois, il est intéressant de noter que tandis que l'approche par filtrage a une meilleure précision sur l'extraction d'arguments, l'approche de la classification a une meilleure précision sur l'identification de prédicats. Deux raisons possibles pour cela sont : en premier lieu, l'extraction d'arguments exige plus d'informations non-locales qui sont disponibles dans l'approche par filtrage alors que l'approche axée sur une classification s'appuie sur des informations locales et est plus propice à l'identification des prédicats simples.

Deuxièmement, l'approche par filtrage utilise la grammaire entière comme un modèle pour l'appariement tandis que la méthode de classification est entraînée sur des

données en nombre nettement inférieure par rapport au nombre d'exemples encodées dans la grammaire assignée pour la récupération de fragments et l'intégration les opérations qui combinent des fragments pour dériver du sens à partir de l'énoncé reconnu.

La question de la combinaison d'approches fondées sur des règles et pilotées par les données a reçu moins d'attention, à l'exception de quelques-uns [Wang et al. \(2000\)](#); [Rayner et Hockey \(2003\)](#); [Wang et Acero \(2003\)](#). Dans un article récent [Rayner et Hockey \(2003\)](#), résoud ce problème en employant un système de Compréhension du Langage Parlé basé sur des listes de décisions en tant que moyen de progresser à partir des modèles fondés sur des règles vers des modèles pilotés par les données lorsque les données deviennent disponibles. Le système utilisant des listes de décisions fournit également une méthode pour la compréhension robuste. En revanche, cette approche peut être utilisée sur des treillis de parole pour produire un treillis de représentations du sens.

### 4.2.3 Le système d'IBM

[Sarikaya et al. \(2004\)](#) propose un système qui génère une liste de N-meilleures hypothèses de mots (N=34) avec un modèle de langage tri-gramme dépendant de l'état du dialogue et qui recalcule les hypothèses avec deux modèles sémantiques.

Une grammaire sémantique indépendante embarquée ( [E.G.](#) ) est définie pour chacun des 17 concepts et exécute le repérage des concepts par la recherche de modèles de phrase correspondant à des concepts. Par conséquent, les hypothèses sémantiques sont générées en remplissant un certain nombre d'emplacements dans une représentation en frames. La décision entre ces hypothèses est basée sur une couverture de mots maximum.

Les probabilités tri-grammes sont utilisées pour calculer des hypothèses avec le modèle [E.G.](#) . Les balises de concepts sont placées au début et en fin de phrases correspondantes dans une séquence d'hypothèses et des probabilités de mots tri-grammes sont du type  $P(de|/MONTANT, vendre)$ , où / indique la fin de la phrase correspondant à la notion de MONTANT. Le score résultant d'une hypothèse est  $P(W, C)$ .

Un deuxième modèle de langage ( [M.L.](#) ), appelé entropie maximale ( [E.M. M.L.](#) ) ou ( [M.E.L.M.](#) ), calcule les probabilités d'un mot, étant donné l'historique, en utilisant un modèle [E.M.](#) comme suit :

$$P(o_i|h_i) = \frac{e^{\left(\sum_i^{N_j} \lambda_j f_j(o_i, h_i)\right)}}{\sum_{o'} e^{\left(\sum_{j=1}^{N_j} \lambda_j f_j(o', h_i)\right)}} \quad (4.20)$$

$o_i$  est le mot courant,  $f_i(o_i, h)$  est l'indicateur de caractéristiques activées par un historique. Le lissage "Fuzzy [E.M.](#) " est utilisé comme proposé dans [Chen et Rosenfeld \(2000\)](#). Il ajoute une gaussienne au préalable pour chaque paramètre. L'écart entre les

paramètres est le même pour toutes les gaussiennes et est entraîné avec l'algorithme de Powell.

Avec ce modèle, il est possible de calculer  $P(W, C)$ . La première étape pour construire le modèle **E.M.** est de représenter un arbre d'analyse obtenu avec la Grammaire **E.G.** par une séquence de mots et des étiquettes. Les règles de grammaire effectuent des compositions de concepts. Chaque élément de la séquence linéaire de symboles est un symbole.

Les caractéristiques du modèle **E.M.** sont des questions sur n-grammes de symboles, l'étiquette parente actuelle, le nombre de mots à la gauche depuis le début du concept actuel, le symbole du mot précédent, les précédents constituants complétés et le nombre de mots à la gauche depuis la fin de celui-ci. Le score de ce **M.L.** est utilisé comme mesure de confiance. Le **M.L.** est entraîné avec 28300 phrases annotées. Il y a 17 grammaires (concepts). Le taux d'erreurs mots (**W.E.R.**) oracle est de 16,2%, **W.E.R.** base est de 28,1%, **E.G. W.E.R.** est de 23,8%, **M.E.L.M. W.E.R.** est de 23,7%, combinaison linéaire entre eux donne 22,8%. Les probabilités a posteriori sont basées sur des "saucisses" obtenues à partir du graphe de mots. Le meilleur chemin de la "saucisse" est émis en hypothèse comme la sortie **R.A.P.**. Chaque phrase est marquée avec les **E.G.** et **M.E.L.M.** pour attribuer la probabilité sémantique à chaque mot. Les caractéristiques sémantiques correspondantes sont extraites de chaque mot.

Un arbre de décision (DT) est construit en utilisant les caractéristiques des mots dans le but de séparer les mots corrects et incorrects. Le DT utilise le score brut des caractéristiques respectives. Il apprend des combinaisons de caractéristiques permettant de prévoir l'acceptation ou le rejet. Ces fonctions sont utilisées pour le calcul des mesures de confiance. La probabilité a posteriori (post) ainsi que les scores des **E.G.** et des **M.E.L.M.** sont utilisés pour décider de l'acceptation ou le rejet.

Dans les versions précédentes du système d'IBM, une classe de base est considérée comme un modèle de langage avec 41 classes. Des grammaires indépendantes du contexte (**C.F.G.**) représentent les relations entre les concepts et les mots. Toutes ces **C.F.G.** et les mots sont combinés en un transducteur à états finis qui produit une analyse de concepts.

La première étape d'analyse est la classification. Les mots sont étiquetés en fonction de leur sens. Seuls les mots voisins sont regroupés par le classifieur. Cela est fait par un arbre de décision et les résultats dans une représentation arborescente d'une profondeur maximale de 2. Des structures plus complexes sont obtenues par l'analyseur qui utilise les règles de composition manuelle dérivés.

L'analyseur apprend les caractéristiques d'une phrase qui conduisent à certaines balises, étiquettes ou extensions. Classifier peut être considéré comme l'analyse superficielle. La sortie du classifieur est utilisée comme une entrée dans l'analyseur. Par conséquent, l'analyse est par essence un processus en deux étapes. La fonction de la classification est de regrouper les mots qui font partie d'un concept. L'analyseur prend la sortie du classifieur et construit un arbre hiérarchique sémantique complet. Les concepts sémantiquement connexes sont regroupés à un niveau supérieur. L'analyseur statis-

tique utilise les données d'entraînement afin d'examiner les phrases d'entraînement pour trouver les meilleures combinaisons d'indices dans les phrases qui fonctionnent le mieux dans toutes les phrases de l'entraînement.

Un arbre d'analyse est représenté comme un graphe connecté, avec une racine et des valeurs caractéristiques à chaque nœud. La probabilité d'une assignation de la valeur caractéristique à un nœud particulier, en principe, est conditionnée par l'information disponible à d'autres nœuds dans l'arbre partiellement construit. L'analyseur assigne la probabilité d'un arbre d'analyse  $T$  étant donné la phrase  $S_{P(TS)}$ .

La méthode d'entropie maximum ( **E.M.** ) présente un cadre pour combiner plusieurs sources d'information qui se chevauchent de façon efficace. Chaque source de connaissances donne lieu à un ensemble de contraintes qui est imposé sur le modèle combiné. Ces contraintes sont généralement exprimées en termes de distributions marginales. L'intersection de toutes les contraintes contient un ensemble de fonctions de probabilité qui sont consistantes avec les sources de connaissances ou les caractéristiques attendues. Parmi ces fonctions de probabilité, l'approche **E.M.** choisit la fonction avec la plus grande entropie. L'entropie maximum a été largement utilisée dans la modélisation statistique du langage. En raison de la commodité de combiner de multiples sources d'information, l' **E.M.** a aussi été utilisée pour la modélisation du langage syntaxique structuré et la modélisation du langage sémantique structural. La modélisation de l'Entropie Maximum fait correspondre les hypothèses de caractéristiques tout en faisant aussi peu que possible d'hypothèses dans le modèle. Elle constitue un cadre unifié pour combiner le contenu lexical et sémantique et la structure de la phrase d'une manière efficace.

Par conséquent, on peut attribuer une probabilité conjointe de chaque mot en fonction de ses historiques lexicaux et sémantiques. Un classifieur et un analyseur syntaxique sont utilisés pour extraire le contenu sémantique d'une phrase. Dans le thème cadre, on peut voir chaque source d'informations comme pouvant définir un ou plusieurs sous-ensembles de l'espace d'événements. Une contrainte est associée à chaque sous-ensemble pour répondre aux statistiques des données d'entraînement sur ce sous-ensemble de l'espace événementiel. Les phrases basées sur des niveaux supérieurs de caractéristiques sémantiques sont ensuite ajoutées au modèle de langage à l'aide de l'entropie maximale ( **E.M.** ) pour recalculer les  $n$ -meilleures hypothèses générées avec le modèle précédent. Les caractéristiques sont fonction d'un mot et de son histoire qui à son tour est représenté par une fonction contextuelle :

Le modèle de langage calcule  $P(W)$  comme suit :

$$P(W) = \sum_{C'} P(W, C') \approx P(W, C) = \sum_{i=1}^I P(t_i | t_1^{i-1}) P(t_i | h_i) = \frac{e^{\sum_{j=1}^{N_j} \lambda_j f_j(t_i, h_i)}}{Z} \quad (4.21)$$

$t_i$  est un symbole qui peut être un mot ou une étiquette trouvée par l'analyseur sémantique une fois que la chaîne analysée est représentée dans une sorte de forme d'arbre de données ("Tree-Bank") avec ses étiquettes. Ce modèle, appelé **M.E.L.M. 2** est

utilisé pour recalculer la liste des  $N$  meilleures étiquettes trouvées avec le modèle initial. **M.E.L.M.** 1 a des questions pour les éléments liés uniquement à des uni-grammes, bi-grammes, tri-grammes et 4-grammes qui sont également intégrés au modèle de Maximum d'Entropie. Dans de nombreux cas, les classes tri-grammes donnent de meilleurs résultats que **M.E.L.M.** et seule une combinaison de ceux-ci conduit à des améliorations mineures.

La première étape dans la construction du modèle d'entropie maximale est de représenter un classifieur / arbre d'analyse comme une séquence de mots, mots clés et des étiquettes. Les étiquettes sont réparties comme "commencement du label" et "fin du label".

Cette représentation nous permet de définir les limites pour les constituants sémantiques et de prendre en compte à long terme des informations sémantiques. Chaque symbole est un résultat du modèle mixte. Puisque les balises sont déjà incluses dans les classes utilisées dans le langage de modélisation basé sur des classes, nous les avons ignorés dans notre analyse. Voir également [Gu et al. \(2006\)](#).

Dans [Thomae et al. \(2005\)](#), la première étape pour construire le modèle d'Entropie Maximum est de représenter un arbre d'analyse comme une séquence de mots et d'étiquettes. Mais, l'analyse est obtenue avec une grammaire hors-contexte, et ne fait pas de captures des nombreuses dépendances contextuelles. En outre, si la structure de phrase pour un concept est une séquence de modèle de langage, il devrait y avoir une fonction caractéristique pour chaque chemin d'accès possible, tout cela est bien intégré dans un **F.S.M.** stochastique (**S.F.S.M.** .) Chaque chemin d'un **S.F.S.M.** devrait être représenté par une caractéristique. Une structure sémantique se produit à un état spécifique / de transition plutôt que d'être obtenue par un moteur de recherche sur les symboles associés à des mot qui est plutôt associé à des modèles de phrases. Les dépendances parmi les nombreuses mesures de confiance / situations sont apprises par des arbres de décisions. Les scores des modèles de langages sémantique sont combinés avec les probabilités a posteriori sur les mots.

### 4.2.4 Le projet Européen LUNA

Dans le projet européen LUNA<sup>2</sup>, les recherches de la **C.L.P.** ont été menées en utilisant un schéma d'architecture de logiciel montré dans la figure 4.1.

Un treillis de mots produit par un système ASR est utilisé pour générer des hypothèses d'étiquettes de concepts dans une couche logicielle indiquée comme  $C_{WP2}$ . Les étiquettes de concepts sont ensuite utilisées pour générer des fragments sémantiques qui sont composés dans la couche indiquée comme  $C_{WP3}$ . Les hypothèses de structure sémantique sont traitées ultérieurement et re-classées en utilisant l'information de l'historique du dialogue dans une couche indiquée comme  $C_{WP4}$ . À chaque niveau, les sources de connaissances sont utilisées et mises à jour en utilisant les méthodes actives

---

2. <http://www.ist-luna.eu>.

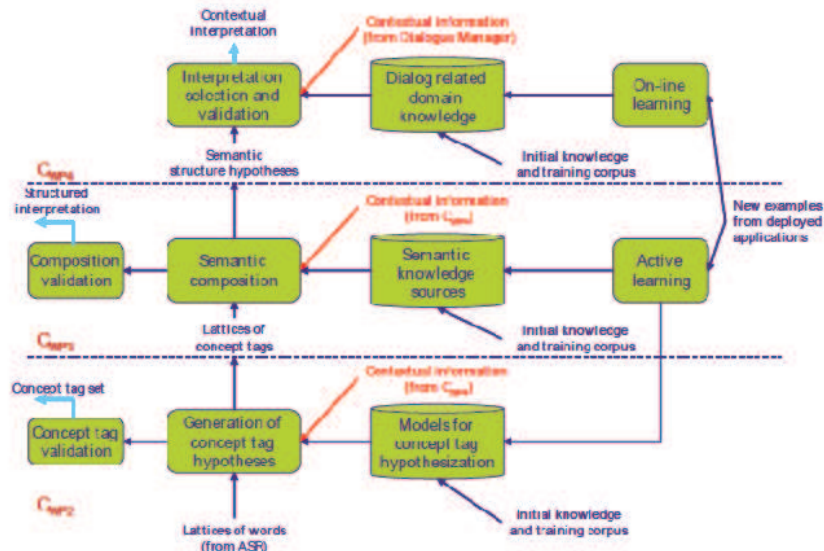


FIGURE 4.1 – Schéma d'architecture de logiciel utilisé par le projet LUNA

d'apprentissage tel que décrites dans Hakkani-Tür et al. (2006). La notation et la validation d'hypothèse sont réalisées en utilisant les indicateurs de confiance appropriés.

Différentes méthodes ont été comparées pour la formulation d'hypothèses de concepts à partir d'hypothèses de mots. Le corpus français MEDIA, corpus de dialogue téléphonique de négociation a été utilisé pour l'entraînement et les méthodes de test basées sur les modèles et les classificateurs suivants :

- des modèles d'entropie maximum pour des traductions automatiques statistiques (S.M.T.)
- des transducteurs à états finis (F.S.T.)
- des machines à vecteurs de support (S.V.M.)
- des réseaux bayésiens dynamiques (D.B.N.)
- des champs de conditions aléatoires (C.R.F.)
- des modèles log-linéaires au niveau positionnel (LLPos)

Les C.R.F. fournissent les meilleurs résultats dans l'interprétation des transcriptions manuelles ainsi que des séquences d'hypothèses de mots générés par un composant R.A.P. . On utilise le même lot d'entraînement, de développement et de tests pour toutes les expériences. Les résultats obtenus sur les noms de balises concepts et des valeurs pour l'ensemble de test sont présentés dans le tableau 4.1. Les expériences ont été réalisées avec 74 balises de concepts. L'entraînement et le corpus de test contiennent respectivement 12.908 et 3518 tours de dialogue téléphonique. Les résultats sur les énoncés de transcriptions sont visés en tant que texte alors que les résultats obtenus avec la meilleure séquence de mots générés par un système R.A.P. (avec un taux d'erreur de mots de 27,4%) sont considérés comme des discours.

Dans une première approche, les compositions par la fusion et l'attachement ont été

<b>Corpus MEDIA</b> <b>modèle</b>	Nom de concept		Nom de concept & valeur	
	<b>texte</b>	<b>parole</b>	<b>texte</b>	<b>parole</b>
C.R.F.	11.5	24.3	14.6	28.8
F.S.T.	14.1	27.5	16.6	31.3
LLPos	14.7	27.5	17.7	32.1
D.B.N.	15.6	29.0	18.3	33.0
S.V.M.	16.5	29.2	19.2	33.1
S.M.T.	15.2	28.7	23.4	34.9

**TABLE 4.1** – Résultats de l'étiquetage en Concept à taux d'erreurs de concept ( *C.E.R.* ) obtenus avec différents modèles et méthodes sur le corpus français *MEDIA* .

réalisées par des règles comme décrit au chapitre 5. On trouve les détails des résultats dans l'appendice A.



## **Deuxième partie**

# **Compréhension de la parole et compositions sémantiques**





## Chapitre 5

# Composition sémantique

### Sommaire

---

<b>5.1</b>	<b>Connaissance fragmentée et composition sémantique</b>	<b>106</b>
5.1.1	Etiquettes “concepts” et connaissance fragmentée	106
5.1.2	Composition par fusion de fragments	110
5.1.3	Composition par attachements	111
5.1.4	Composition par attachements et inférences	112
5.1.5	Probabilités de frames	113

---

Les travaux menés sur la compréhension automatique de la langue naturelle montrent la difficulté pour un système informatique de modéliser, comprendre ce que dit l'homme. L'interprétation de la langue naturelle écrite semble être plus facile à appréhender pour un système informatique car, elle est écrite avec une syntaxe et une grammaire rigoureuses et dans la plupart des cas, les systèmes exécutent une analyse syntaxique profonde (voir chapitre 2), avec des grammaires du langage. L'utilisation de ces approches pour la langue naturelle orale, est beaucoup plus compliquée et moins performante car la transcription de la langue orale est soumise à beaucoup d'erreurs liées à la reconnaissance de la parole (bruit, qualité du signal vocal et de la transcription) ou disfluences dans l'énoncé, auto-corrections, phrases asyntaxiques, agrammaticales. L'utilisation des approches d'analyses superficielles (voir chapitre 3) permet de pallier à certains de ces problèmes. Pour ces raisons, les approches stochastiques à grammaires syntaxiques superficielles et locales, notamment à base de classifieurs, sont en effet plus robustes aux erreurs syntaxiques et grammaticales des énoncés.

La contribution de cette thèse à la compréhension de la parole est l'utilisation de règles logiques et d'ontologies, afin de pouvoir créer des liens entre les composants sémantiques (connaissances segmentées Fikes et Nilsson (1971)) basiques trouvés par les différents systèmes de Compréhension du Langage Parlé . Cette approche vise à représenter des connaissances sémantiques contenues dans le signal, avec une représentation hiérarchisée, plus complexe, et moins plane que les constituants de bases obtenus par les systèmes de C.L.P. . Cette représentation s'appuie sur une ontologie du domaine liée à l'énoncé (voir appendice A), et qui va permettre au système de lier les concepts basiques en respectant la cohérence sémantique de l'énoncé. Cette représentation se base sur le paradigme des frames (Voir section 5.1), qui propose une représentation d'entités conceptuelles avec des attributs qui remplissent une frame au fur et à mesure qu'une valeur peut lui être ajoutée.

### 5.1 Connaissance fragmentée et composition sémantique

Les concepts exprimés dans une phrase sont souvent reliés sémantiquement. Les relations sémantiques, telles que les relations spatiales, sont indépendantes de la langue, tandis que les relations entre leurs constituants conceptuels et des mots dans une phrase varient selon la langue. Une approche d'entrelacement d'analyse syntaxique et d'interprétation est maintenant introduite. L'approche est basée sur la détection précoce des concepts exprimés par des relations syntaxiques locales telles que les phrases nominales et la composition successive des hypothèses sur les constituants conceptuels, représentés par des balises qu'on appellera "concepts", dans les structures sémantiques.

#### 5.1.1 Etiquettes "concepts" et connaissance fragmentée

Soit  $V$  le lexique de mots du système de R.A.P. . Soit  $C_1^K = C_1, \dots, C_k, \dots, C_K$ , la séquence d'étiquettes de concepts, appartenant au vocabulaire de concepts  $V_C, C_k \in V_C$ . Elle peut être hypothétisée à partir de la séquence de mots hypothèses générée par le

système de R.A.P. . Soit  $\sigma_k$  indiquant le *support* de  $C_k$ . Dans l'exemple 5.1.1 suivant les concepts associés aux mots, sont représentés sous la forme d'une paire attribut-valeur entre crochets. La séquence de mots est entre chevrons et représente un segment de phrase appelé *support*. La phrase est issue du corpus français MEDIA [Bonneau-Maynard et al. \(2005a\)](#).

Exemple (5.1.1)

"< je souhaite réserver > [command(reservation)] < un hotel > [object(hotel)] < avec parking > [hotel\_facility(parking)] < a Lyon > [city(Lyon)]".

Une frame décrit un prototype d'entité conceptuelle générale. Une instance de celle-ci décrit une réalisation particulière du prototype. Pour des raisons de simplicité, une frame sera désignée par son nom. Soit  $F_d$  une frame introduite pour décrire une connaissance sémantique d'un domaine d'application. La frame  $F_d$  a un nom et un ensemble de propriétés. La  $j^{\text{ième}}$  propriété est représentée par l'emplacement  $s_{d,j}$  (appelé aussi élément de frame ou "slot"). Cet emplacement peut être vide ou rempli par une valeur. Une instance de frame est caractérisée par l'association des valeurs des emplacements. Certaines valeurs peuvent être des instances d'autres frames. Des structures de ce type partiellement remplies sont appelées des fragments. Les fragments peuvent être composés dans des structures sémantiques plus complexes. L'interprétation sémantique d'une phrase ou d'un tour de dialogue est une structure sémantique représentée par un réseau d'instances de frames. L'instance d'une frame peut être décrite par une expression en logique du premier ordre.

**Définition 1** Soit  $G_{d,j}$  le nom de la propriété représentée par le  $j^{\text{ième}}$  emplacement  $s_{d,j}$  de la frame  $F_d$ . Soit  $\Gamma_{d,a}$  une instance de la frame  $F_d$ . Certaines valeurs de  $\Gamma_{d,a}$  peuvent être d'autres instances de frames ayant d'autres frames comme valeurs d'emplacement et ainsi de suite. Si une instance  $\Gamma_{q,z}$  de la frame  $F_q$  est une valeur de l'emplacement nommé  $G_{d,j}$  et  $s_{q,n}$  est un emplacement de  $\Gamma_{q,z}$ , alors la relation entre  $F_d$  et  $s_{q,n}$  est une chaîne d'emplacements indiquée comme suit :  $\delta_n = F_d \cdot G_{d,j} F_q \cdot s_{q,n}$ .

Un fragment d'instances est un ensemble de chaînes d'emplacements. Une chaîne d'emplacements peut être obtenue en enchaînant des emplacements de la manière suivante.  $\delta_z = F_k \cdot G_{k,l} \delta_x^\varepsilon F_q \cdot s_{q,z}$  où  $\delta_x^\varepsilon$  est une chaîne d'emplacements se terminant par un emplacement vide. Le nom de la première frame est appelé "tête" de la chaîne et le nom du dernier emplacement est la "queue" de la chaîne.

**Définition 2** Un fragment d'instance  $\Gamma_{i,a}$  est une structure sémantique constituée d'un nom de frame et d'une liste d'emplacements ("slot list") représentée par :  $\Gamma_{i,a} = F_i \cdot sl_{i,a}$

Une liste d'emplacements  $.sl_{i,a}$  est représentée comme suit, comme un ensemble de noms et de valeurs d'emplacements.  $.sl_{i,a} = [G_{i,1}v_{i,a,1}, \dots, G_{i,j}v_{i,a,j}, \dots, G_{i,J}v_{i,a,J}]$  où  $G_{i,j}$  est le nom d'un emplacement et  $v_{i,a,j}$  est sa valeur.

Chaque emplacement  $s_{i,j}$  est associé avec une facette  $\Phi_{i,j}$  qui est une liste de valeurs types. Les types de valeurs peuvent être des noms de frames dont les instances peuvent être les valeurs possibles des emplacements. Une facette est représentée comme suit :  $\Phi_{i,j} = \{\varphi_{i,k,m}\}$

$\varphi_{i,j,m} = F_q \vee V_q$

$F_q$  est le nom d'une frame, et  $V_q$  est un autre type de valeur.

La frame  $F_i$  est la tête du fragment. Tous les emplacements non remplis du fragment sont des éléments de la liste de la "queue de fragment".

Certaines valeurs d'emplacements peuvent être représentées par des variables dont les valeurs sont obtenues par un processus d'interprétation. Une valeur peut être un symbole représentant une procédure à exécuter en vue de donner une valeur à une variable.. Une procédure peut exécuter une requête à une base de données spécifiée par certaines valeurs d'autres emplacements de l'instance.

Chaque balise de concepts  $C_k$  caractérise une partie des connaissances et peut correspondre à un ou plusieurs fragments de l'ontologie de l'application, en fonction de certaines caractéristiques de l'expression de son support et son contexte. Une telle correspondance est établie par le concepteur de l'ontologie de l'application. Les balises de concepts de  $V_C$  font partie du cahier des charges pour l'annotation des corpus. Dans l'application `MEDIA`, certaines balises concepts génèrent des fragments représentant des actes de langage. Les supports de ces fragments sont des verbes et le rôle des verbes sont les valeurs de certains fragments de frame.

Dans l'exemple 5.1.2, un fragment correspondant à une étiquette de concept. Le support de ce fragment est le support de l'hypothèse du concept correspondant.

Exemple (5.1.2)

Concept  $C_k$  : `hotel_facility(parking)`

Fragment  $\Gamma_k$  : `HOTEL.[h_facility FACILITY.[facility_type parking]]`

Le fragment  $\Gamma_k$  est une représentation linéaire d'une structure en frame dans laquelle les noms de frames sont en lettres capitales, la liste des emplacements d'une frame est représentée par des crochets, ses éléments sont séparés par une virgule et la valeur d'un emplacement suit le nom de l'emplacement après un espace. Lorsqu'une valeur est une constante et non à une instance de frame, elle est représentée en italique. La relation entre un nom de frame et de sa liste d'emplacements est représentée par un point.

Les fragments d'instances sont hypothétisés à partir de la séquence  $C_1^K$  grâce à une table d'équivalence. Quand plusieurs choix sont possibles pour un concept  $C_k$ , un ensemble fini d'instances de fragments  $\mathfrak{S}_k$  lui est associé et la désambiguïsation est basée sur les propriétés du support  $\sigma_k$  et son contexte.

Par souci de comparaison avec la grammaire catégorielle,  $C_k$  peut être considérée comme l'étiquette pour une composition réalisée avec une inférence dépendante du contexte en utilisant ce type de connaissance :

## Exemple (5.1.3)

Dans la phrase : “un hôtel avec parking”

“un hôtel” := NP :  $\lambda(x).is\_a(x, HOTEL)$

“parking” := NP :  $is\_value\_of(parking, facility\_type)$

Ces composants agissent à titre de contexte dont le sens est composée avec celui de la préposition qui suit :

“avec” := (S/NP)/NP :

$$\lambda(x).is\_a(x, HOTEL) \quad \wedge \quad HOTEL.h\_facility(x, \lambda(y).is\_a(y, FACILITY) \quad \wedge \quad FACILITY.facility\_type(y, parking))$$

La lambda-expression associée avec le mot “avec” est obtenue par inférence en utilisant la connaissance de l’ontologie du domaine comme suit :

$$\lambda(x).is\_a(x, HOTEL) \quad \wedge \quad is\_property\_of(h\_facility, HOTEL) \quad \wedge \quad contains\_facet\_of(h\_facility, FACILITY) \quad \wedge \quad FACILITY.facility\_type(y, parking)) \Rightarrow HOTEL.h\_facility(x, \lambda(y).is\_a(y, FACILITY) \quad \wedge \quad FACILITY.facility\_type(y, parking))$$

Dans la pratique, au lieu d’analyser complètement une phrase, une analyse de surface peut extraire des structures syntaxiques simples, comme les NP. Des procédures qui peuvent contenir des lambda-expressions, effectuent des compositions sémantiques fondées sur une inférence sur des connaissance parcellaires, le support et la signification des mots. La connaissance associée à des mots peut être automatiquement apprise et les compositions multiples possibles peuvent être corrigées à l’aide de modèles exponentiels tel que décrit en Zettlemoyer et Collins (2009). Des inférences utiles peuvent être effectuées en utilisant l’ontologie de domaine et la définition de facettes frames-emplacements.

Les étiquettes de concept peuvent être associées avec des segments de phrases, avec l’annotation manuelle comme dans MEDIA . Les fragments correspondants peuvent également être dérivés manuellement. Les méthodes utilisées pour obtenir des hypothèses d’étiquettes de concepts issues de différentes analyses de surface ou partielles ont été comparées dans le corpus MEDIA et sont décrites dans le rapport final du projet LUNA <sup>1</sup>. Elles seront résumées dans la section 4.2.4.

Une instance de frame représente une relation sémantique n-aire entre un nom de frame, ses emplacements et ses valeurs. Un fragment d’instance, et en général une instance d’une structure sémantique, représentent une relation complexe n-aire entre les noms de frame, leurs emplacements et leurs valeurs.

Les structures sémantiques instituant des relations n-aires entre des constituants sémantiques peuvent être composées en utilisant des relations binaires. Les supports de ces relations sont représentés par des modèles contenant les caractéristiques des constituants de support et leurs liens. Les liens sont souvent représentés par des relations syntaxiques locales qui relient, par exemple, un syntagme nominal exprimant une étiquette avec un autre syntagme nominal exprimé par une autre balise. La connexion peut être représentée, par exemple, par une préposition. Des explications détaillées peuvent être trouvées dans Mori (2011).

1. disponible sur <http://www.ist-luna.eu>

## 5.1.2 Composition par fusion de fragments

La composition par fusion permet, sous certaines conditions, à deux instances de frames portant des têtes identiques de les rassembler en une seule instance de frames.

Considérons le fragment d'instance suivant :

$\Gamma_{i,d} = F_i.sl_{i,d} = F_i. [G_{i,1}v_{i,d,1}, \dots, G_{i,j}F_q.sl_{q,d}, \dots, G_{i,J}v_{i,d,J}]$  dans lequel la  $j^{ième}$  valeur de la liste d'emplacements  $sl_{i,d}$  est un autre fragment :  $v_{i,d,j} = .F_q.sl_{q,d}$  Supposons que ce fragment a la même "tête" qu'un autre fragment hypothétisé  $\Gamma_{q,e} = F_q.sl_{q,e}$ ,  $head(v_{i,d,j}) = head(\Gamma_{q,e})$  Une composition possible peut être produite en fusionnant  $v_{i,d,j}$  avec  $\Gamma_{q,e}$  attendu, que certaines conditions, introduites ci-après, soient vérifiées. Soient les listes d'emplacements  $v_{i,d,j} = .F_q.sl_{q,d}$  et  $\Gamma_{q,e} = F_q.sl_{q,e}$  respectivement :

$sl_{q,d} = G_{q,1}v_{q,d,1}, \dots, G_{q,j}v_{q,d,j}, \dots, G_{q,J}v_{q,d,J}$   
et

$sl_{q,e} = G_{q,1}v_{q,e,1}, \dots, G_{q,j}v_{q,e,j}, \dots, G_{q,J}v_{q,e,J}$  La fusion de  $v_{i,d,j}$  et  $\Gamma_{q,e}$  est possible s'il y a un support pour ceux-ci. La fusion est confirmée par la vérification du prédicat :

$supp(fusion(F_q.sl_{q,d}, F_q.sl_{q,e}))$  et seulement si :  $\forall(j)((v_{q,d,j} = \varepsilon) \vee (v_{q,e,j} = \varepsilon))$  où le symbole  $\varepsilon$  représente le fait que l'emplacement correspondant n'est pas encore rempli. Une condition plus complexe sur la compatibilité des deux valeurs peut aussi être considérée. L'action de composition est réalisée comme une conséquence de l'assertion du prédicat  $fusion(F_q.sl_{q,d}, F_q.sl_{q,e})$ , faite par la règle d'inférence suivante :

$[head(v_{i,d,j}) = head(\Gamma_{q,e})] \wedge supp[fusion(F_q.sl_{q,d}, F_q.sl_{q,e})] \wedge [\forall(j)((v_{q,d,j} = \varepsilon) \vee (v_{q,e,j} = \varepsilon))] \Rightarrow fusion(F_q.sl_{q,d}, F_q.sl_{q,e})$ . Le prédicat  $supp(x)$  est vrai quand un motif  $\pi_k(x)$ , associé à celui-ci, correspond aux hypothèses ou données disponibles. Les différentes valeurs de  $k$  correspondent aux différents motifs qui doivent s'assortir avec les données pour l'assertion du support. Il peut y avoir de multiples fusions possibles, dont certaines peuvent ne pas avoir de support dans les données. Les conflits possibles dans le processus de fusion sont résolus en assignant des priorités aux fusions candidates, en se basant sur les probabilités des fusions produites par les données.

### Exemple (5.1.4)

Soit

- La phrase : "Un hôtel quatre étoiles avec parking"

- Fragments :

-  $HOTEL.[luxury\ 4\_etoiles]$

-  $HOTEL.[h\_facilityFACILITY.[facility\_type\ parking]]$

- Tête de frames et valeurs de fragments :

-  $F_q : HOTEL$

-  $s_{q,d} : [luxury\ 4\_etoiles]$

-  $s_{q,e} : [h\_facilityFACILITY.[facility\_type\ parking]]$

- Supports :

-  $\sigma_1$  : hôtel quatre étoiles

-  $\sigma_2$  : parking

-  $supp(fusion(F_q.sl_{q,d}, F_q.sl_{q,e})) = correspondance[< \sigma_1 "avec" \sigma_2 >, data]$ , où  $< \sigma_1 "avec" \sigma_2$  est un modèle associé à un fragment de fusion impliquant des propriétés d'un hôtel.

- Résultat de composition :  $HOTEL.[luxury\ 4\_etoiles, h\_facility\ FACILITY.[facility\_type\ parking]]$

### 5.1.3 Composition par attachements

La composition par attachements permet de positionner une tête de frame adéquate, en tant que valeur d'un attribut pour une autre frame.

Etant donné un fragment d'instance  $\Gamma_{i,a}$ , considérons l'ensemble  $\Delta_i$  de toutes les chaînes d'emplacements, avec  $F_i$  comme "tête" de frame et se terminant par des emplacements non remplis. Soit  $\delta_e \in \Gamma_i$  une de ces chaînes.

Supposons que deux instances  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$  ont été hypothétisées. Soit  $head_h$  le nom de la frame qui est la tête de  $\Gamma_{i,a}$ . Soit la queue de  $\delta_z$ , un emplacement non rempli, ayant pour nom  $B_{q,z}$  et pour facette  $\Phi_{q,z}$ . Soit  $\Phi_{q,z}$  comportant un nom de frame qui est égal à  $head_h$ . Une relation sémantique binaire peut être créée entre l'emplacement  $B_{q,z}$  de  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$ , si il existe un support dans les données d'où les instances de  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$  ont été hypothétisées. La présence de ce support est exprimée par la véracité du prédicat  $supp[link(B_{q,z}, \Gamma_{h,b})]$ . La relation sémantique établit un lien de composition qui assigne  $\Gamma_{h,b}$  comme une valeur de  $B_{q,z}$ . L'action de composition est exécutée par l'assertion du prédicat  $link(B_{q,z}, \Gamma_{h,b})$  avec la règle de d'inférence suivante :  $contains(\Phi_{q,z}, head_h) \wedge supp[link(B_{q,z}, \Gamma_{h,b})] \Rightarrow link(B_{q,z}, \Gamma_{h,b})$

Le prédicat  $contains(\Phi_{q,z}, head_h)$  est vrai quand la facette  $\Phi_{q,z}$  contient  $head_h$ .

Dans le corpus MEDIA, les entités sémantiques appelées "spécifieurs" ont été utilisées pour supporter les liens de certaines compositions et pour effectuer une sélection quand une étiquette de concept correspond à plus d'un fragment.

#### Exemple (5.1.5)

Soit la phrase : "Un hôtel quatre étoiles avec parking à Lyon".

- Les fragments sont :
  - $\Gamma_{i,a} : HOTEL.[luxury\ quatre\_etoiles, h\_facility FACILITY.[facility\_type\ parking]]$
  - $\Gamma_{h,b} : ADDRESS.[adr\_city\ Lyon]$
- la chaîne d'emplacements de  $\Gamma_{i,a}$  :
  - $\delta_z : HOTEL.at\_loc$
  - $tail(\delta_z) : B_{q,z} = at\_loc$
  - $\Phi_{q,z} = LOCATION, ADDRESS$
- $head_h = ADDRESS; head_h \in \Phi_{q,z}$
- les supports de fragments :  $\sigma_1 : [HOTEL], \sigma_2 : [ADDRESS]$
- $supp[link(B_{q,z}, \Gamma_{h,b})] = match\{< \sigma[HOTEL] - - \sigma[ADDRESS] >, data\}$

Le support de la relation est généralisé en considérant  $\sigma[HOTEL]$  comme une abstraction de toutes les séquences de mots exprimant un fragment comportant HOTEL comme tête et  $\sigma[ADDRESS]$  comme abstraction de tous les supports pour ADDRESS.

Résultats de la composition :  $HOTEL.[luxury, quatre\_etoiles, h\_facility FACILITY.[facility\_type\ parking], at\_loc ADDRESS.[adr\_city\ Lyon]]$

La désambiguïsation est exécutée lorsque nécessaire en utilisant les probabilités postérieures pour les supports engendrés par les données.



### 5.1.4 Composition par attachements et inférences

La composition par inférence permet, en cohérence avec l'ontologie, de créer une frame puis de la lier à deux frames existantes.

Soit  $\Gamma_{i,a}$  un fragment d'instance et  $\Gamma_{h,b}$  un autre fragment d'instance qui ne peut pas être composé avec  $\Gamma_{i,a}$  par fusion ou attachements. Soit  $\delta_z = F_i.G_{i,l} \delta_x F_q.B_{q,z}$ , une chaîne d'emplacements de  $\Gamma_{i,a}$  et  $\delta_v = F_c.\delta_x F_u.S_{u,v}$  une chaîne d'emplacements consistante avec l'ontologie d'application telle que :

$$\text{contains}[\Phi_{q,z}, \text{head}(\delta_v)] \wedge \text{contains}[\Phi_{u,v}, \text{head}_h] \text{ et } \Phi_{u,v} = \Phi(\text{tail}(\delta_v)).$$

Une règle d'inférence établit un lien représenté par la véracité du prédicat  $\text{link}(B_{q,z}, \delta_v.\Gamma_{h,b})$ , si un support convenable existe. Ceci est représenté par la véracité du prédicat  $\text{supp}[\text{link}(B_{q,z}, \delta_v.\Gamma_{h,b})]$ .

La règle d'inférence est :  $\text{contains}[\Phi_{q,z}, \text{head}(\delta_v)] \wedge \text{contains}[\Phi_{u,v}, \text{head}_h] \wedge \text{supp}[\text{link}(B_{q,z}, \delta_v.\Gamma_{h,b})] \Rightarrow \text{link}(B_{q,z}, \delta_v.\Gamma_{h,b})$

Une règle d'inférence similaire existe pour exécuter l'inférence d'une chaîne d'emplacements avant d'inférer une composition par fusion.

#### Exemple (5.1.6)

Soit la phrase : "J'appelle pour réserver dans un hôtel quatre étoiles avec parking à Lyon" :

- $\Gamma_{i,a} = \delta_z : \text{REQUEST}.[\text{request\_theme RESERVATION.reserv\_obj}]$
- $B_{q,z} : \text{reserv\_obj}$
- $\Phi_{q,z} = \text{ROOM}, \dots$
- $\Gamma_{h,b} : \text{HOTEL}.[\text{luxury quatre étoiles}, h\_facility \text{ FACILITY}.[\text{facility\_type parking}], \text{at\_loc ADDRESS}.[\text{adr\_city Lyon}]]$
- $\delta_v : \text{ROOM.establishment}$
- $\text{tail}(\delta_v) : \text{establishment}$
- $\Phi_{u,v} = \Phi(\text{tail}(\delta_v)) = \text{HOTEL}, \dots$
- $\text{supp}[\text{link}(B_{q,z}, \delta_v.\Gamma_{h,b})] = \text{match} < \sigma[\text{RESERVATION}] \text{ "dans"} \sigma[\text{HOTEL}] >, \text{data}$

Le résultat de la composition est :

$\text{REQUEST}.[\text{request\_theme RESERVATION}.[\text{reserv\_obj ROOM}.[\text{establishment HOTEL}.[\text{luxury quatre étoiles}, h\_facility \text{ FACILITY}.[\text{facility\_type parking}], \text{at\_loc ADDRESS}.[\text{adr\_city Lyon}]]]]]$

Différentes méthodes ont été considérées et comparées dans le projet LUNA pour exécuter la composition par fusion et par attachement. Elles utilisent des règles obtenues par apprentissage par explication ("Explanation Based Learning" EBL) avec les supports caractérisés par les relations syntaxiques locales, par modèles génératifs basés sur les réseaux bayésiens dynamiques ( D.B.N. ) et la classification basée sur les champs conditionnels aléatoires ( C.R.F. : Conditional Random Fields). Ces méthodes et résultats sont décrits dans ?.

### 5.1.5 Probabilités de frames

Un calcul possible des probabilités de frames consiste à considérer explicitement une séquence  $C$  d'hypothèses de constituants sémantiques :

$$\begin{aligned} P(\Gamma|Y) &= P(\Gamma|CW)P(C|W)P(W|Y) \\ P(\Gamma|CW) &= P(\Gamma|\Phi(CW)) \end{aligned} \quad (5.1)$$

où  $\Phi(CW)$  est un ensemble de caractéristiques exprimant le support des relations entre les supports des remplisseurs d'emplacements.

La probabilité  $P(\Gamma|\Phi(CW))$  peut être calculée par des modèles exponentiels [C.R.F.](#), avec des fonctions dans les exposants évaluant la correspondance de motifs  $\pi_q$  avec des caractéristiques extraites de  $CW$ .



## Chapitre 6

# Contribution à la compréhension automatique de la parole

### Sommaire

---

<b>6.1</b>	<b>Composition par règles</b>	<b>116</b>
6.1.1	Le corpus MEDIA et la génération des constituants de base	117
6.1.2	Composer des relations sémantiques dans les structures	118
6.1.3	Annotation progressive du corpus par des structures sémantiques	119
6.1.4	Résultats préliminaires	121
6.1.5	L'outil d'annotation et de composition	122
6.1.6	Utilisation de fragments de frames et inférences	125
<b>6.2</b>	<b>Acquisition progressive des caractéristiques syntaxiques supportant les relations sémantiques</b>	<b>127</b>
6.2.1	Annotation des relations binaires	129
6.2.2	Processus de composition sémantique	131
<b>6.3</b>	<b>Expérience de composition sémantique</b>	<b>133</b>
6.3.1	Résultats	135
6.3.2	Désambiguïsation sémantique	137
6.3.3	Exemple de désambiguïsation d'un tour de dialogue	138
<b>6.4</b>	<b>Détection de lambda-expressions</b>	<b>141</b>
6.4.1	Calcul des probabilités	147
6.4.2	Les fonctions caractéristiques	148
6.4.3	Expérimentations	149

---

Les travaux de compréhension de la parole de cette thèse s'articulent autour de la composition automatique de frames par des règles, respectant la cohérence du discours à l'aide d'une ontologie (voir appendice A). Cette approche, décrite dans la section 6.1, utilise des règles logiques basée sur l'approche décrite dans le chapitre 5 qui s'appliquent à des instances de frames dont les prototypes sont définis par une ontologie. Par la suite cette méthode a été utilisée en combinaison avec des C.R.F. dont les paramètres ont été estimés par apprentissage automatique permettant ainsi d'associer une probabilité aux hypothèses émises. Notre étude a été menée sur la découverte des supports syntaxiques des relations sémantiques dans la section 6.2. Nous avons considéré l'utilisation de la composition par règles et des inférences pour la désambiguïsation de concepts, c'est-à-dire l'ajout de spécificateurs dans les concepts, nous avons combiné l'approche décrite dans 6.1 pour émettre des hypothèses sur les possibles spécificateurs à utiliser, cette approche est décrite dans la section 6.3. Nous avons établi en section 6.4 un ensemble de motifs pour la détection locale et syntaxique de références en utilisant des C.R.F. pour découvrir un ensemble de fonctions logiques liées au discours.

### 6.1 Composition par règles

La compréhension de la parole nécessite certains outils notamment la sémantique. Elle est la description et l'organisation des signaux et de leurs relations. La compréhension du langage parlé est l'interprétation de ces signes reçus par un signal vocal. Les relations sont représentées par une Source de Connaissance (SC) et appliquées par des processus utilisant des stratégies de contrôle de connaissances. Cette tâche est ardue car mélangée parmi d'autres informations comme du bruit dans l'environnement.

Les phrases issues du langage naturel sont difficiles à traiter et souvent sans forme grammaticale formelle. Les hypothèses de mots des transcriptions ont des erreurs induisant ainsi une connaissance imparfaite.

Dans le but de minimiser les effets de ces imprécisions, l'interprétation doit être conçue comme un processus de décision qui peut être décomposé conceptuellement en sous-tâches.

On propose ici d'utiliser la méthode décrite dans le chapitre 5 pour la combinaison de constituants dans des structures sémantiques ainsi qu'un processus d'évaluation. Les constituants sont générés par un processus de traduction à partir d'un treillis de mots. Les correspondances entre constituants sémantiques et leur expression en langue naturelle appelé *support*, sont établies par des motifs, Raymond et al. (2006). Quand les motifs correspondent à certaines caractéristiques basées sur les constituants et les mots en hypothèses, on exécute des procédures de construction de structures. Cette approche a été testée sur un corpus français, MEDIA, disponible par l'agence ELDA. Les résultats expérimentaux issus de cette approche sont évalués avec un protocole décrit dans l'appendice B.

### 6.1.1 Le corpus MEDIA et la génération des constituants de base

#### Description du corpus

Le corpus MEDIA [Bonneau-Maynard et al. \(2005b\)](#) a été enregistré en utilisant un système de simulation (*Magicien d'Oz*) de serveur vocal téléphonique pour des informations touristiques et réservations d'hôtels. Huit catégories de scénario ont été définies avec des degrés de complexité différents. Le corpus compte 1257 dialogues, 250 interlocuteurs et contient environ 70 heures de dialogues. La portion d'entraînement du corpus est riche de plus de 80 concepts de bases transcrits et annotés manuellement.

Cette représentation sémantique *plate* est enrichie par des étiquettes qui peuvent être considérées comme des traces de la représentation hiérarchique sous-jacente. La représentation hiérarchique permet de représenter explicitement des relations entre segments potentiellement non-adjacents dans la transcription d'une requête. Cependant une représentation *plate* facilite l'annotation manuelle des données. Il a été décidé pour le schéma d'annotation de MEDIA de préserver les relations en définissant des spécifieurs qui sont combinés avec les rôles de bases. Il y a 19 spécifieurs dans le modèle sémantique de MEDIA .

Un exemple de l'annotation de MEDIA est donné dans le tableau 6.1. Le spécifieur *réservation* est attribué aux concepts *commande, nombre-chambre, nombre-nuit* comme structure hiérarchique. Cela représente une réservation déclenchée par le concept *commande* et remplie avec les éléments trouvés dans *commande, nombre-chambre* et *nombre-nuit*.

TABLE 6.1 – Exemple d'annotation sémantique du corpus MEDIA

$n$	$W^{c_n}$	$c_n$	specifieur	valeur
1	<i>eh bien</i>	null		
2	<i>je souhaiterais réserver</i>	commande		réservation
3	<i>à l'hôtel Richard Lenoir</i>	nom-hotel		richard_lenoir
4	<i>six</i>	nombre-chambre	réservation	6
5	<i>chambres individuelles</i>	type-chambre		simple
6	<i>à partir du trente-et-un mai</i>	date	réservation	31/05
7	<i>deux jours hum deux nuits</i>	nombre-nuit	réservation	2

La combinaison des spécifieurs et des attributs permet de recomposer une structure hiérarchique à partir de sa représentation *plate*. Cette annotation fournit des étiquettes comparables aux constituants sémantiques hypothésés par un parseur sémantique superficiel. La combinaison des rôles de base et des spécifieurs donne 1121 attributs potentiels. Un total de 144 attributs distincts apparaît dans le corpus d'entraînement avec environ 2200 valeurs normalisées différentes.

L'annotation du corpus MEDIA a été enrichie avec une annotation en structure de frames par un processus semi-automatique dont tous les résultats ont été validés ou corrigés par un expert.

### Décodage conceptuel pour la génération de constituants basiques

Le processus de décodage conceptuel est vu comme un processus de transduction dans lequel chaque concept est représenté par modèle de langage stochastique implémenté par un automate à états finis ( **F.S.M.** ). Ces transducteurs retournent des hypothèses de composants sémantiques. Il y a un **F.S.M.** pour chaque composant conceptuel élémentaire. Ils sont appliqués à un graphe de mots sorti par le décodeur de la reconnaissance automatique de la parole ( **R.A.P.** ) grâce à une opération de composition d'automates.

Pour trouver la meilleure séquence de concepts et de mots un **H.M.M.** est utilisé pour recalculer chaque chemin dans le graphe de mots/concepts. Ce **H.M.M.** est entraîné sur le corpus d'entraînement de **MEDIA** . Cette approche est appelée approche de décodage *intégrée*, puisque le module de **R.A.P.** et le processus de compréhension du langage sont réalisés en même temps en recherchant de manière simultanée la meilleure séquence de mots et de concepts. Le résultat du processus de traduction est une liste structurée des  $n$ -meilleures séquences d'hypothèses de concepts qui peuvent être vues comme les interprétations possibles d'un énoncé.

### Ajout des spécifieurs aux séquences de concepts

Les interprétations conceptuelles issues de la liste des  $n$ -meilleures n'ont pas de spécifieurs. Ces spécifieurs sont ajoutés dans une seconde phase par un processus de marquage basé sur des classifieurs discriminants **Duvert et al. (2008)**. Les **C.R.F.** retenus dans notre étude, ont été largement utilisés pour différents traitements d'étiquetage de mots comme les *Part-Of-Speech* ou pour la détection des entités nommées. L'utilisation des **C.R.F.** est une approche discriminante qui a donné de meilleurs résultats sur ce genre de processus que des approches génératives basées sur les **H.M.M.** . Le principal avantage des **C.R.F.** est la capacité de prédire des hypothèses, en tenant compte de l'ensemble de éléments relatifs au message complet.

L'entraînement pour l'étiquetage des spécifieurs se fait sur le corpus **MEDIA** , chaque message est une séquence d'éléments (mots, attributs, valeurs) étiquetée par un spécifieur ou par le symbole **NULL**. Lors du décodage, chaque séquence hypothèse de mots/concepts de la liste de  $n$ -meilleures séquences est traitée par l'étiqueteur pour ajouter les spécifieurs. L'outil **CRF++**<sup>1</sup> est utilisé. Les **C.R.F.** prennent en compte les dépendances sur de longues distances qui supportent les composants des structures sémantiques sans utiliser des règles pré-établies d'analyse.

#### 6.1.2 Composer des relations sémantiques dans les structures

Les structures sémantiques peuvent être dérivées à partir d'une source de connaissances obtenue par une théorie sémantique. Ces structures sont représentées par des

---

1. <http://crfpp.sourceforge.net/>

structures de données telles les frames. Une frame est un modèle pour la représentation sémantique d'entités, leurs propriétés et leurs relations. Les frames sont capables de représenter des types de structures conceptuelles aussi bien que des instances de celles-ci.

Un élément de frame est un composant de la frame qui décrit :

- les propriétés d'une structure sémantique,
- les valeurs, peut être une instance d'une autre frame,
- les contraintes qui doivent être respectées par les valeurs,
- les procédures pour obtenir les valeurs des propriétés à partir du signal

Le remplissage des Elements d'une frame par des valeurs engendre l'instanciation de cette frame. Une grammaire de frame permet de caractériser l'ensemble des frames acceptables pour la représentation sémantique d'un domaine.

### 6.1.3 Annotation progressive du corpus par des structures sémantiques

Une ontologie ou source de connaissances basée sur les frames a été préparée manuellement pour décrire la connaissance sémantique du domaine de *MEDIA*. Quelques frames décrivent des connaissances génériques comme des relations spatiales, d'autres sont spécifiques à l'application. Ces frames sont définies en tenant compte du paradigme des *FrameNet* de Berkeley adopté par *LUNA* (2009).

L'ontologie *MEDIA* est composée de 51 frames de base avec un total de 201 éléments de frames. La représentation du langage contient des composants conceptuels et des procédures de compositions de structures sémantiques.

Quand un motif correspond aux données d'entrée (mots ou concepts), des instances de frames sont créées. Des inférences basées sur les instances de frames sont alors réalisées. Différentes frames reliées par des relations peuvent être instanciées par un simple motif. Un outil de visualisation de frames appelé *FriZ*, dédié au traitement du langage dans le dialogue a été développé comme support de l'annotation manuelle et de la vérification pour d'ultérieures annotations automatiques. Par exemple la phrase "*J'accepte la réservation*" est annotée avec trois frames :

```
ACCEPTER [ (est_un:verbe) (sujet:person) (theme:reservation) ]
PERSONNE [ (est_un:humain) (categorie:utilisateur) ... ]
RESERVATION [ (est_un:objet) ... ]
```

Les motifs sont généralisés par l'annotation progressive des données avec les connaissances disponibles. On évalue la confiance des résultats et on annote des exemples manuellement si la confiance est trop faible.

Les procédures attachées sont intégrées dans le processus d'interprétation pour fournir automatiquement des annotations de frames sur le corpus d'entraînement ainsi que des instances d'hypothèses avec le corpus de test. Le processus est capable de réaliser des inférences sur des frames dont l'instance est sous-entendue par une autre instance de frame. Une centaine de règles génère les instances à partir de combinaisons de mots ou de motifs de constituants sémantiques et réalise les inférences sur les résultats. Il y a environ 30 formules d'inférences utilisées dans le processus.



Lors du décodage une fois la liste des  $n$ -meilleures interprétations obtenue, avec des spécifieurs présentés en Section 6.1.1, chaque séquence de mots/concepts est analysée par les règles produites pour le corpus d’entraînement de MEDIA . Ces règles utilisent les attributs, les valeurs et les spécifieurs obtenus lors de la première phase du décodage pour inférer des frames.

Les figures 6.1, 6.2 et 6.3 montrent respectivement les étapes d’annotation en concepts du tour “Vous pouvez me confirmer tous les éléments, ils ont bien une baignoire un téléphone et c’est à quatre-vingt-dix euros à côté de Paris”, de mises en hypothèses des fragments de frames à partir des concepts et la composition des fragments de frames.

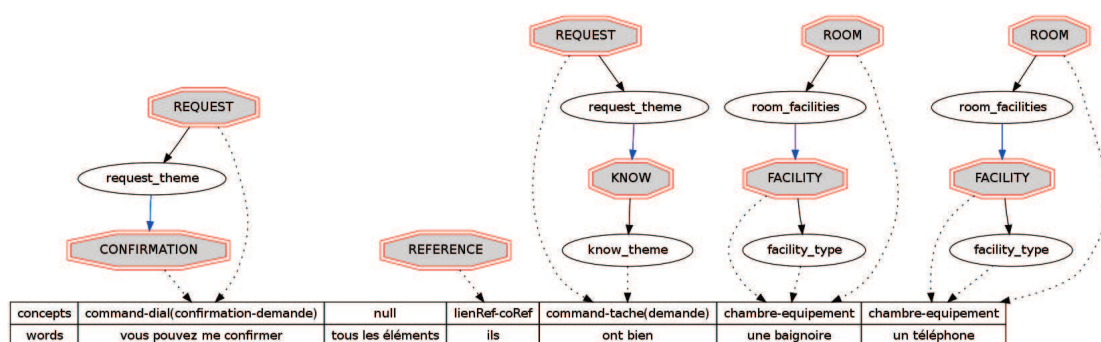


FIGURE 6.1 – Etape de mise en hypothèses des fragments de frames à partir des concepts - 1<sup>er</sup> partie de phrase

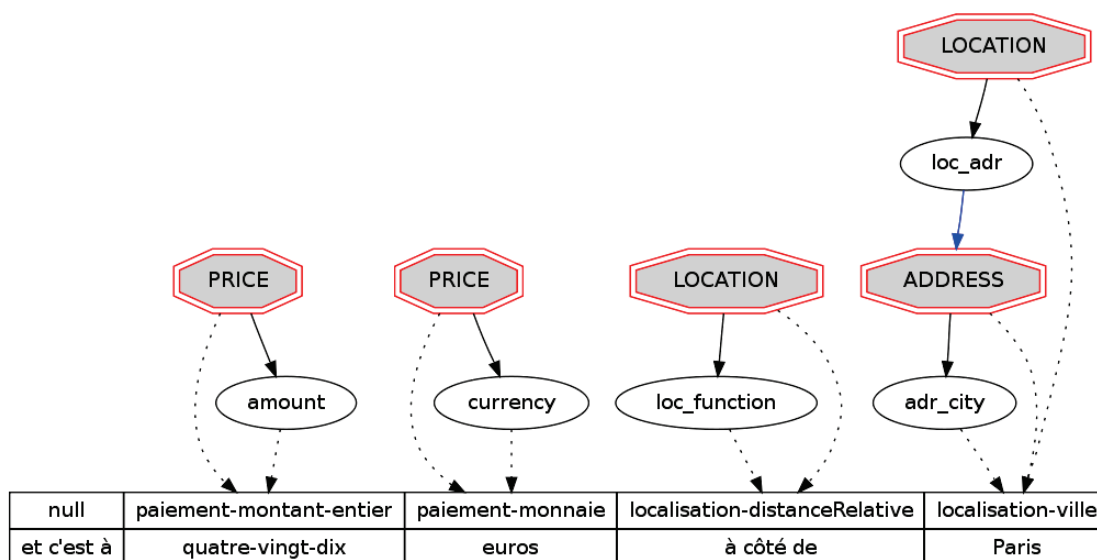


FIGURE 6.2 – Etape de mise en hypothèses des fragments de frames à partir des concepts - 2<sup>ième</sup> partie de phrase

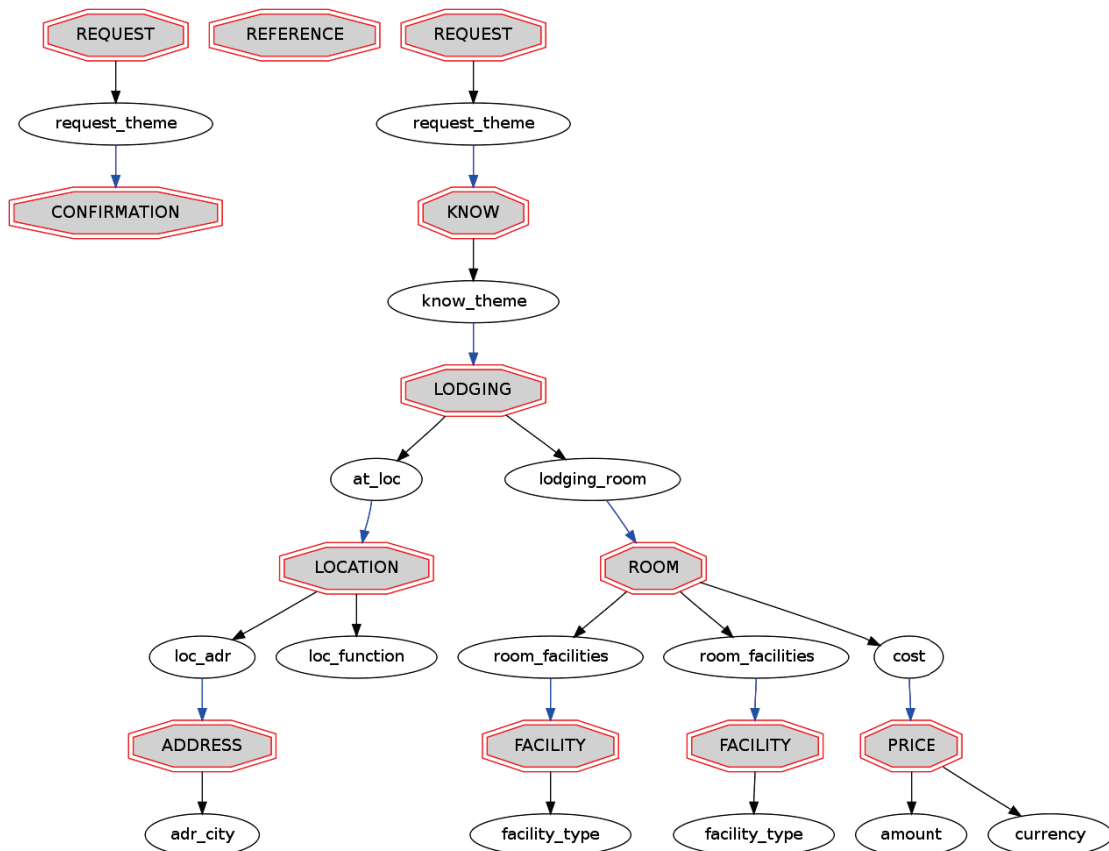


FIGURE 6.3 – Etape de mise en composition des hypothèses de fragments de frames

### 6.1.4 Résultats préliminaires

Un premier test a été réalisé sur un corpus de 1249 tours de dialogues pour un total de 2938 composants sémantiques. Le tableau 6.2 donne le taux d’erreurs obtenu après la phase de décodage conceptuel. Pour un taux d’erreurs “mot” de 30.3%, le taux d’erreurs “attribut” est d’environ 25%. Chaque information, telle que les spécifieurs et les valeurs normalisées, ajoute approximativement 6% aux taux d’erreurs. Les taux d’erreurs “Oracle” obtenus en sélectionnant manuellement les meilleures hypothèses issues de la liste des  $n$ -meilleures interprétations (avec  $n = 20$ ) sont inférieurs de 8% en valeur absolue par rapport aux taux d’erreurs de la 1-meilleure interprétation.

TABLE 6.2 – Taux d’erreurs (ER) et Taux d’erreurs Oracle (OER) sur la  $n$ -meilleure liste des interprétations pour les mots, les concepts, les concepts avec des spécifieurs et valeurs

	corr(%)	sub(%)	suppr(%)	ins(%)	ER(%)	OER(%)
mot	75.9	15.3	8.8	6.2	30.3	22.5
concept	85.0	8.7	6.3	10.3	25.3	19.2
+ specif	78.6	15.2	6.2	10.2	31.6	23.4
+ value	72.5	21.4	6.1	10.1	37.6	25.2

Les hypothèses de frames obtenues sur la sortie du processus d'interprétation ont aussi été évaluées. Les annotations manuelles en frames n'étaient pas disponibles pour le corpus de test. Les annotations manuelles des mots et des concepts ont été utilisées pour dériver des annotations références en frames. La composition et l'inférence des connaissances, décrites dans la sous section précédente, ont été appliquées.

Un échantillonnage aléatoire sur des tours de dialogues utilisateurs a été prélevé pour estimer, par deux experts humains, l'exactitude de l'annotation automatique des structures sémantiques. Une F-mesure de 0.90 (précision de 0.96, rappel de 0.85) a été mesurée sur 100 tours de dialogues, par comparaison des annotations manuelles avec l'annotation automatique en frame sur des transcriptions exactes. Cette forte précision permet d'utiliser une annotation automatique comme annotation de référence.

La composition et l'inférence ont été appliquées automatiquement sur la liste des  $n$ -meilleures interprétations. L'évaluation a été faite en estimant la précision, le rappel et la F-mesure sur la détection des types corrects de frames en utilisant les annotations automatiques de références décrites plus haut. La F-mesure Oracle de la liste des  $n$ -meilleures interprétations est donnée par la figure 6.4.

Une F-mesure de 0.92 (précision de 0.90, rappel de 0.94) a été obtenue sur la meilleure hypothèse d'interprétation, pour les 1249 tours de dialogues. Ces résultats tendent à montrer que le plus haut niveau d'annotation sémantique (identité des frames) est robuste aux erreurs de reconnaissance automatique du langage, les erreurs d'interprétations apparaissant le plus souvent au niveau des éléments des frames.

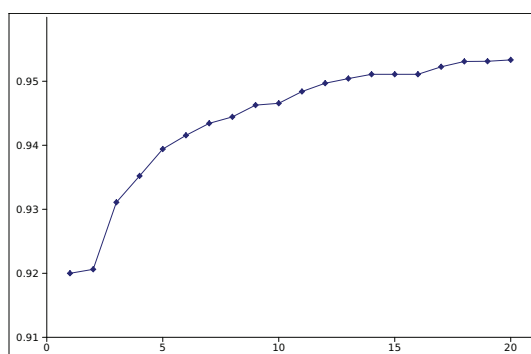


FIGURE 6.4 – F-mesure Oracle pour l'identification des frames instanciées sur les premières  $n$ -meilleures séquences d'éléments conceptuels extraits à partir d'un treillis de décodage, en fonction de  $n$

### 6.1.5 L'outil d'annotation et de composition

Un premier outil a été développé pour représenter les tours de dialogue, sous forme de structure sémantique à base de frames. Il prend en entrée le tour de dialogue. Il peut être soit tour de référence, il est alors transcrit manuellement et annoté en concepts et spécifieurs manuellement. Ou alors c'est un tour de transcription automatique, issu

de la *R.A.P.* . L'outil décompose les informations contenues dans le tour en plusieurs ensembles :

- $W_n$  : ensemble des mots issus de la transcription,
- $C_{W_n}$  : ensemble des concepts liés aux mots. Ces concepts contiennent un certain nombre d'informations supplémentaires, comme le spécifieur, et la valeur du concept.

Les règles logiques de premier ordre sont introduites dans le système. Elles sont sous la forme d'implication logique où un ensemble de prédicats ou préconditions doit être vérifié pour générer les valeurs introduites dans la conséquence de la formule logique. Ainsi dans la formule logique 6.1, si « chambre-type » et « nombre-chambre » sont présents, alors on génère le fragment de frame de la partie conséquence de la formule. Les fragments de frames ainsi créés servent à l'annotation en frames.

$$\left\{ \begin{array}{l} \text{chambre} - \text{type} \\ \text{nombre} - \text{chambre} \end{array} \wedge \right\} \rightarrow \left\{ \begin{array}{l} \text{ROOM.} [ \\ \text{lodg\_qty.} \\ \text{NUMBER.} [\text{type.cardinal}], \\ \text{room\_type.} \\ \text{OCCUPANCY.occ\_type} ] \end{array} \right\} \quad (6.1)$$

Les règles peuvent s'appliquer sur les mots, les concepts, les spécifieurs, les frames et les emplacements de frames. L'exemple de la formule logique 6.2 montre une règle vérifiant la présence de frames hypothétisées afin de pouvoir les combiner. Dans cet exemple, pour créer un lien sémantique entre RESERVATION et NUMBER, il faut que ces deux frames soient présentes et on doit vérifier également la non présence des frames PRICE et ROOM, puisque le nombre aurait pu être affecté à un prix, par exemple « 90 euros » ou au nombre de chambre, par exemple « 3 chambres ».

$$\left\{ \begin{array}{l} \text{RESERVATION} \\ \text{NUMBER} \\ \neg (\text{PRICE} \vee \text{ROOM}) \end{array} \wedge \right\} \rightarrow \{ \text{RESERVATION.res\_file\_nb.NUMBER} \} \quad (6.2)$$

Les règles sont écrites dans le format XML, car la représentation arborescente de ce format permet une meilleure intégration dans le prototype. En effet chaque nœud de la règle correspond à un connecteur, qui peut admettre au moins deux prédicats ou formules de prédicats. Les feuilles du document XML représentent les éléments du tour de dialogue dont on doit vérifier la présence. Le prototype gère le type d'éléments par le nom de la balise : mot, concept, frame, emplacement (nommé également « frame element »). L'exemple suivant permet de voir le document XML correspondant à l'équation 6.2. Si toutes les préconditions sont remplies, le prototype génère une action liée aux contraintes dictées dans la conséquence. Cela peut être la création, la modification, la suppression d'un élément ou de la liaison entre deux éléments. Dans l'exemple suivant la liaison se fait entre les deux éléments "f1" et "f2" en attachant "f2" sur "f1.fe1".

```
<ruleframe id="R22_bis">
  <imply>
    <preconditions>
      <and>
```

```

    <frame name="RESERVATION" id="f1" mode="require">
      <frameElement name="res_file_nb"
        id="fel" mode="require" />
    </frame>
    <frame name="NUMBER" id="f2" mode="require">
    </frame>
    <not>
      <or>
        <frame name="PRICE" id="f3" mode="require">
        </frame>
        <frame name="ROOM" id="f4" mode="require">
        </frame>
      </or>
    </not>
  </and>
</preconditions>
<consequence>
  <and>
    <frame name="RESERVATION" id="f1" mode="modify">
      <frameElement name="res_file_nb"
        id="fel" idput="f2" mode="modify" />
    </frame>
  </and>
</consequence>
</imply>
</ruleframe>

```

Les règles sont conçues en respectant l'ontologie définie pour le domaine de l'application MEDIA. Elles sont appliquées sur deux niveaux, le premier étant un ensemble de règles dont les préconditions sont des concepts et des mots, le deuxième étant un ensemble de règles dont les préconditions sont des frames. Un mécanisme de rappel de règles est mis en place à chaque instantiation de nouvelles frames ou liaison entre frames, afin que les règles n'ayant pas été vérifiées avant ces instantiations puissent être à nouveau mises en queue de traitement. Cette procédure permet d'éviter des oublis de vérification de règles, liés à l'approche séquentielle de la vérification des règles. Soit  $R$  un ensemble de règles. L'ordre de vérification des règles est aléatoire, et indépendant du concepteur des règles. Si on prend les règles dans l'ordre suivant :

1.  $R_1 : C \wedge D \rightarrow E$
2.  $R_2 : A \wedge C \rightarrow D$
3.  $R_3 : A \wedge B \rightarrow C$

en présence de  $A$  et de  $B$ , seule  $R_3$  est vérifiée et  $C$  est générée, les règles  $R_1$  et  $R_2$  ne donnent pas de résultats. Avec le rappel de règles, le déroulement du processus de vérification est le suivant

1.  $A$  et  $B$  sont présents,

2.  $R_1 : C \wedge D \rightarrow E$
3.  $R_1$  faux
4.  $R_2 : A \wedge C \rightarrow D$
5.  $R_2$  faux
6.  $R_3 : A \wedge B \rightarrow C$
7.  $R_3$  vrai, instantiation de C,
8.  $R_1$  contient C,  $R_1$  est ajoutée en queue de vérification,
9.  $R_2$  contient C,  $R_2$  est ajoutée en queue de vérification
10.  $R_1 : C \wedge D \rightarrow E$
11.  $R_1$  faux
12.  $R_2 : A \wedge C \rightarrow D$
13.  $R_2$  vrai , instantiation de D
14.  $R_1$  contient D,  $R_1$  est ajoutée en queue de vérification,
15.  $R_1 : C \wedge D \rightarrow E$
16.  $R_1$  vrai , instantiation de E
17. ...

### 6.1.6 Utilisation de fragments de frames et inférences

Ce prototype est une amélioration du précédent, et introduit des règles de composition liées aux facettes des emplacements de frames. L'ontologie fournit au prototype les facettes acceptables pour la composition de frames. Les règles utilisent des concepts, pour générer des structures plus complètes de frames : les fragments de frames. Le résultat final est un tableau qui fait correspondre à chaque concept un ou plusieurs fragments. En cas d'ambiguïté, le choix dépend des mots et du contexte. Le tableau 6.3 présente un extrait de ces correspondances. Les hypothèses génèrent ainsi des segments de connaissance ("Knowledge Chunks" [Fikes et Nilsson \(1971\)](#)).

Concept MEDIA	Fragment de frames associé
chambre-equipement	ROOM . r_facilities . FACILITY . r_facility_type . x
chambre-fumeur	ROOM . smoker . x
chambre-standing	ROOM . room_quality . x
chambre-type	ROOM . room_type . OCCUPANCY . occ_type . x
command-dial{annulation}	CANCEL . cancel_theme . x
command-dial{attente}	HOLD
command-dial{confirmation-annonce}	CONFIRM . confirm_theme . x
command-dial{confirmation-demande}	REQUEST . request_theme . CONFIRM . confirm_theme . x
command-dial{erreur}	ASSERT . assert_theme . error

Concept MEDIA	Fragment de frames associé
command-dial{modification}	CORRECT_MODIFY . [corr_theme . x1, corr_new_value . x2]
command-dial{repetition-demande}	REQUEST . request_theme . REPEAT
command-dial{repetition-demande}{+}	REPEAT . repeat_theme . REQUEST . request_theme
command-tache{guide}	LISTEN . listen_theme . guide
command-tache{information+}	REQUEST . request_theme . INFORMATION_TR
command-tache{reservation}	RESERVATION . reserv_obj . REFERENCE . ref_dest . prompt
comparatif-paiement	PRICE . price_relative_to . x
comparatif-temps	TIME . daytime_relative_to . x

TABLE 6.3 – Tableau de correspondance entre concept et fragments de frames

Un procédé de détection des relations entre frames, se basant sur les facettes possibles d’après l’ontologie, génère les liaisons. Les liaisons sont générées sur plusieurs niveaux, le niveau 0 correspond à la liaison entre un emplacement et une frame (tête de fragment) déjà existante et faisant partie d’une facette. Le niveau 1 correspond à l’inférence d’une frame  $F_{inf}$  correspondant à une facette d’un emplacement  $F.e$ , puis à la création d’un emplacement  $F_{inf}.e_2$  correspondant à une frame déjà existante. Le niveau 2 correspond à la création de deux frames faisant partie d’une chaîne de frames et dont la tête correspond à une frame facette et la queue correspond à un emplacement dont la facette est une frame déjà existante.

Les figures 6.5 et 6.7 montrent les fragments de frames tels qu’obtenus par correspondance entre les concepts et avant le processus d’inférence. La figure 6.6 compose par inférence de niveau 0, les fragments sont ainsi associés par le processus respectant l’ontologie décrite en Appendice A.2, et relie par exemple l’emplacement “establishment” avec la facette “HOTEL”, dont une instance est présente dans les têtes de fragments.

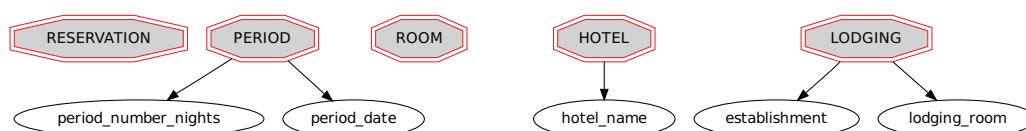


FIGURE 6.5 – Exemple de fragments qui peuvent être obtenus avant inférence de niveau 0

La figure 6.8 présente une inférence de niveau 1, avec l’inférence de la frame “LODGING” qui n’est pas présente dans les fragments de la figure 6.7. L’emplacement “establishment” dans la frame “LODGING” est créé pour que “establishment” se lie avec la frame “HOTEL” qui est une de ses facettes.

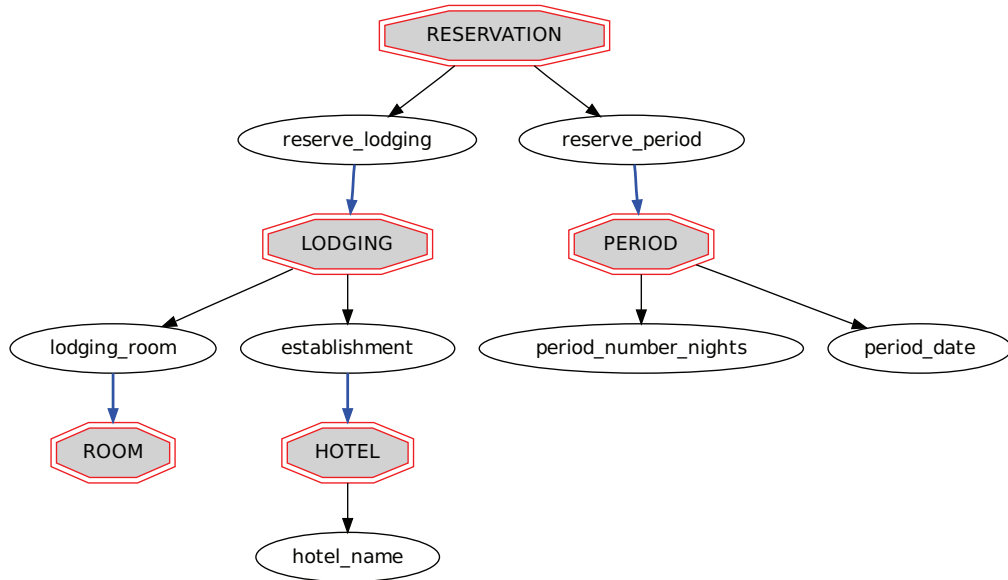


FIGURE 6.6 – Exemple de fragments composés par inférence de niveau 0 à partir de la figure 6.5

## 6.2 Acquisition progressive des caractéristiques syntaxiques supportant les relations sémantiques

Soit  $\Gamma_{i,a}$  une structure sémantique faite d'une instance de trame  $F_i$  avec les valeurs de certains de ses emplacements. La trame  $F_i$  est la tête de la structure  $\Gamma_{i,a}$ . Certaines valeurs d'emplacements de l'instance de  $F_i$  peuvent être des instances d'autres trames qui peuvent être des têtes d'autres structures. Soit  $G_{i,j}$ , soit le nom du  $j^{\text{ème}}$  emplacement de  $F_i$  et  $\sigma_{i,j,a,n}$  une chaîne d'emplacements reliant  $G_{i,j}$  à un emplacement  $B_{q,n}$  de l'instance d'une autre trame  $F_q$ .  $G_{i,j}$  est la tête de la chaîne et  $B_{q,n}$  est la queue. L'instance de la trame  $F_q$  peut être la valeur de  $G_{i,j}$ . Dans ce cas  $\sigma_{i,j,a,n} = G_{i,j}.F_q.B_{q,n}$ . Sinon,  $\sigma_{i,j,a,n} = G_{i,j}.\Sigma_x.F_q.B_{q,n}$  où  $\Sigma_x$  est une autre chaîne d'emplacements. Un exemple la chaîne  $\eta_{i,a,n}$

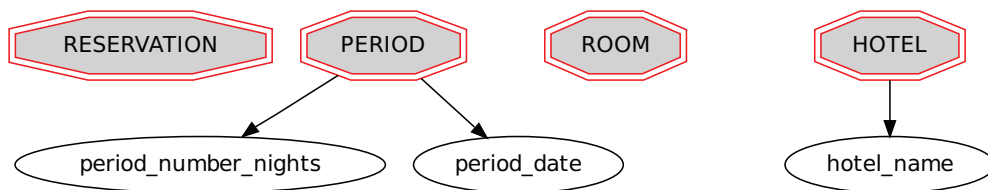


FIGURE 6.7 – Exemple de fragments qui peuvent être obtenus avant inférence de niveau 0



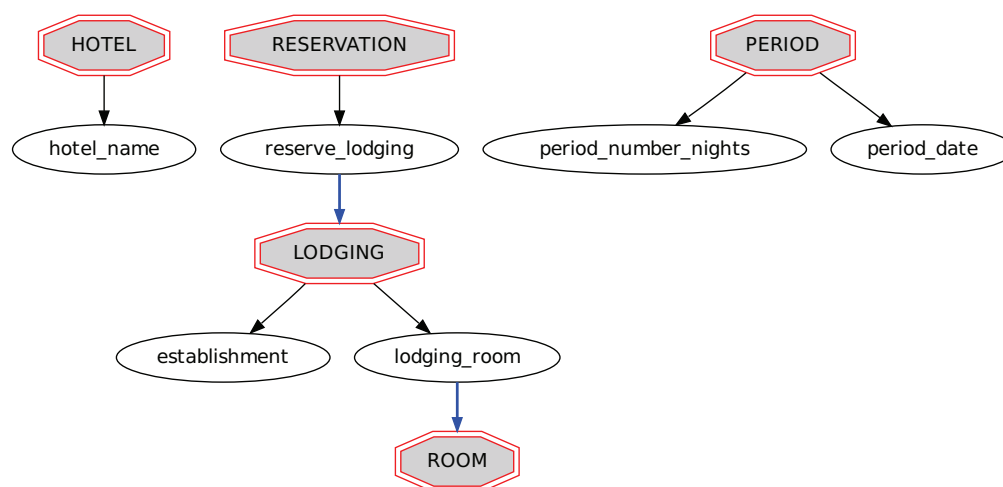


FIGURE 6.8 – Exemple de fragments composés par inférence de niveau 1 à partir de la figure 6.7

de  $\Gamma_{i,a}$  se terminant par l'emplacement  $B_{q,n}$  est définie comme suit  $\eta_{i,a,n} = F_i \cdot \sigma_{i,j,a,n}$ . L'emplacement  $B_{q,n}$  est alors la queue de  $\eta_{i,a,n}$ .

Une structure  $\Gamma_{i,a}$  a une seule tête et est une collection d'instances de chaînes dont chacune d'elle possède une queue. Le point "." indique une relation. La profondeur d'un symbole dans une chaîne d'instances est le nombre de symboles le précédant dans la chaîne.

Les motifs de support de relations sémantiques peuvent être obtenus en concevant des modèles de calcul sur de la connaissance explicite ou par apprentissage automatique à partir des données annotées. Des C.R.F. (Raymond et Riccardi (2007)), les machines à états finis (F.S.M.), les classificateurs et les réseaux bayésiens dynamiques (D.B.N.) Meurs et al. (2008) ont été utilisés à cette fin. Il y a des limites pratiques sur les méthodes d'apprentissage automatique lorsque les données disponibles pour l'entraînement du modèle sont insuffisantes Mitchell (1997). Lorsque l'apprentissage automatique ne peut pas être effectué, le support pour les relations sémantiques peut être obtenu par l'analyse des phrases parlées et par des fonctions de transposition des arbres d'analyse dans des hypothèses sémantiques. Les experts humains analysent les types de phrases difficiles, expliquent leurs relations avec les sens et conçoivent des motifs de supports afin de généraliser leurs conclusions.

Ce type de généralisation est réalisé grâce aux caractéristiques obtenues par apprentissage automatique. Un schéma de support obtenu de cette manière peut s'étendre sur tout le tour d'un dialogue et un mot représentant la tête de la structure est utilisé pour l'indexer. Ces supports sont mis en mémoire cache et utilisés comme déclencheurs signalant la présence de la phrase ou des types et leur interprétation associés. Avec cette forme d'apprentissage basée sur l'exploration (Explanation Based Learning (EBL)), il

est possible de combiner une connaissance préalable avec ce qui est obtenu par apprentissage automatique.

Une relation n-aire sémantique existe en général entre une instance de cadre et ses créneaux. Une telle relation peut être considérée comme une collection de relations binaires entre la frame et chaque emplacement. En outre, de nombreuses relations sémantiques sont prises en charge par les modèles exprimant des relations syntaxiques locales entre les supports des relations des constituants. Une discussion sur l'avantage d'obtenir des relations n-aires à partir de relations binaires peut être trouvée dans [Punyakank et al. \(2004\)](#).

### 6.2.1 Annotation des relations binaires

Un type de EBL peut être effectué si un corpus est annoté en termes de composants sémantiques et si les supports pour les constituants sont disponibles. Dans ce cas, les modèles pour les relations syntaxiques locales pour supporter les relations sémantiques peuvent être obtenues par les humains. Lorsque ces motifs ne s'appliquent pas, des fragments de structures sémantiques sont émis en hypothèse dans un tour de dialogue sans être composés. Ce fait devient une condition préalable à la présentation d'un exemple pour l'analyse humaine de l'annotation ou à l'exécution en ligne d'analyse lors de l'interprétation automatique.

Soit  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$  deux structures sémantiques qui peuvent être composées par une relation sémantique  $R_{i,h}$ . C'est le cas pour lequel  $\Gamma_{h,b}$  est la valeur d'un emplacement de  $\Gamma_{i,a}$ . Le support  $\Pi_{i,h}$  pour la relation sémantique  $R_{i,h}$  peut aussi être représenté par une relation syntaxique  $SYNR_{i,h} = SYNR(\Pi_{i,a}, \Pi_{h,b})$  entre le support  $\Pi_{i,a}$  de  $\Gamma_{i,a}$  et le support  $\Pi_{h,b}$  de  $\Gamma_{h,b}$ . Si  $\Pi_{i,a}$  et  $\Pi_{h,b}$  sont des supports de fragments obtenus à partir de concepts annotés dans *MEDIA*, ils sont manuellement hypothésés et validés. Leur relation syntaxique  $SYNR_{i,h}$  peut aussi être établie par EBL. Par exemple, si  $\Pi_{i,a}$  est une phrase verbale,  $\Pi_{h,b}$  est une phrase nominale et la séquence  $\Pi_{i,a}\Pi_{h,b}$  supporte une composition sémantique valide de  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$ , il est possible de se contenter d'observer la séquence de la validation de la composition, sans l'analyse complète de la phrase contenant les modèles. Cela correspond à stocker des caractéristiques liées pour les réutiliser. Dans le corpus d'apprentissage de *MEDIA*, ce type de  $SYNR_{i,h}$  est observé pour 74% des séquences  $\Pi_{i,a}, \Pi_{h,b}$ . En outre, dans 12% des cas, les modèles  $\pi_{i,a}$  et  $\pi_{h,b}$  sont séparés par un seul ou deux mots qui contiennent un mot fonction qui spécifie une relation syntaxique supportant une relation sémantique. Aussi dans ces cas, ces relations sont stockées pour être réutilisées. De cette façon, les relations syntaxiques complexes apparaissent sur moins de 15% des tours de dialogue. Dans ces cas, l'application de modèles de soutien à la recherche de relations peuvent nécessiter une connaissance contextuelle contenant les dépendances à distance entre les hypothèses de relations déjà émises. Des classes de supports peuvent être construites pour chaque concept. Cela nécessite que les hypothèses de relations soient émises, par étapes successives de processus de composition hiérarchique [Toutanova et al. \(2005\)](#). Les relations  $SYNR_{i,h}$  pour les supports consécutifs  $\pi_{i,a}$  et  $\pi_{h,b}$  sont manuellement validées et stockées dans une table  $TABSYNR(0)$ .

Pour les cas dans lesquels  $\pi_{i,a}$  et  $\pi_{h,b}$  sont séparés par un seul mot, un nouveau tableau *TABSYNR*(1) est établi. Ce tableau contient toutes les relations validées pour chaque type de mots entre les deux modèles. Les types les plus fréquents sont des mots hésitations ( par exemple, euh ...), les mots de fonctions, prépositions, déterminants, pronoms, les verbes spécifiques, des adverbes. Certains de ces mots sont des unités lexicales qui soutiennent l'instanciation de certaines composantes sémantiques, les références, connecteurs, la séparation des structures. Certains autres mots ne sont que la politesse ou un autre type de termes de comblement. La composition résultante sémantique est également une composition directe possible.

Pour les cas dans lesquels deux supports sont séparés par deux ou plusieurs mots, d'autres tableaux sont construits pour en faire une interprétation univoque possible, sinon les exemples sont proposés pour l'annotation manuelle.

Les relations syntaxiques binaires peuvent également être liées dans une chaîne. L'inspection et la généralisation sont effectuées pour déterminer si deux structures sémantiques peuvent être composées si leurs supports ne figurent pas dans une relation binaire. Mais ils apparaissent dans une chaîne de relations binaires ou sont liés par d'autres caractéristiques obtenues, par exemple, par l'analyse d'une phrase.

Les relations syntaxiques locales ne peuvent soutenir qu'une des relations sémantiques possibles parce que les deux structures sémantiques dont les supports apparaissent dans la relation syntaxique sont liées par une chaîne d'emplacements. Dans ce cas, une composition sémantique ne peut être effectuée que lorsque tous les éléments de la chaîne sont hypothétiques et leurs relations sont établies. Certains éléments peuvent être émis en hypothèse, par déduction, tandis que d'autres peuvent être émis en hypothèse par d'autres relations syntaxiques. La composition devient dans ce cas un problème de résolution. Les cas qui ne peuvent pas être résolus sans ambiguïté sont proposés à des annotateurs humains.

La procédure d'annotation a été appliquée à un ensemble de 17777 tours de dialogue subdivisés en différents ensembles. Chaque ensemble est caractérisé par le nombre de constituants hypothèses dans un tour. La forme générale de l'annotation est l'instance de  $\Gamma_{i,a}$  d'une structure ou un ensemble de structures d'instance. Une structure peut contenir l'instance d'une frame ou tout simplement l'exemple de certains emplacements de frames avec ou sans valeurs. Les valeurs peuvent être des instances d'autres frames en fonction du contenu des facettes associées à chaque emplacement.

Quand une instance  $\Gamma_{i,a}$  est générée par la détection d'un composant  $C_m$ , la validation est effectuée lorsque la relation de ce type s'applique :  $\pi_{m,s} \rightarrow C_m \rightarrow \Gamma_{i,a}$ . Une validation manuelle est effectuée seulement pour évaluer qu'un motif  $\pi_{m,s}$  engendre les phrases qui transmettent le sens représenté par  $\Gamma_{i,a}$ .

Il y avait 12.559 tours contenant deux constituants ou moins déjà annotés. Cela correspond à une couverture de 70,6 % du corpus. Parmi eux, 10.484 ont été trouvés par la validation automatique. Un échantillonnage aléatoire a été effectuée sur ces tours pour chaque type de support et ces exemples ont été soumis à la validation humaine en utilisant un outil de visualisation graphique appelé *Friz*, élaboré à cet effet. Tous les autres

cas ont été vérifiés manuellement. Les tours plus complexes ont été regroupés sur la base des constituants annotés. Des échantillons de chaque groupe ont été analysés par des experts humains. Les supports qui ont été identifiés, ont été généralisés et ajoutés à la connaissance d'interprétation.

Une procédure d'apprentissage progressif actif est en place sur la base de laquelle les exemples sont analysés avec des méthodes différentes, à savoir des F.S.M. et analyses de surface. Le désaccord dans une proposition résulte des annotateurs humains. Afin de mieux détecter les erreurs de suppression, des annotations automatiques sont comparées aux attentes établies par un modèle probabiliste de l'histoire du dialogue. La présence d'interprétations inattendues ou l'absence de celles attendues provoque une inspection humaine.

### 6.2.2 Processus de composition sémantique

La relation sémantique entre une instance de frame et son emplacement peut être considérée comme une collection de relations binaires entre la frame et chaque emplacement. Pour certains cas de la composition sémantique, il est possible d'entraîner des classificateurs pour l'apprentissage des supports des relations sémantiques binaires quand une quantité suffisante de données est disponible. Au contraire, il est peu probable que suffisamment de données disponibles pour l'apprentissage global des relations n-aires, car la taille des cas possibles croît exponentiellement avec  $n$ . Le problème de l'obtention des relations n-aires à partir de relations binaires est discuté dans McDonald et al. (2005) où des algorithmes sont proposés pour l'extraction des relations complexes pris en compte dans un ensemble de relations binaires. Pour ces raisons, le support pour la composition sémantique a été fondé sur les relations binaires.

Le processus de composition sémantique exécute une séquence de compositions binaires selon la théorie présentée chapitre 5.

Les instances  $\Gamma_{i,a}$  des structures sémantiques sont émises en hypothèse à partir de modèles de support  $\pi_{i,a}$ . Deux instances  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$  ont été émises en hypothèse et une relation syntaxique  $SYNR_{i,h}$  a été constatée entre eux. Le cas peut être composé en une nouvelle structure  $\Gamma_{i,c} = \Gamma_{i,a} \cdot \Gamma_{h,b}$  si la tête de frame  $F_h$  apparaît dans la facette de la queue d'une instance d'emplacement  $\eta_{i,a,n}$  de  $\Gamma_{i,a}$ .

Une autre composition possible est  $\Gamma_{h,c} = \Gamma_{h,b} \cdot \Gamma_{i,a}$  si la tête de frame  $F_i$  dans la facette de la queue d'une instance d'emplacement  $\eta_{h,b,y}$  de  $\Gamma_{h,b}$ . Il peut arriver que des compositions multiples soient possibles pour les deux instances  $\Gamma_{i,a}$  et  $\Gamma_{h,b}$ . Dans ce cas, une liste de compositions possibles (noté PCL, pour Possible Composition List)  $PCL(\Gamma_{i,a}, \Gamma_{h,b})$  est formée. La liste  $PCL(\Gamma_{i,a}, \Gamma_{h,b})$  est peuplée par des éléments après avoir évalué les prédicats et les fonctions qui sont désormais définis. Soit  $FACET(\eta_{i,a,n})$  la facette associée à la queue  $B_{q_n}$  de  $\eta_{i,a,n}$ . Plus généralement, soit  $FACET(B_{q,n})$  la facette associée à l'emplacement  $B_{q,n}$ . Tous les emplacements vides de l'instance  $\Gamma_{i,a}$  sont des candidats pouvant être remplis par une nouvelle instance  $\Gamma_{h,b}$ . Pour cette raison, il est utile de définir une liste  $UFS_{i,a}$  des emplacements vides pour une instance  $\Gamma_{i,a}$ .

La liste  $UFS_{i,a}$  peut être remplie par l'insertion de tous les emplacements vacants de l'instance de la frame  $\Gamma_{i,a}$  à partir de la tête de frame  $F_i$  et un examen systématique des têtes de frame des instances remplissant les emplacements de  $\Gamma_{i,a}$ .

Soit  $contains(FACET(B_{q,n}), F_h)$  un prédicat qui est vrai si et seulement si  $FACET(B_{q,n})$  contient  $F_h$  comme un type possible pour l'emplacement  $B_{q,n}$ . La définition est identique pour la facette d'une chaîne par exemple.

S'il est possible d'effectuer la composition  $\Gamma_{h,c} = \Gamma_{h,b} \cdot \Gamma_{i,a}$  par l'établissement du lien à la fin de la chaîne  $\eta_{i,a,n}$ , alors  $\eta_{i,a,n}$  est appelée instance de composition et la composition est indiquée comme suit  $\Gamma_{i,c}(\eta_{i,a,n}) = \Gamma_{i,a} \cdot \Gamma_{h,b}(\eta_{i,a,n})$ . Soit  $insert(L, e)$ , une fonction qui insère une composition possible  $e$  dans une liste  $L$ .

La liste  $PCL(\Gamma_{i,a} \cdot \Gamma_{h,b})$  est initialement vide. Tous les éléments de  $UFS_{i,a}$  sont examinés et les compositions possibles avec  $\Gamma_{h,b}$  sont identifiées et insérées dans  $PCL(\Gamma_{i,a} \cdot \Gamma_{h,b})$  avec la règle *précondition/action* :

$$\begin{aligned} \forall(x) \forall(s \in UFS_{i,a}) SYN R_{i,h} \wedge contains(FACET(s, F_h)) \\ \rightarrow insert(PCL(\Gamma_{i,a}, \Gamma_{h,b}), \Gamma_{i,a} \cdot \Gamma_{h,b}(\eta_{i,a,s})) \end{aligned}$$

où  $SYN R_{i,h}$  est le prédicat qui est vrai lorsqu'une relation syntaxique existe,  $\eta_{i,a,s}$  est l'instance de chaîne avec la tête de frame  $F_i$  et une queue  $s$ .

Avec la méthode proposée, l'ontologie de domaine définit les relations sémantiques possibles impliquant les constituants hypothèses du signal de parole. Les schémas de support pour les constituants sont connus et analysés hors ligne. Les relations sémantiques sont prises en charge par les caractéristiques des relations entre ces fragments d'analyse. La prise en charge de ces relations peut être obtenue hors ligne avec des analyseurs de surface et les classificateurs s'il y a assez de données disponibles ou par EBL autrement. On a fait observer très souvent dans les dialogues *MEDIA*, le support syntaxique pour les relations sémantiques sont répétitifs et de fréquentes paires de constituants présentent un nombre limité de modèles de mots supports de leurs relations sémantiques. Ainsi, les relations syntaxiques pour soutenir les relations sémantiques, ont été déterminées avec l'analyse superficielle et EBL.

Si les composants sont l'hypothèse de séquences de mots touchés par les erreurs, alors l'analyse du support correct correspondant est prise en compte. Les candidats initiaux pour les compositions sont obtenus par le calcul de la liste structurée des  $n$  meilleures hypothèses de constituants pour lesquels le prédicat  $contains(FACET(s, F_h))$  est vrai. L'inférence logique peut être effectuée pour en déduire la chaîne des possibles fragments manquants dans une chaîne d'instances. Prenons comme exemple la chaîne  $\eta_{1,j,a,n} = F_1 \cdot A_{1,j} \cdot \Gamma_{2,1} \cdot D_{2,k} \cdot \Gamma_{3,1} \cdot B_{3,n}$  et soit  $V_{3,n}$  la valeur de l'emplacement  $B_{3,n}$ . Cette instance peut être considérée comme une relation sémantique entre l'instance de  $\Gamma_1$  de  $F_1$  et la paire de valeurs d'emplacement  $(B_{3,n}, V_{3,n})$ .

Considérons  $R_{1,3,n}$  comme une telle relation. Supposons que seuls les composants sémantiques  $C_1$  et  $C_3$  supportant respectivement  $\Gamma_1$  et  $(B_{3,n}, V_{3,n})$ , ont été détectés à

partir d'une séquence hypothèse de mots  $W$  à partir du signal acoustique représenté par des caractéristiques acoustiques  $Y$ . Supposons que la relation syntaxique  $SYNR_{1,3}$  a été détectée entre les supports des deux constituants. L'inférence logique consiste à déduire  $\eta_{1,j,a,n} \cdot V_{3,n}$  de l'affirmation de  $\Gamma_1$  et  $(B_{3,n}, V_{3,n})$  soutenu par la chaîne  $C_1.SYNR_{1,3}.C_3$  qui est, à son tour soutenu par  $W$ , qui est soutenu par  $Y$ .

Le problème est de prévoir un lien du type  $R_{1,3,n}$  entre  $\Gamma_1$  et  $(B_{3,n}, V_{3,n})$  qui peut être formulé en termes de structures relationnelles probabilistes. La procédure pour obtenir une liste structurée de constituants hypothésés est décrite dans [Raymond et al. \(2006\)](#). Les candidats potentiels sont ensuite validés par la détection des relations syntaxiques et sont évalués sur la base des preuves acoustiques et linguistiques de leurs supports.

La procédure proposée pour la composition des structures sémantiques a été testée sur 2593 tours annotés manuellement impliquant la composition d'un maximum de cinq instances des frames. Les résultats présentés dans le tableau 6.4 ont été obtenus avec [F.S.M.](#) et des [C.R.F.](#). Le taux d'erreurs mot est de 29,1% et le taux d'erreurs sur les constituants est de 28,9% avec les [C.R.F.](#) et de 31,3% avec [F.S.M.](#). Le tableau 6.4 montre le pourcentage de tours correctement représentés par la composition des structures sémantiques, après déduction des inférences, et la recherche dans la liste structurée des constituants, des segments de la chaîne d'instances hypothésée par les possibles inférences réalisées sur les chaînes d'instances disponibles.

Compositions faites à partir de	1-meilleure	avec inférences	récupérées à partir des n-meilleures
constituants <a href="#">F.S.M.</a>	72.50%	74.47%	79.37%
constituants <a href="#">C.R.F.</a>	75.16%	77.47%	82.37%

TABLE 6.4 – Pourcentage de tours correctement interprétés sur le corpus de test (2593 tours)

Un résultat Oracle de 2332 (89,9%) tours correctement interprétés a été trouvé considérant les hypothèses générées à partir d'un treillis de mots. Les tours de plus de cinq frames sont moins fréquents (moins de 30 % de l'ensemble du corpus).

### 6.3 Expérience de composition sémantique

Le système automatique transforme les concepts en fragments de frames, par exemple : le concept `hotel_facility(parking)` est associé au fragment : `HOTEL.[h_facilityFACILITY.[facility_type.parking]]`.

Le système produisant automatiquement des frames fonctionne sur les concepts sémantiques de base. Il dérive alors ces concepts en fragments de frames. Ils respectent une ontologie liée au domaine d'application. Cette ontologie liée au domaine est créée par un expert, ainsi que les règles d'appariement entre concepts et frames. Chaque concept caractérise une connaissance et peut correspondre à un ou plusieurs fragments de l'ontologie. Cela dépend de la fonction du mot, de son support et du contexte. Dans l'application `MEDIA`, certains concepts génèrent des fragments représentant des actes



de dialogues. Le support pour ces fragments est des verbes et les rôles de ces verbes sont les valeurs de fragments. L'exemple suivant montre la correspondance entre un concept et le fragment de frame associé.

Si nous prenons la phrase issue de la reconnaissance : "je voudrais que euh euh euh pour quatre nuits cela ne coûte moins de cent euros donc des informations sur le prix des chambres a soissons". L'annotation automatique en constituants sémantiques basiques hypothétise les concepts suivants en italique, leur support étant entre accolades : *request*{je voudrais} *null*{que euh euh euh} *sejour-nbNuit*{pour quatre nuits} *null*{cela ne coute} *comparatif-paiement*{moins de} *paiement-montant-entier*{cent} *paiement-monnaie*{euros} *connectProp*{donc} *know*{des informations} *null*{sur} *lienRef-coRef*{le} *objet*{prix} *null*{des} *objet*{chambres} *localisation-ville*{a soissons}.

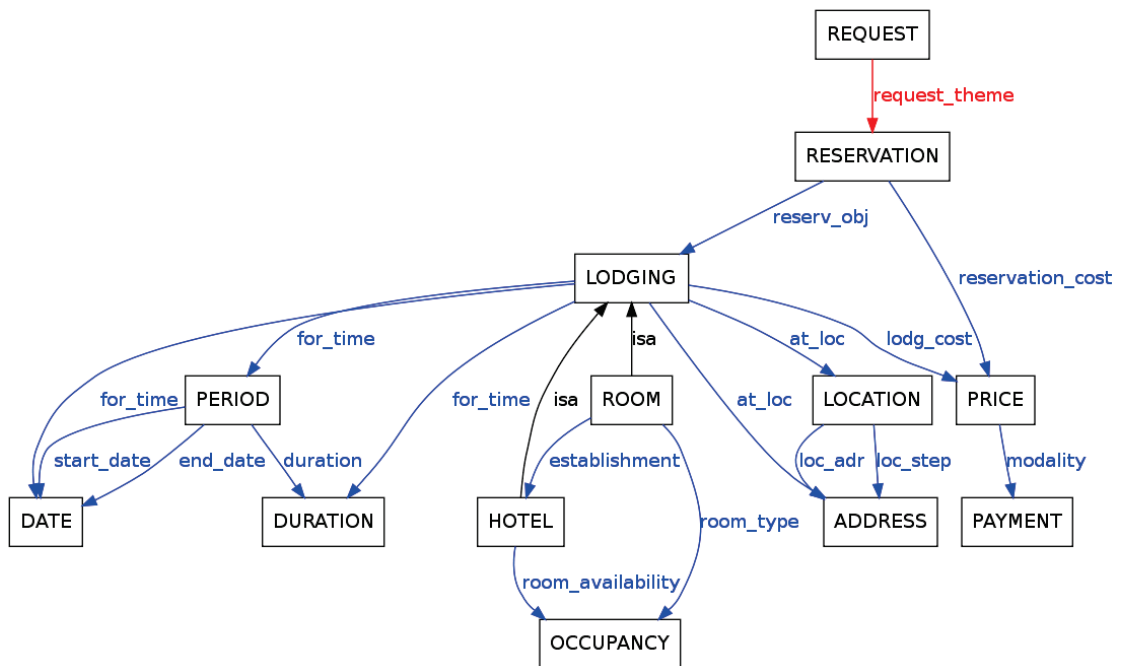
L'étape de transformation de concepts en fragments de frames donne les frames suivantes en majuscules et les emplacements en minuscules : *REQUEST.request\_theme*, *ROOM*, *LOCATION.loc\_adr.ADDRESS.adr\_city*, *PRICE.[amount, currency, price\_relative\_to]*,...

Le système compose ensuite des fragments de frames, crée un graphe d'hypothèses des relations possibles entre fragments. Cette composition reste et doit rester cohérente par rapport à l'ontologie fixée. La composition se fait en regardant l'ensemble des frames qui peuvent s'associer avec un emplacement vide d'un fragment, en évitant les cycles. Un système d'inférences de frames est mis en place pour relier deux frames même si la frame intermédiaire les reliant n'est pas détectée par le système de reconnaissance. Les inférences permettent la création des frames intermédiaires. Soit  $\delta_n = F_d.G_{d,j} F_q.S_{q,n}$  et  $\delta_m = F_r.S_{r,m}$ , en respectant l'ontologie :  $\exists F_h | F_r.G_{r,m} F_h.G_{h,p} F_d.G_{d,j} F_q.S_{q,n}$ .

Si on considère l'ontologie du domaine de reservation d'hôtel, représentée partiellement par la figure 6.9, on réalise ainsi les connections entre les fragments de frames.

Le graphe des associations possibles entre frames est dérivé ensuite au moyen de règles de transpositions vers des constituants sémantiques basiques auxquels, on rajoute des spécifieurs correspondant à une partie du graphe traité, à l'aide de règles de correspondances définies par l'expert. On obtient un ensemble de concepts et spécifieurs possibles s'appliquant à un tour de dialogue. Pour chaque constituant du départ, on cherche alors la séquence de spécifieurs la plus longue parmi les possibilités données par le graphe. On désambiguïse ainsi les concepts. Cela permet de représenter, non plus la phrase ou le tour de dialogue dans une représentation plate, mais dans une représentation hiérarchisée avec des liens entre les concepts. La désambiguïsement des concepts permet de prendre en compte le contexte général de la phrase et permet au moyen d'inférences de pallier certaines erreurs dues essentiellement à la R.A.P. .

Un exemple de graphe d'hypothèses est présenté dans la figure 6.13. On voit sur la figure 6.9, que l'on peut inférer plusieurs frames. Par exemple : on infère la frame *RESERVATION*, entre *REQUEST* et *ROOM* puisque *ROOM* hérite de *LODGING* (lien  $is_a$ ), le nouveau fragment de est *REQUEST.request\_theme.RESERVATION.reserv\_obj.ROOM*. On peut aussi inférer la frame *HOTEL* par une règle logique du type : "Si  $\exists(x, y, z) | x \in$



**FIGURE 6.9** – Représentation partielle de l'ontologie du domaine, les frames sont en majuscules et les emplacements ou éléments de frames sont en minuscules, sur les flèches. Celles-ci représentent les liens entre frames. Cette ontologie a été créée à partir de "Protégé 3.4.3" disponible sur <http://protege.stanford.edu>

$LOCATION \wedge y \in ROOM \wedge z \in RESERVATION$  alors on infère la frame HOTEL.

Cette étape terminée, le graphe des hypothèses de composition permet de transformer ces groupes ces liens entre les frames en une structure plate, c'est à dire concept + spécifieur. Dans l'exemple suivant, on obtient : *request-reservation*{je voudrais} *null*{que euh euh euh} *sejour-nbNuit-reservation*{pour quatre nuits} *null*{cela ne coute} *comparatif-paiement*{moins de} *paiement-montant-entier-reservation-chambre*{cent} *paiement-monnaie*{euros} *connectProp*{donc} *know*{des informations} *null*{sur} *lienRef-coRef*{le} *objet*{prix} *null*{des} *objet*{chambres} *localisation-ville-hotel*{a soissons}.

### 6.3.1 Résultats

Les dialogues ont été transcrits par le système Reconnaissance Automatique de la Parole ( R.A.P. ) SPEERAL Nocera et al. (2002) et les transcriptions ont été annotées en termes de concepts et structures de frames.

Les expérimentations ont porté sur le corpus de test de MEDIA , sur 3000 tours de dialogue. Ce corpus a été manuellement annoté en termes de concepts, valeurs, spécifieurs sémantiques. Les annotations de référence en frames ont été obtenues automatiquement comme décrit dans le rapport final LUNA (2009), et toutes manuellement corrigées. On reporte ici les résultats obtenus sur les transcriptions automatiques de la



R.A.P. avec un taux erreurs mots de 27.4%.

Les tours de dialogues sont évalués par rapport à l’annotation en frames de référence. On compare la production de frames et la composition en frames complètes dans le tour. Deux systèmes sont évalués pour cette production. Système A - composition par règles en utilisant seulement des relations syntaxiques locales. Système B - composition par apprentissage à l’aide des C.R.F. utilisés pour calculer les probabilités utiles pour la décision finale. Les deux systèmes utilisent ensuite un même processus de génération de structures de frames par automates à états finis et des C.R.F. .

Les deux systèmes d’annotations présenté ici, ont été évalués sur le corpus de test de MEDIA avec le protocole suivant. Un étiqueteur C.R.F. de concepts est appliqué sur les transcriptions automatiques pour produire les séquences de concepts ; puis nous appliquons les systèmes d’annotations en frames directement sur les séquences de concepts et les transcriptions. Les métriques employées ici pour l’évaluation sont décrites dans l’appendix B, Les tableaux 6.5 et 6.6 résument les résultats obtenus respectivement avec le système d’annotation de frames à base de règles, et le systèmes d’annotation de frames à base de C.R.F. .Le taux d’erreurs “mots” ( W.E.R. ) du module de R.A.P. est de 27.4%, et le taux d’erreurs “concepts” ( C.E.R. ) de la séquence de concepts est de 31.3%.

Métrique	Précision	Rappel	F-mesure
Reconnaissance de frames	78.15	75.90	77.01
Détection des limites des arguments	87.21	76.99	81.78
Détection des étiquettes des arguments	76.78	72.05	74.34
Reconnaissance des arguments	70.55	66.20	68.31
Réalisation des frames	59.87	59.03	59.45
Composition des frames	73.56	63.73	68.29
Analyse complète du tour	46.77	45.98	46.37

TABLE 6.5 – Résultats du système à base de règles sur le corpus de test

Métrique	Précision	Rappel	F-mesure
Reconnaissance de frames	78.37	77.28	77.82
Détection des limites des arguments	75.41	66.22	70.52
Détection des étiquettes des arguments	76.24	70.74	73.39
Reconnaissance des arguments	70.06	65.01	67.44
Réalisation des frames	59.17	57.46	58.30
Composition des frames	72.73	62.01	66.94
Analyse complète du tour	46.37	45.46	45.91

TABLE 6.6 – Résultats du système à base de C.R.F. sur le corpus de test

Nous avons comparé notre travail avec un système de composition de frames : la composition sémantique avec des Réseaux Dynamiques Bayésien ( D.B.N. ). Un système D.B.N. a été proposé et testé pour dériver une représentation sémantique hié-

rarchique en frame, d'un tour de dialogue. Cette approche est basée sur un double processus de décodage utilisant les *D.B.N.* et un processus final de composition défini par *Meurs et al. (2008)* utilisant les Machines à Vecteurs de Support ( *S.V.M.*<sup>2</sup> décrit *Collobert et Bengio (2001)*). Le double processus de décodage est utilisé pour fournir les concepts basiques puis les frames. Dans une étape finale, un processus de recombinaison de sous-arbre compose les branches hypothétisées. Un classifieur est utilisé pour décider comment les branches seront connectées. Les résultats de cette méthode sont dans le rapport final de *LUNA (2009)*, avec une F-mesure sur les métriques.

Comme on peut le constater l'approche basée sur les règles et les inférences pour la désambiguïisation donne de moins bons résultats sur les métriques que la méthode utilisée avec les *C.R.F.* . Cette approche est moins robuste car il ne considère que les relations syntaxiques locales. Cependant la méthode décrite ici, permet de voir que la génération et l'inférence de frames dans un contexte hypothétisé (et où le langage naturel oral produit des erreurs liées aux disfluences, bruits etc.) semblent prometteuses puisque les résultats sont proches de ceux des *C.R.F.* . Cet ensemble de règles sont générées par un expert. Cette méthode se déroule sans aucun apprentissage et permet d'avoir des résultats proches des systèmes à base d'apprentissage. Les frames générées à partir des hypothèses de transcription, des concepts et des spécifieurs ajoutés aux concepts, permettent d'interpréter ce qui a été dit, et d'émettre des hypothèses sur ce qui aurait pu être mis en défaut par la *R.A.P.* , tout en respectant l'ontologie définie.

#### 6.3.2 Désambiguïisation sémantique

L'approche que nous abordons ici a pour but d'utiliser les informations, générées par la compréhension et la composition automatique, décrite dans la section précédente, dans le but de désambiguïiser les concepts de base, au moyen de informations complémentaires appelées les spécifieurs.

Différentes approches ont été menées pour la désambiguïisation, *Dahlgren (1988)* utilise plusieurs informations pour la compréhension du langage naturel : des groupes de mots figés, des sélections syntactico-sémantiques et un moteur d'inférences qui cherche un ancêtre commun à deux mots appartenant au contexte dans une ontologie comme le fait *Resnik (1995)*. Ces méthodes pour la plupart fonctionnent avec une ontologie générale, et non spécifique au domaine, et se basent en général sur Wordnet *Fellbaum (1998)*. Ces systèmes fonctionnent sur des règles syntaxiques et des mots du langage naturel se rapportant au contexte, l'ontologie doit être très riche et est non robuste aux erreurs liées à la *R.A.P.* . Notre approche favorise la conception d'une ontologie basée sur les concepts et non sur les mots du vocabulaire. Elle est liée au domaine, créée par un expert humain, reproductible suivant le domaine.

D'autres approches favorisent l'apprentissage c'est le cas du système qui utilise des *C.R.F.* , principalement pour l'étiquetage de mots, d'entités nommées ou des parties du discours. Les *C.R.F.* ainsi que l'utilisation des "Dynamic Bayesian Networks" avec

---

2. Disponible sur <http://www.support-vector-machines.org>

des classifieurs [Lefèvre \(2007\)](#), sont des approches discriminantes. Le principal avantage des [C.R.F.](#) correspond à la capacité de prédire une étiquette conformément à un ensemble de caractéristiques liées à l'intégralité du message, et pas seulement la faible proximité du mot à étiqueter. Ceci est très important pour la tâche d'ajout de spécifieurs à des concepts. Cette information dépend de caractéristiques qui peuvent être loin de l'étiquette dans le message. Cette approche est similaire à notre système dans le fonctionnement puisque l'information concernant la relation entre deux concepts peut être récupérée dans le tour du dialogue. Cependant les [C.R.F.](#) nécessitent un apprentissage, et une annotation d'une référence au préalable de l'ensemble d'entraînement. L'utilisation des règles définies, permet de garder la cohérence des relations et la génération des relations sans apprentissage.

### 6.3.3 Exemple de désambiguïisation d'un tour de dialogue

Le système présenté ici, est une dérivation de l'usage du système précédent. Il propose la désambiguïisation des concepts à l'aide du graphe des possibles liaisons de frames. Le système compose les fragments de frames, crée un graphe d'hypothèses des relations possibles entre fragments. Cette composition reste et doit rester cohérente par rapport à l'ontologie fixée. La composition se fait en regardant l'ensemble des frames qui peuvent s'associer avec un emplacement vide d'un fragment, en évitant les cycles. Un système d'inférences de frames est mis en place pour relier deux frames même si la frame intermédiaire les reliant n'est pas détectée par le système de reconnaissance. Les inférences permettent la création des frames intermédiaires.

Le graphe des associations possibles entre frames est dérivé ensuite au moyen de règles de transpositions vers des constituants sémantiques basiques auxquels, on rajoute des spécifieurs correspondant à une partie du graphe traité, à l'aide de règles de correspondances définies par l'expert. On obtient un ensemble de concepts et spécifieurs possibles s'appliquant à un tour de dialogue. Pour chaque constituant du départ, on cherche alors la séquence de spécifieurs la plus longue parmi les possibilités données par le graphe. On désambiguïse ainsi les concepts. Cela permet de représenter, non plus la phrase ou le tour de dialogue dans une représentation plate, mais dans une représentation hiérarchisée avec des liens entre les concepts. La désambiguïisation des concepts permet de prendre en compte le contexte général de la phrase et permet aux moyens d'inférences de pallier certaines erreurs dues essentiellement à la [R.A.P.](#) .

Voici un exemple complet du processus tiré du corpus de test dont la transcription et l'annotation en concepts sont issus de processus automatiques. Cet exemple est le même que le tour de dialogue présenté dans les figures ???. On reprend et détaille ici ce tour de dialogue afin d'en faciliter la lecture.

#### Exemple (6.3.7)

Soit la phrase de référence : « euh non c'est pas gênant ça c' est pas gênant euh vous pouvez euh alors est-ce que vous pouvez me confirmer tous les éléments donc ils ont bien une baignoire un téléphone et c' est à quatre-vingt-dix euros c' est ça et à côté de notre-dame de paris ».

Les concepts de référence correspondant et leur spécifieur sont montrés dans le tableau 6.7 :

TABLE 6.7 – Exemple de la transcription et des annotations manuelles d'un tour de dialogue du corpus *MEDIA*, en concepts, valeurs et spécifieurs

Concept	Spécifieur	Valeur	Transcription
null	null	null	euh
reponse		non	non
null	null	null	c'est pas gênant
null	null	null	ça c' est pas gênant
null	null	null	euh vous pouvez euh
null	null	null	alors est-ce que
command-dial		confirmation-demande	vous pouvez me confirmer
objet		undetermined	tous les éléments
connectProp		implique	donc
lienRef-coRef		pluriel	ils
null	null	null	ont bien
chambre-equipement		bain	une baignoire
chambre-equipement		telephone	un téléphone
connectProp		addition	et
null	null	null	c' est à
paiement-montant-entier	-reservation-chambre	90	quatre-vingt-dix
paiement-monnaie		euro	euros
command-dial		confirmation-demande	c' est ça
connectProp		addition	et
localisation-distanceRelative	-hotel	proche	à côté de
localisation-lieuRelatif	-nomme-hotel	notre dame de paris	notre-dame de paris

Le tableau 6.8 propose la transcription automatique par un système de *R.A.P.* du même tour de dialogue.

A partir de cette transcription automatique et des concepts issus du système automatique, des fragments de frames sont dérivés. Ces fragments sont illustrés dans les figures 6.11 et 6.12.

Le système analyse ensuite les fragments et compose toutes les associations et inférences possibles en se basant sur l'ontologie. La figure 6.13 montre toutes les associations possibles pouvant être faites sur l'exemple.

Ces inférences et associations ne permettent pas une interprétation du tour de dialogue, puisque le système produit toutes les relations possibles, mais ces associations permettent de mettre en contexte les différents fragments. Ainsi les concepts sont mis en relation dans le tour de dialogue. On peut donc dériver les hypothèses d'associations en concept avec des spécifieurs. Les concepts et spécifieurs sont passés en entrée

**TABLE 6.8** – Exemple de la transcription et d'annotations automatiques d'un tour de dialogue du corpus *MEDIA*, en concepts et valeurs

Concept	Valeur	Transcription <i>R.A.P.</i>
null	null	euh euh euh oov s' agit d'
lienRef-coDom	pluriel	autres
objetBD	hotel	hotels
null	null	euh vous pouvez me alors
QUERY	VALEUR	est_ce_que
null	null	couple tous les elements donc
lienRef-coRef	pluriel	ils
null	null	ont
chambre-equipement	bain	des baignoires
chambre-equipement	telephone	et telephones
null	null	de est a
paiement-montant	90	quatre vingt dix
paiement-monnaie	euro	euros
command-dial	confirmation-demande	c' est ça
connectProp	addition	et
localisation- distanceRelative	proche	a cote
null	null	d' autre part
localisation-ville	paris	de paris

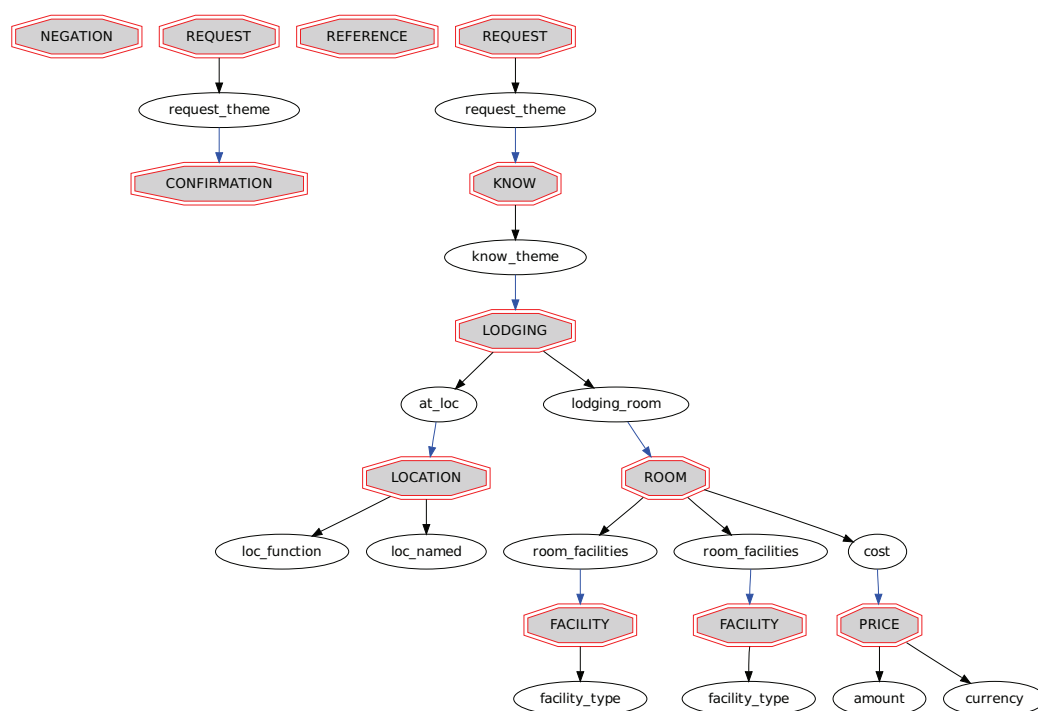


FIGURE 6.10 – Annotation de référence en frame basée sur la phrase de l'exemple 6.3.7

au système qui génère les interprétations en frames pour le tour de dialogue présenté dans la figure 6.14.

## 6.4 Détection de lambda-expressions

Dans cette partie, nous proposons une méthode pour l'hypothétisation de lambda-expressions d'objets en référence dans les dialogues téléphoniques du corpus *MEDIA*. Les relations entre les mots, les constituants sémantiques et leurs compositions dans des lambda-expressions sont modélisés par des *C.R.F.*, dans lesquels des fonctions intègrent des modèles de motifs dérivés manuellement avec des hypothèses de mots et de concepts. Des réductions d'erreurs substantielles sont obtenues en utilisant ces fonctions en lieu et place de flux séparés en entrée par les *n*-grammes de mots et de concepts. Les modèles et les motifs dérivés manuellement semblent être très utiles et non difficiles à obtenir, en généralisant des exemples significatifs détectés par la présence de constituants sémantiques spécifiques.

Dans le corpus *MEDIA*, il y a beaucoup de tours de dialogues dont l'interprétation nécessite des activités qui sont plus complexes que le remplissage d'emplacements de

**TABLE 6.9** – Exemple de la transcription et d’annotations automatiques d’un tour de dialogue du corpus *MEDIA*, en concepts, valeurs et spécifieurs.

Concept	Spécifieur	Valeur	Transcription R.A.P.
null		null	euh euh euh oov s’
lienRef-coDom	-hotel	pluriel	agit d’
objetBD		hotel	autres
null		null	hotels
QUERY		VALEUR	euh vous pouvez me
null		null	alors
lienRef-coRef	-hotel-reservation	pluriel	est_ce_que
null		null	couple tous les ele-
chambre-equipement		bain	ments donc
chambre-equipement		telephone	ils
null		null	ont
paiement-montant	-entier-reservation	90	des baignoires
paiement-monnaie		euro	et telephones
command-dial		confirmation-	de est a
connectProp		demande	quatre vingt dix
localisation-	-hotel	addition	euros
distanceRelative		proche	c’ est ça
null		null	et
localisation-ville	-hotel	paris	a cote
			d’ autre part
			de paris

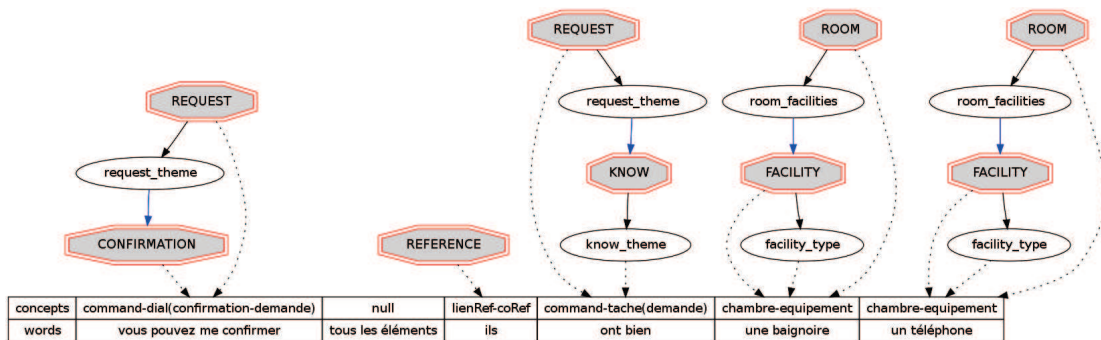


FIGURE 6.11 – Etape de mise en hypothèses des fragments de frames à partir des concepts - 1<sup>er</sup> partie de phrase

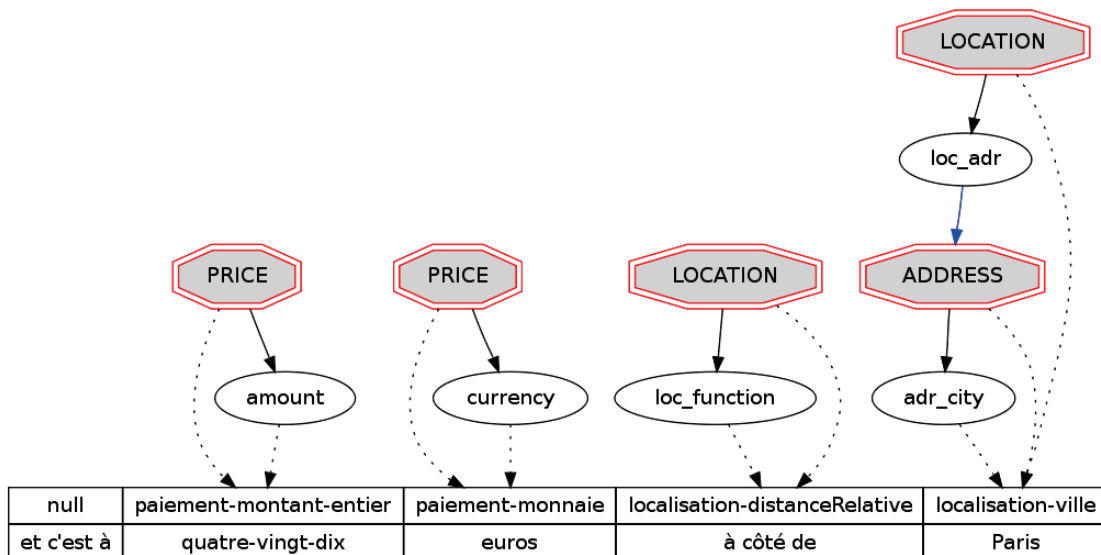


FIGURE 6.12 – Etape de mise en hypothèses des fragments de frames à partir des concepts - 2<sup>ième</sup> partie de phrase

frames. Dans beaucoup de tours, les références sont faites vers des objets en spécifiant certaines propriétés. Le type des objets référencés peut être déduit par des fonctions associées à des emplacements de frame, pour l'évaluation des valeurs des emplacements. Ces fonctions ne sont pas dans le corpus annoté. Considérons, par exemple, la phrase :

#### Exemple (6.4.8)

Exemple de phrase tirée du corpus MEDIA :

S'il vous plaît, faites la réservation la plus proche de l'événement

dans un tour de dialogue qui suit une négociation dans laquelle plusieurs hypothèses ont été envisagées et progressivement affinées. Le choix entre elles est indiqué par une fonction logique dont la valeur est l'une des structures hypothèses. Ces fonctions, peuvent être représentées par des expressions de lambda-calcul qui sont des extensions



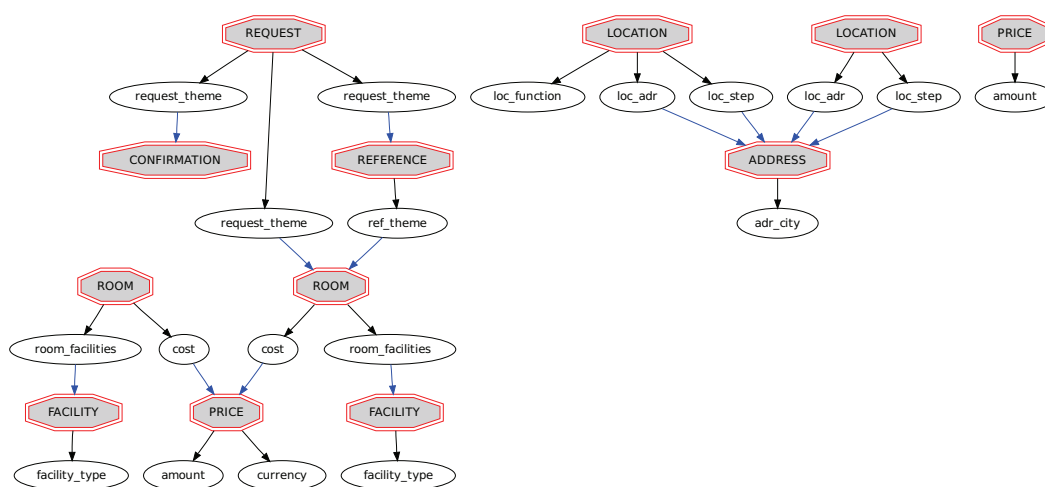


FIGURE 6.13 – Graphe des possibles associations à partir des fragments du tableau 6.8

de notation de la logique du premier ordre.

Récemment, des approches ont été proposées pour apprendre comment dériver des lambda-expressions décrivant le sens d’une phrase dans le texte écrit Wong et Mooney (2007); Zettlemoyer et Collins (2009). Nous proposons une approche pour effectuer ce type d’inférences à partir de transcriptions de R.A.P. sujettes aux erreurs sur des tours de dialogues parlés. A cause du bruit dans les transcriptions automatiques de phrases prononcées, plutôt que d’effectuer des compositions de lambda-fonctions à partir de la sémantique des mots, nous proposons une méthode pour supposer directement une lambda-expression à partir d’hypothèses déjà émises de mots et de composants sémantiques. Les relations entre les mots, les concepts et les expressions sont représentées par des graphes. Sur la base de cette représentation, il apparaît que les expressions peuvent être émises en hypothèse par des modèles de C.R.F. . Les relations étudiées ici ne sont pas des co-références, même si certaines considérations utilisées dans la conception des C.R.F. sont similaires à celles utilisées, par exemple dans McCallum et Wellner (2003) pour la modélisation des incertitudes d’identité avec application à la co-référence.

Une analyse d’erreurs est également effectuée. Elle indique comment les différentes séries d’exemples peuvent être construits de telle manière que chaque ensemble concerne des erreurs introduites par des composants spécifiques de R.A.P. ou de C.L.P. . Les données de chaque série peuvent être utilisées pour améliorer les performances du module correspondant.

L’exemple 6.4.9 montre les balises de concepts et leurs supports de mots pour une phrase du corpus MEDIA .

Exemple (6.4.9)

<Reservez> [command(reservation)] <le deuxième> [number(second)] <a Lyon > [city(Lyon)]

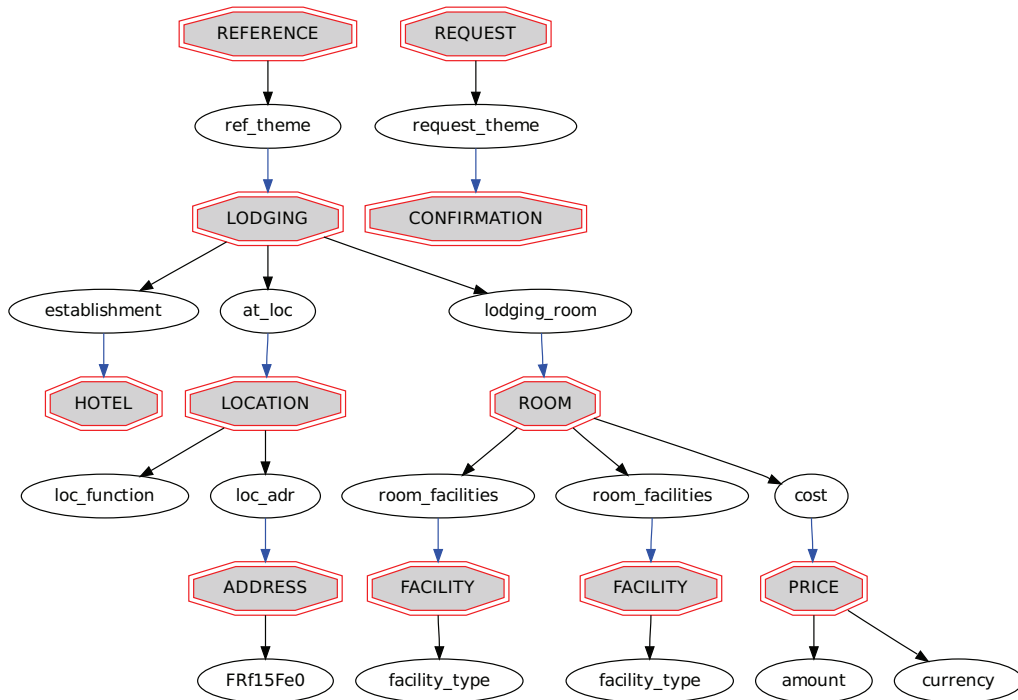


FIGURE 6.14 – Interprétation du tour de dialogue issu du système *R.A.P.* à partir des concepts et spécificateurs donnés dans le tableau 6.9

Soit  $C_1^B = c_1, \dots, c_b, \dots, c_B$ , avec  $c_b \in V_C$ , la séquence d'étiquettes de concepts représentés entre crochets. Une approche pour la mise en hypothèse de ces concepts dans un système de *C.L.P.* est proposé dans *Raymond et Riccardi (2007)*. Un tour contient une demande concernant des instances de structures sémantiques déjà émises en hypothèse dans l'historique du dialogue. Une fonction doit être introduite pour récupérer la structure visée. Soit  $\Psi$ , la structure visée par la requête. Dans l'exemple 6.4.9,  $\Psi$  est l'ensemble des solutions proposées par le système en réponse aux précédentes demandes des utilisateurs. Une fonction doit être introduite pour représenter une référence permettant de récupérer  $\Psi$ . Cette fonction, indiquée  $\lambda_j(x_1^L)$ , est une lambda-expression de  $L$  variables établissant une relation sémantique entre certains concepts de  $C_1^B$  et  $\Psi$ . Soit  $W = w_1, \dots, w_n, \dots, W_N$ , la séquence de mots émis en hypothèse dans le tour de dialogue. L'information disponible pour la formulation d'hypothèses  $\lambda_j(x_1^L)$  est la séquence d'hypothèses de mots et d'étiquettes de concepts.

Les supports des balises de concepts ne se chevauchent pas. Les mots qui n'appartiennent pas au support de la balise de concepts sont associés au concept "null", tandis que chaque autre mot appartenant à un support de concept est associé avec le symbole de concept, il contribue au support. Le résultat de cette association est indiqué comme :  $[cw]_1^N$ . La structure de frames suivante représente la structure sémantique ex-

primée dans l'exemple 6.4.9.

$$\Gamma : \left( \begin{array}{l} \text{RESERVE}.[\text{reserve\_theme REFERENCE}. \\ [\text{referred\_object } x_1 / \lambda_j(x_1, x_2)]] \end{array} \right) \quad (6.3)$$

Les noms de frames sont en lettres majuscules. Les emplacements de frames sont représentés entre crochets. Cette structure est mise en hypothèse à partir de  $[cw]_1^N$ . Les instances de frames sont générées à l'aide d'un tableau de correspondance à partir d'étiquettes de concepts et composés en utilisant deux règles de composition de frames générales. La fonction  $\lambda_j(x_1^L)$  dans cet exemple est la lambda-expression suivante :

$$\lambda_j(x_1, x_2) : \left( \begin{array}{l} \lambda(x_1).\lambda(x_2).\text{is\_a}(x_1, \mathcal{E}) \wedge \text{is\_a}(x_2, \text{set}) \wedge \\ \text{el\_of}(x_1, x_2) \wedge \text{rank\_of}(x_1, 2) \wedge \\ \text{contains}(\mathcal{E}, \text{HOTEL}). \\ [\text{at\_loc ADDRESS}.[\text{city Lyon}]] \end{array} \right). \quad (6.4)$$

où  $\mathcal{E}$  est le type d'informations dans l'historique du dialogue qui doit être récupéré par une action représentée par la lambda-expression. Dans l'exemple 6.4.9  $\mathcal{E}$  est le type de  $\Psi$ . La structure de frame est une instance de la frame RESERVE dont le nom est la tête de la structure. Les emplacements de cette instance apparaissent dans la liste d'emplacements entre crochets. Les éléments de la liste sont des paires d'attributs / valeurs séparées par une virgule. Les valeurs peuvent être d'autres instances de frames. Le contenu de  $\mathcal{E}$  indiqué dans le prédicat *contains* est *chaîne d'emplacements*.

Le système de C.L.P. effectue un ensemble d'actions de compréhension dont l'une exécute  $\lambda_j(x_1, x_2)$  pour récupérer  $\Psi$ . Les actions de compréhension sont aussi effectuées quand le tour courant comporte des informations sémantiques qui doivent être composées avec la structure  $\Psi$  à récupérer.

Les types d'expressions faisant référence prises en compte dans les expériences décrites dans cette sections sont des références spécifiques décrites par les lambda-expressions qui pourraient être composés de balises de concepts et le sens de certains mots qui ne sont pas utilisés pour l'hypothèses des balises. Le but est de dériver des modèles de motifs pour obtenir directement des hypothèses d'expressions sans effectuer de compositions. Les principales motivations pour étudier cette possibilité sont que souvent, dans les dialogues parlés, certains types de références sont exprimés par des phrases courtes dans des contextes spécifiques à partir desquels l'humain peut en déduire le sens exact même en présence de la parole et d'erreurs de disfluences dues à la R.A.P. .

Après une analyse attentive du corpus, les lambda-expressions de référence sont listées dans le tableau 6.10 ont été obtenues. Les types de valeurs de prédicats sont indiqués entre parenthèses. Les variables représentant les arguments de prédicats doivent être des instances de classes. N\_OBJECT indique tous les types d'objets de négociation, comme RESERVATION, PRICE n'est pas inclus dans N\_OBJECT car les cas de prix sont simplement des valeurs des propriétés d'enregistrement. Le mode d'une phrase est une balise annotation supplémentaire prévue dans le corpus MEDIA .

$\lambda_n$	description	lambda-expression
$\lambda_1$	référence générique - declarative	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge mode(declarative)$
$\lambda_2$	référence générique - question	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge mode(question)$
$\lambda_3$	cout d'une référence	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge cost\_of(x, [PRICE])$
$\lambda_4$	cout unitaire d'une référence	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge unit\_cost\_of(x, [PRICE])$
$\lambda_5$	durée d'une référence	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge duration\_of(x, [DURATION])$
$\lambda_6$	référence d'éléments génériques d'un ensemble	$\lambda(x).\lambda(y).\{rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge is\_a(y, set) \wedge size\_of(y, [CARD\_NUMBER]) \wedge el\_of(x, y)\}$
$\lambda_7$	référence d'un objet ayant une propriété	$\lambda(x).rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge has\_property[PROPERTY]$
$\lambda_8$	référence d'éléments spécifiques d'un ensemble	$\lambda(x).\lambda(y).\{rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge is\_a(y, set) \wedge size\_of(y, [CARD\_NUMBER]) \wedge ref\_position(x, [ORD\_NUMBER]) \wedge el\_of(x, y)\}$
$\lambda_9$	référence sur plus d'éléments d'un ensemble	$\lambda(x).\lambda(y)..\{rfrd\_obj(x) \wedge is\_a(x, [N\_OBJECT]) \wedge is\_a(y, set) \wedge size\_of(y, [CARD\_NUMBER]) \wedge nb\_obj(x, [CARD\_NUMBER]) \wedge el\_of(x, y)\}$

TABLE 6.10 – Liste des types d'expressions de références

On suppose que la phrase de l'exemple 6.4.9 continue en demandant la réservation de cinq chambres. Cette information est composée avec  $\Psi$  par une action de compréhension exécutant la fonction de composition suivante :

$$UA : compose(head(\Psi).\sigma.ROOM, ROOM.[lodg - qty five]) \quad (6.5)$$

$head(\Psi).\sigma.ROOM$  et  $\sigma$  représentent des chaînes d'emplacements (slot-chains) comme décrit dans la section 5.1. Neuf lambda-fonctions ont été introduites basées sur l'ontologie d'application (Voir section A.2), pour exprimer les relations de propriétés de temps, d'espace, d'équipements, de prix sur des objets référencés.

### 6.4.1 Calcul des probabilités

Etant donné une séquence d'hypothèses de mots  $W$  et la séquence d'hypothèses de concepts  $C$  correspondante, un modèle graphique peut être considéré dans lequel les étiquettes de concepts et les mots supportent la génération des hypothèses de  $\lambda_j$  comme montré dans la figure 6.15. Le symbole  $\lambda_j$  est l'étiquette de sortie du nœud du graphe. La fonction  $\lambda_j$  est associée à l'emplacement *referred\_object* dans l'instance de frame REFERENCE de  $\Gamma$ .

FIGURE 6.15 – Représentation d'une relation et de son support.

Les étiquettes de concepts  $C$  et les structures  $\Gamma$  sont mises en hypothèses basées sur des caractéristiques  $F_i$ . Les concepts  $C$  contribuent avec  $W$  à déterminer les caractéristiques  $F_j$  utilisées pour mettre en hypothèses  $\lambda_j$  qui permet de récupérer  $\Psi$ . En se basant sur cette structure, les modèles peuvent être traités comme des champs de Markov aléatoires [Boykov et al. \(2001\)](#). Un symbole  $\lambda_n$  est associé à chaque élément de la séquence  $[cw]_1^N$ . Un symbole spécial  $\lambda_{null}$  est associé avec le  $n^{ième}$  élément de la séquence si cet élément ne contribue pas à former l'hypothèse d'une des lambda-fonctions. Les hypothèses sur les expressions d'une séquence de symboles  $\lambda_1^N$  sont calculées avec la probabilité suivante :

$$P(\lambda_1^N|Y) = \sum_{W,C} P(\lambda_1^N, C, W|Y) \approx \sum_{W,C} P(\lambda_1^N|C, W)P(C, W|Y) \quad (6.6)$$

$[cw]_1^N$  a été représenté comme  $CW$  et  $Y$  est la séquence des vecteurs de caractéristiques acoustiques décrivant un tour de dialogue parlé. La probabilité  $P(C, W|Y)$  est calculée comme décrit dans [Raymond et al. \(2006\)](#), et  $P(\lambda_1^N|C, W)$  est calculée comme suit :

$$P(\lambda_1^N|C, W) = \frac{1}{Z_{\lambda_1^N}} \exp \left\{ \sum_{n=1}^N \sum_{q=1}^Q \gamma_q g_q(\lambda_n, C, W) \right\} \quad (6.7)$$

Chaque fonction  $g_q(\cdot)$  est pondérée par un coefficient  $\gamma_q$ . Les fonctions  $g_q(\cdot)$  sont des fonctions binaires du type  $g_q(\lambda_n, v_{n-2}^{n+2})$ , exprimant le fait que les caractéristiques  $v_{n-2}^{n+2}$  introduites dans la sous-section [6.4.2](#), indiquent la présence, dans l'énoncé, d'une référence décrite par une lambda-expression  $\lambda_n$ .

La somme dans l'équation [6.6](#) peut être faite sur les  $N$ -meilleures séquences de mots et d'hypothèses d'étiquettes de concepts [Raymond et al. \(2006\)](#) ou plus simplement sur la meilleure séquence de mots et d'étiquettes de concepts. Pour l'exemple [6.4.9](#), la probabilité  $P(\lambda_1^N|Y)$  est le score de la relation entre *reserve\_theme* de l'instance de frame RESERVE et la structure  $\Psi$  qui sera récupérée. Les hypothèses des lambda-expressions sont calculées avec des probabilités dont la précision dépend des fonctions caractéristiques utilisées.

## 6.4.2 Les fonctions caractéristiques

Les lambda-expressions expriment des relations entre des entités sémantiques et des étiquettes de concepts qui peuvent être d'importants résumés de structures syntaxiques sémantiquement pertinentes. Cela suggère de composer manuellement un ensemble de modèles de motifs principalement en utilisant les étiquettes de concepts et des mots "fonctions" exprimant des relations syntaxiques parmi les supports de concepts. Les modèles ont été obtenus en utilisant des connaissances linguistiques pour la généralisation d'exemples de l'ensemble d'entraînement. Ces exemples sont automatiquement sélectionnés en utilisant des préconditions faites à partir de mots annotés et d'étiquettes de concepts disponibles dans le corpus. L'ensemble de développement a été utilisé pour raffiner la composition des motifs.

Les fonctions d'un des modèles conditionnels pour la mise en hypothèse des lambda-expressions renvoient la valeur "1" pour  $\lambda_j$  à la position "n" seulement si un des modèles de motifs pour  $\lambda_j$  correspond à CW et la correspondance inclut  $w_n$  ou ses étiquettes associées. Une fonction  $g_q(\lambda_n, C, W)$  effectuant la correspondance est construite à cet effet. D'autres fonctions construites automatiquement pour la correspondance de mots et de concepts n-grammes ont été considérées. Du fait de la quantité de données disponibles pour l'entraînement des modèles, l'utilisation de n-grammes et de fonctions de recherche de correspondance avec de multiples flots de données d'entrées, n'a pas produit de bons résultats (voir la sous-section 6.4.3). Au contraire des résultats intéressants ont été obtenus en représentant les modèles sur un seul flot de données d'entrée de symboles  $v_n$  obtenus comme suit :

$$v_n = \begin{cases} \mu_j & \text{si } m(j, n) = 1 \\ w_n & \text{sinon} \end{cases} \quad (6.8)$$

La variable  $\mu_j$  est introduite pour supposer des valeurs dépendant de la véracité du prédicat  $m(j, n)$  qui est vrai si le  $n^{\text{ième}}$  symbole d'entrée est une partie de la correspondance avec n'importe quel modèle de motifs supportant  $\lambda_j$ . Le concept "null" n'est pas considéré. La fonction  $g_q(\cdot)$  utilisée dans l'équation 6.7 est  $g_q(\lambda_n, v_{n-2}^{n+2})$ . Les résultats obtenus avec ces modèles sont indiqués comme des *flots composés*. Ajouter des concepts aux flots ne change pas les résultats.  $v_n = w_n$  et la définition suivante ont été considérées :

$$v_n = \begin{cases} c_n & \text{si } c_n \neq \text{null} \\ w_n & \text{sinon} \end{cases} \quad (6.9)$$

$c_n$  est une étiquette de concepts duquel le  $n^{\text{ième}}$  mot dans l'énoncé est une partie du support. Les valeurs de  $v_n$  sont calculées avant l'évaluation des fonctions  $g_q(\lambda_n, v_{n-2}^{n+2})$ . Les résultats obtenus avec ces modèles sont respectivement indiqués comme *n-grammes de mots*, et *n-grammes de concepts*.

### 6.4.3 Expérimentations

Pour rappel le corpus MEDIA utilisé pour les expériences est fait de 1250 dialogues, dont 208 sont dans le corpus de test. Les dialogues sont transcrits et annotés avec 83 étiquettes de concepts de base traduisant les actes de langage, des entités, des propriétés et des références génériques. L'annotation a été augmentée dans le projet LUNA pour contenir des structures de frames décrivant les entités et les propriétés exprimées à chaque tour de dialogue. Le corpus est divisé en trois parties contenant des données issues de trois séries différentes de locuteurs. L'annotation existante a été enrichie avec l'annotation en termes de lambda-expressions explicitement pour ces expériences. Une procédure semi-automatique a été utilisée pour annoter le corpus d'entraînement et de développement alors que le corpus de test a été annoté manuellement. La composition du corpus pour les expériences décrites ici est montrée dans le tableau 6.11 avec le nombre de tours de dialogue exprimant lambda-expression entre parenthèses.

**TABLE 6.11** – Composition du corpus *MEDIA* avec le nombre de tours de dialogue exprimant les *lambda-expression* entre parenthèses.

train	development	test
12908 (2339)	1259 (274)	3518 (701)

Le système de Reconnaissance Automatique de la Parole utilisé pour ces expériences est décrit dans [Barrault et al. \(2008\)](#). Les résultats expérimentaux en termes de précision et rappel sont résumés dans les tableaux suivants 6.12 et 6.13 pour le corpus de tests. Ils sont obtenus en utilisant les fonctions introduites précédemment avec des *C.R.F.* entraînés respectivement sur des données manuellement annotées sur le corpus d’entraînement (Tableau 6.12) et sur la meilleure hypothèse de séquence de mots obtenue avec un système de *R.A.P.* sur le corpus d’entraînement (Tableau 6.13)

Les résultats ne changent pas si le seuil d’acceptation des probabilité calculées par les *C.R.F.* varient entre 0.5 et 0.8. Les résultats montrent que l’utilisation des modèles de motifs introduisent une amélioration significative sur l’utilisation des *n-grammes* de mots et des *n-grammes* de mots / concepts. Cela indique que certaines relations sémantiques sont bien supportées par les compositions d’hypothèses de composants sémantiques et de mots exprimant des relations syntaxiques parmi les supports de concepts. L’entraînement avec des annotations manuellement annotées fournit les meilleures *F-mesures*.

**TABLE 6.12** – Résultats expérimentaux en termes de précision, rappel et *F-mesure* pour le corpus de test obtenu avec différents types de fonctions *C.R.F.* par entraînement des *C.R.F.* en utilisant le corpus d’entraînement annoté manuellement.

fonctions <i>C.R.F.</i>	Rappel	Precision	F-mesure
<i>mots n-grammes</i>	0.79	0.45	0.57
<i>mots-concepts n-grammes</i>	0.77	0.38	0.51
<i>motifs seulement</i>	<b>0.83</b>	0.83	0.83
<i>flot composé</i>	0.82	0.84	0.83

**TABLE 6.13** – Résultats expérimentaux en termes de précision, rappel et *F-mesure* pour le corpus de test obtenu avec différents types de fonctions *C.R.F.* par entraînement des *C.R.F.* en utilisant le corpus d’entraînement obtenu par un système *R.A.P.* .

fonctions <i>C.R.F.</i>	Rappel	Precision	F-mesure
<i>mots n-grammes</i>	0.75	0.51	0.61
<i>mots-concepts n-grammes</i>	0.77	0.45	0.57
<i>motifs seulement</i>	0.74	0.87	0.80
<i>flot composé</i>	0.76	<b>0.89</b>	0.82

Les erreurs de l’ensemble de test ont été analysées. Quatre types d’erreurs ont été identifiées. Chaque type d’erreur est causé par les limites d’un élément spécifique du système de *C.L.P.* et de *R.A.P.* .



Le premier type d'erreurs est causé par le détecteur d'activité vocale en coupant une partie substantielle d'un tour. Par conséquent, aucune étiquette de concepts n'a été émise en hypothèse par la composante de Compréhension du Langage Parlé . Cette erreur a été observée dans 17 % des cas d'erreurs de suppression.

Le deuxième type d'erreur est causé par des erreurs de reconnaissance des mots. La plupart d'entre elles est due à trop peu de phonèmes ou simplement de confusion homophone. Le remède consisterait à améliorer les modèles acoustiques et à enrichir le modèle de langage avec des structures de la langue plus grandes que des tri-grammes, entre autres, un accord entre le genre et le nombre du sujet, du verbe, des compléments et des adjectifs. Ce type d'erreurs a été observé dans 38 % des cas d'erreurs de suppression. Notez que les expressions de références décrivent un type de relations sémantiques représentées dans le langage naturel par des modèles spécifiques de motifs.

Le troisième type d'erreurs est dû aux limites de la connaissance *C.L.P.* . Elles représentent 26 % des cas d'erreurs de suppression. Les erreurs restantes sont dues à des erreurs mineures de *R.A.P.* .

Presque toutes les erreurs d'insertion sont dues à des erreurs *R.A.P.* . Parmi elles, 4% sont des références génériques en remplacement de références spécifiques. Cela est dû à des erreurs du système *R.A.P.* sur les mots monosyllabiques , 20 % sont des insertions que les gestionnaires de dialogue peuvent facilement ignorer, car il n'y a pas d'éléments correspondants dans le contexte du dialogue et les autres sont des erreurs importantes en raison de la non reconnaissance de plus d'une syllabe. Beaucoup d'erreurs d'insertion pourraient probablement être évitées par l'adaptation des modèles acoustiques et de langage avec plus de données.

Les tableaux suivants montrent d'autres résultats en détail sur différents critères d'utilisation et de composition des lambda-expressions pour la détection de fonctions logiques. Ces résultats sont composés en deux grandes parties, d'une part les *C.R.F.* sont appris sur le corpus d'entraînement annoté manuellement, d'autre part, les *C.R.F.* sont appris sur le corpus d'entraînement issu d'un système *R.A.P.* . Chaque système d'apprentissage est évalué sur les différents corpus : les lots "train", "dev", "test". Les différentes évaluations portent sur les fonctions *C.R.F.* , les motifs utilisés, les processus post-traitements de motifs et un entraînement particulier.

Les fonctions caractéristiques décrites ici sont pour la plupart les mêmes que celles décrites dans la section 6.4.2. Des changements mineurs sont détaillés plus bas.

Le *flot composé*  $W,C,T$  décrit dans l'équation 6.10 est une variante de la fonction caractéristique décrite dans l'équation 6.9.

$$v_n = \begin{cases} \mu_j & \text{if } m(j, n) = 1 \\ c_n & \text{if } c_n \neq \text{null} \wedge m(j, n) = 0 \\ w_n & \text{otherwise} \end{cases} \quad (6.10)$$

Le *flot composé*  $W,T$  est décrit dans l'équation 6.8. Les fonctions caractéristiques *mots n-grammes* sont les suites de mots sans  $\mu_j$ . Les fonctions *labels n-grammes* sont des suites de labels  $T : \mu_j$  sans mots ni concepts. Les fonctions *mots-concepts-labels n-grammes* sont



des suites de mots, pour chaque groupe de mots, on y ajoute le concept correspondant ainsi que  $T : \mu_j$  si  $m(j, n) = 1$ .

Des variations sont apportées aux motifs utilisés pour l'apprentissage. Les modèles de motifs utilisés dans l'apprentissage sont différents de ceux utilisés pour la référence puisque certains modèles observés dans les données d'entraînement issues de la R.A.P. et annotés manuellement, pour un même  $\mu_j$ , ne sont pas applicables réciproquement, du fait d'un trop grand nombre d'erreurs de la R.A.P. . Les motifs utilisés pour la référence sont appelés  $A_x$ . Pour certaines évaluations dans les données d'entraînements, une combinaison des motifs usuels  $\mu_j$  et des motifs  $A_x$  sont utilisés et notés dans les tableaux sous la forme "+ $A_x$ ".

Une autre variation de l'évaluation consiste à appliquer un traitement post-processus à l'étiquetage, qui consiste à forcer les labels  $A_x$  dans les tours où le processus normal n'a pas trouvé de labels  $\mu_j$ .

Enfin, une variation affectant l'entraînement normal dit "V0" est envisagée. La variation "V1" modifie l'entraînement pour réutiliser  $A_x$  quand il n'a pas trouvé de cas de labels  $\mu_j$ . La variation "V2" modifie l'entraînement pour réutiliser  $A_x$  quand il n'a pas trouvé de cas de labels  $\mu_j$  et quand il trouve un label et que le label  $A_x$  n'est pas en référence alors on remplace le label du tour par la version sans tag du tour.

Les tableaux suivants présentent ces résultats avec les mesures de Précision (Prec.), Rappel (Rap.), et F-mesure (F-mes.). Les tableaux 6.14, 6.15 et 6.16 présentent respectivement les résultats du corpus de développement, d'entraînement et de test avec un entraînement à partir du corpus annoté par un système de R.A.P. . Les tableaux 6.17, 6.18, 6.19 sont respectivement les tableaux de résultats sur les corpus issus d'un système de R.A.P. , de développement, d'entraînement et de test, à partir du corpus d'entraînement annoté manuellement en référence.

**TABLE 6.14** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de développement issu d'un système de R.A.P. , avec un corpus d'entraînement issu d'un système de R.A.P. .

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,89	0,73	0,80
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,89	0,71	0,79
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,89	0,71	0,79
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,71	0,74	0,72
<i>flot composé W,C,T</i>		+ Ax	V2	0,68	0,78	0,73
<i>flot composé W,T</i>			V0	0,88	0,73	0,80
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,88	0,72	0,79
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,88	0,72	0,79
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,70	0,73	0,72
<i>mots n-grammes</i>			V0	0,47	0,66	0,55
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,47	0,66	0,55
<i>labels n-grammes</i>			V0	0,89	0,72	0,80
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,91	0,70	0,79
<i>mots-concepts n-grammes</i>			V0	0,45	0,69	0,55
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,45	0,69	0,55
<i>mots-concepts-labels n-grammes</i>			V0	0,69	0,76	0,72
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,69	0,75	0,72

**TABLE 6.15** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus d'entraînement issu d'un système de R.A.P. , avec un corpus d'entraînement issu d'un système de R.A.P. .

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,95	0,83	0,88
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,95	0,80	0,87
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,95	0,80	0,87
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,77	0,83	0,80
<i>flot composé W,C,T</i>		+ Ax	V2	0,78	0,87	0,82
<i>flot composé W,T</i>			V0	0,95	0,83	0,88
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,95	0,80	0,87
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,95	0,80	0,87
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,77	0,83	0,80
<i>mots n-grammes</i>			V0	0,63	0,82	0,71
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,63	0,82	0,71
<i>labels n-grammes</i>			V0	0,95	0,79	0,86
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,95	0,77	0,85
<i>mots-concepts n-grammes</i>			V0	0,57	0,84	0,68
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,57	0,84	0,68
<i>mots-concepts-labels n-grammes</i>			V0	0,85	0,86	0,85
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,85	0,86	0,85

**TABLE 6.16** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de test issu d'un système de R.A.P. , avec un corpus d'entraînement issu d'un système de R.A.P.

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,86	0,77	0,81
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,87	0,75	0,81
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,87	0,75	0,81
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,70	0,79	0,74
<i>flot composé W,C,T</i>		+ Ax	V2	0,69	0,82	0,75
<i>flot composé W,T</i>			V0	0,88	0,77	0,82
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,89	0,75	0,81
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,89	0,75	0,81
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,70	0,78	0,74
<i>mots n-grammes</i>			V0	0,50	0,75	0,60
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,50	0,75	0,60
<i>labels n-grammes</i>			V0	0,88	0,73	0,80
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,89	0,73	0,80
<i>mots-concepts n-grammes</i>			V0	0,45	0,77	0,57
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,45	0,77	0,57
<i>mots-concepts-labels n-grammes</i>			V0	0,72	0,78	0,75
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,73	0,78	0,75

**TABLE 6.17** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de développement issu d'un système de R.A.P. , avec un entraînement annoté manuellement

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,80	0,77	0,78
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,77	0,71	0,74
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,77	0,71	0,74
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,56	0,76	0,65
<i>flot composé W,C,T</i>		+ Ax	V2	0,52	0,82	0,63
<i>flot composé W,T</i>			V0	0,82	0,76	0,79
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,77	0,71	0,74
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,77	0,71	0,74
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,57	0,76	0,65
<i>mots n-grammes</i>			V0	0,42	0,73	0,53
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,42	0,73	0,53
<i>labels n-grammes</i>			V0	0,82	0,74	0,78
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,71	0,73	0,72
<i>mots-concepts n-grammes</i>			V0	0,35	0,72	0,47
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,35	0,72	0,47
<i>mots-concepts-labels n-grammes</i>			V0	0,59	0,76	0,66
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,54	0,74	0,63

**TABLE 6.18** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus d'entraînement issu d'un système de R.A.P. , avec corpus d'un entraînement annoté manuellement

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,85	0,88	0,87
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,85	0,83	0,84
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,85	0,83	0,84
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,66	0,84	0,74
<i>flot composé W,C,T</i>		+ Ax	V2	0,61	0,90	0,73
<i>flot composé W,T</i>			V0	0,89	0,87	0,88
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,87	0,82	0,84
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,87	0,82	0,84
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,66	0,84	0,74
<i>mots n-grammes</i>			V0	0,54	0,86	0,67
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,54	0,86	0,67
<i>labels n-grammes</i>			V0	0,90	0,83	0,86
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,82	0,83	0,82
<i>mots-concepts n-grammes</i>			V0	0,44	0,82	0,58
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,44	0,82	0,58
<i>mots-concepts-labels n-grammes</i>			V0	0,72	0,86	0,78
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,67	0,84	0,74

**TABLE 6.19** – Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de test issu d'un système de R.A.P. , avec corpus d'un entraînement annoté manuellement

Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
<i>flot composé W,C,T</i>			V0	0,78	0,82	0,80
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V0	0,79	0,78	0,78
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V1	0,79	0,78	0,78
<i>flot composé W,C,T</i>	+ Ax	+ Ax	V2	0,59	0,81	0,68
<i>flot composé W,C,T</i>		+ Ax	V2	0,54	0,86	0,66
<i>flot composé W,T</i>			V0	0,80	0,82	0,81
<i>flot composé W,T</i>	+ Ax	+ Ax	V0	0,80	0,77	0,79
<i>flot composé W,T</i>	+ Ax	+ Ax	V1	0,80	0,77	0,79
<i>flot composé W,T</i>	+ Ax	+ Ax	V2	0,59	0,81	0,68
<i>mots n-grammes</i>			V0	0,45	0,79	0,57
<i>mots n-grammes</i>	+ Ax	+ Ax	V0	0,45	0,79	0,57
<i>labels n-grammes</i>			V0	0,82	0,77	0,80
<i>labels n-grammes</i>	+ Ax	+ Ax	V0	0,78	0,77	0,78
<i>mots-concepts n-grammes</i>			V0	0,38	0,77	0,51
<i>mots-concepts n-grammes</i>	+ Ax	+ Ax	V0	0,38	0,77	0,51
<i>mots-concepts-labels n-grammes</i>			V0	0,62	0,80	0,70
<i>mots-concepts-labels n-grammes</i>	+ Ax	+ Ax	V0	0,58	0,80	0,67

En observant les résultats, on s'aperçoit qu'un apprentissage par corpus issu d'un système *R.A.P.* donne de meilleurs résultats en F-mesure, et en Précision. L'apprentissage par corpus issu d'un système de *R.A.P.*, va trouver plus facilement les fonctions caractéristiques, puisqu'il pourra mieux reconnaître les motifs appris. Cependant en règle générale, le Rappel est plus faible, car les motifs de la référence ne seront pas tous reconnus du fait des erreurs liées à la Reconnaissance Automatique de la Parole. Un système apprenant en même temps avec un corpus issu de la *R.A.P.* et un corpus annoté manuellement a été envisagé sur les fonctions caractéristiques en *flot composé W,C,T* et donne des résultats intermédiaires montrés dans le tableau 6.20. Le choix de ce test a été de déterminer si un apprentissage combiné pouvait améliorer de manière significative les résultats par rapport à un entraînement non combiné.

**TABLE 6.20** – Résultats des fonctions caractéristiques en *flot composé W,C,T* sur les corpus de *dev,train* et *test* issu d'un système de *R.A.P.*, avec un corpus d'entraînement annoté manuellement combiné à entraînement issu d'un système de *R.A.P.*

Corpus	Fonctions C.R.F.	Motif	Post traitement	Entrain- nement	Prec.	Rap.	F-mes.
dev	<i>flot composé W,C,T</i>		+ Ax	V2	0,56	0,82	0,67
train	<i>flot composé W,C,T</i>		+ Ax	V2	0,66	0,89	0,76
test	<i>flot composé W,C,T</i>		+ Ax	V2	0,58	0,85	0,69

Par ailleurs, l'utilisation d'étiquettes dans les fonctions caractéristiques améliore le système de façon significative. Les étiquettes jouent le rôle de point d'appui du support des références. Elles permettent de faire concentrer l'information importante en se basant sur la sortie de la Reconnaissance Automatique de la Parole. Les sorties ne sont pas toujours correctes grammaticalement ou contiennent un ensemble de mots hors vocabulaire. L'utilisation des labels pour l'apprentissage permet de recentrer l'information portée sur les références, et de pointer lors de l'apprentissage les supports adéquats. L'utilisation combinée de mots et d'étiquettes est une des meilleures fonctions caractéristiques employées, l'utilisation des concepts dans cette combinaison, n'apporte aucune amélioration.

# Conclusions et Perspectives

## 6.5 Conclusions

La tâche d'interprétation de la langue naturelle est complexe et difficile. Deux types d'analyseurs ont été élaborés pour l'automatisation de cette tâche par un ordinateur. L'analyseur profond qui se base sur une syntaxe rigoureuse et une grammaire du langage complètes. L'analyseur superficiel qui prend en compte des parties de la phrase pour établir des modèles d'interprétation, en se basant sur des grammaires segmentées.

L'utilisation des analyseurs syntaxiques profonds, semble incompatible pour répondre à la problématique de la compréhension et de l'interprétation de la langue naturelle orale, puisque beaucoup de paramètres extérieurs viennent perturber la bonne détection des mots dans la Reconnaissance Automatique de la Parole. De plus la langue orale n'est pas ou peu structurée comme la langue naturelle écrite, les analyseurs profonds ne peuvent donc pas fonctionner correctement sur des phrases orales.

La représentation sémantique en frames est utilisée car elle permet une utilisation partielle des connaissances pour l'inférence et la composition sémantique dans les énoncés oraux. Des processus d'attachements, de fusion et d'inférence de frames ont été mis en place à cet effet.

Nous avons donc considéré les approches dites par analyses de surface et utilisé les segmenteurs, classificateurs et étiqueteurs, pour introduire nos approches sur la composition sémantique, l'acquisition de support pour la composition, la désambiguïsation de concepts et la détection de lambda-expressions pour la détection des références.

- Un formalisme de représentation des connaissances pour la compréhension de la parole a été introduit. Il a été utilisé pour une annotation incrémentale et automatique du corpus MEDIA en termes des structures sémantiques. Les annotations automatiques ont été évaluées et soumises à des experts humains quand la confiance était faible. Un processus d'annotation automatique a été introduit pour composer des structures sémantiques à partir d'éléments sémantiques basiques, en utilisant des motifs impliquant des constituants et des mots. Le traitement utilise des procédures pour obtenir des compositions sémantiques et pour générer les hypothèses de frames par inférences.

Les résultats que nous présentons, montrent que la source de connaissance et les procédures attachés ont de bonnes capacités à produire des hypothèses de

structures sémantiques. Ces travaux de recherche pourraient être poursuivis en utilisant des structures sémantiques pour sélectionner les possibles composants sémantiques au-delà de la 1-meilleure interprétation dans tout le treillis d'hypothèses de concepts.

- Une procédure incrémentale basée sur un apprentissage actif et exploratoire a été proposée, pour l'annotation supervisée de tours de dialogues. Une nouvelle méthode pour la composition de structures sémantiques partielles, a été introduite avec la possibilité d'effectuer des inférences logiques et les inférences possibles peuvent être confirmées par la détection de constituants sémantiques dans une liste structurée. Des améliorations significatives ont été observées avec les deux types d'inférences.

- Une approche a été introduite pour utiliser une ontologie et inférer des éléments sémantiques non hypothésisés par un système de Reconnaissance Automatique de la Parole. Celui-ci génère des transcriptions automatiques à partir de la langue naturelle parlée. Ces transcriptions sont affectées par des erreurs dues aux limitations du module de [R.A.P.](#) . Pour cette raison, il est proposé de considérer des solutions spécifiques pour l'interprétation sémantique sur des séquences de mots corrompues. Comme des hypothèses de base des constituants sémantiques sont disponibles, les relations ont été étudiées entre les mots, les concepts et les fragments dérivés de ceux-ci.

Une ontologie du domaine réalisée par un expert, détermine les relations sémantiques autorisées et constitue une base de connaissances qui nous permet de définir les liens logiques qui existent entre les frames. Une méthode de désambiguïsation des concepts basiques a été proposée avec la possibilité d'effectuer des inférences logiques et des déductions possibles pouvant hypothésiser la présence de concepts non reconnus ou ambigus. Des résultats prometteurs ont été observés avec ces types de déductions en comparaison avec des systèmes d'apprentissage. La recherche pourrait se poursuivre pour l'apprentissage et la combinaison de cette approche avec des outils tels que les [C.R.F.](#) pour prendre en charge les compositions plus complexes, et récupérer les erreurs liées à la [R.A.P.](#) en étudiant par exemple la couverture de ces deux systèmes.

- Une approche a été introduite pour générer les lambda-expressions décrivant des interprétations sémantiques dans des tours de dialogue parlé. Les résultats montrent des performances différentes entre tous les modèles conditionnels testés en termes de précision et de rappel. Les erreurs observées perturbent 6 % de l'ensemble du corpus.

Un gestionnaire de dialogues peut faire un bon usage des hypothèses sémantiques générées par les différents modèles en évaluant leur cohérence avec le contexte du dialogue disponible. Une analyse d'erreur a été effectuée en indiquant diverses sources d'erreurs provenant des différentes étapes du système : le système [R.A.P.](#) , le décodage et le processus d'interprétation.

La dérivation des modèles supportant des relations sémantiques complexes en



présence de bruit semble être un axe de recherche prometteur dans lequel les efforts de l'homme peuvent être combinés avec des méthodes automatiques. L'apprentissage discriminant peut être utilisé pour l'apprentissage des informations utiles pour être introduites avec la supervision de l'homme dans les modèles de motifs. Des expériences préliminaires ont indiqué que cela est prometteur, même si le principal problème semble venir du taux d'erreurs mots de la Reconnaissance Automatique de la Parole qui est de 28,1 ( $\pm 0.55$ )%.

Les erreurs liées au signal de parole, à la Reconnaissance Automatique de la Parole, engendrent un ensemble d'erreurs d'interprétation dans la composition sémantique et la compréhension de la parole. Ceci conduit à utiliser un ensemble d'approches basées sur l'apprentissage de données corrompues. Ces apprentissages permettent des améliorations significatives par rapport à des systèmes basés sur des grammaires strictes. L'utilisation des ontologies, notamment du domaine permet de contrôler et de restreindre les inférences aux relations possibles que l'on a dans l'énoncé. Cela demande un effort supplémentaire dans le premier stade de l'apprentissage mais, permet de gagner du temps dans la recherche d'erreurs car certaines sont évitées par le respect et la cohérence ontologique.

L'apprentissage et la combinaison des règles d'application ontologique, ainsi que la combinaison de systèmes, permettent une amélioration significative du système global de compréhension. Les systèmes lorsqu'ils ont une approche différentes pallient les manques des uns, mais peuvent engendrer des erreurs qui n'auraient pas été levées avec un système. Il est donc important de définir une stratégie dans la prise de décisions des systèmes afin d'avoir un taux d'erreurs minimal. Un indice de confiance est nécessaire pour cette stratégie.

La prise en compte de l'historique du dialogue pourrait permettre une meilleure récupération des erreurs, notamment pour les références et la détection des lambda-expressions pour les objets référencés.

## 6.6 Perspectives

Les travaux de cette thèse ne sont pas exhaustifs, mais ouvrent la voie vers d'autres possibilités d'appréhender la compréhension de la parole. Proposer une meilleure interaction entre l'homme et la machine passe par la compréhension de l'homme par la machine. Pour cela nous avons proposé des systèmes de détection de liens sémantiques qui existent dans la transcription du signal audio par une modélisation de la parole. Un système d'apprentissage d'erreurs permettrait de modéliser les erreurs de compréhension et de faciliter le retour des erreurs vers les systèmes de transcription de la parole. Cela pourrait pallier certaines erreurs dans le module de R.A.P. et permettre la désambiguïsation de certains mots par exemple.

L'utilisation de la modélisation des énoncés oraux pourrait permettre à des systèmes de ré-écrire les énoncés pour qu'ils soient syntaxiquement et grammaticalement corrects. Les outils de traitement de la langue naturelle pourraient apporter d'éventuelles



améliorations à la modélisation et confronter les hypothèses de transcriptions avec le contexte. Ainsi la mise en concurrence des analyseurs, permettrait de vérifier, la bonne modélisation de l'énoncé par la confrontation avec l'historique du dialogue et la cohérence de l'interprétation du dialogue.

Certaines expériences nous ont montré qu'une phrase pouvait être modélisée et interprétée de plusieurs manières différentes. La machine n'en accepte qu'une seule. Un indice de confiance pour l'acceptation d'une modélisation de phrase, étant donné son contexte et ses mots permettrait d'obtenir plusieurs modélisations acceptables et correctes pour cette même phrase. Plusieurs systèmes différents et plusieurs personnes pourraient dialoguer ainsi dans un contexte de "dialogue" multi-locuteur. Cela permettrait l'intégration d'un module d'interprétation dans un système multi-agents avec des agents cognitifs et communicants [Jonquet et Cerri \(2005\)](#); [Duvert et al. \(2006\)](#).

# Liste des illustrations

1.1	Schéma des architectures des premiers systèmes de C.L.P. . . . . .	20
1.2	Exemple de Grammaire à Réseaux de Transitions Augmentées . . . . .	28
2.1	Représentation des dérivations G.C.C. en arbres . . . . .	44
2.2	Représentation des dérivations G.C.C. en arbres . . . . .	45
2.3	Dérivation de "that IBM bought" . . . . .	46
2.4	Dérivation de "she bought and sold shares" . . . . .	46
2.5	Dérivation de "spent \$325,000 in 1989 and \$340,000 in 1990" . . . . .	46
2.6	Dérivation de la phrase "I offered and may give a flower to a policeman" . . . . .	47
2.7	Dérivation de la phrase "articles that I file without reading" . . . . .	48
2.8	Dérivation de la phrase "John buys shares" . . . . .	48
2.9	Exemple de phrase analysée avec un modèle à vecteurs d'états cachés . . . . .	53
3.1	Exemple de treillis de mots $G_W$ issu d'un module de reconnaissance de la parole . . . . .	64
3.2	Exemple de transducteur mot-concept $T_{WC}$ correspondant à la composition de l'automate $G_w$ avec le transducteur $T_{concept}$ . . . . .	64
3.3	Liste des $N$ meilleures interprétations conceptuelles basiques $I_i$ avec leur automate $G_{w_i}$ correspondant . . . . .	65
3.4	Exemple de modèles de D.B.N. . . . .	66
3.5	Exemple d'un arbre de classification sémantique S.C.T. . . . .	69
4.1	Schéma d'architecture de logiciel utilisé par le projet LUNA . . . . .	101
6.1	Etape de mise en hypothèses des fragments de frames à partir des concepts - 1 <sup>er</sup> partie de phrase . . . . .	120
6.2	Etape de mise en hypothèses des fragments de frames à partir des concepts - 2 <sup>ième</sup> partie de phrase . . . . .	120
6.3	Etape de mise en composition des hypothèses de fragments de frames . . . . .	121
6.4	F-mesure Oracle pour l'identification des frames instanciées sur les premières $n$ -meilleures séquences d'éléments conceptuels extraits à partir d'un treillis de décodage, en fonction de $n$ . . . . .	122

---

6.5	Exemple de fragments qui peuvent être obtenus avant inférence de niveau 0 . . . . .	126
6.6	Exemple de fragments composés par inférence de niveau 0 à partir de la figure 6.5 . . . . .	127
6.7	Exemple de fragments qui peuvent être obtenus avant inférence de niveau 0 . . . . .	127
6.8	Exemple de fragments composés par inférence de niveau 1 à partir de la figure 6.7 . . . . .	128
6.9	Représentation partielle de l'ontologie du domaine, les frames sont en majuscules et les emplacements ou éléments de frames sont en minuscules, sur les flèches. Celles-ci représentent les liens entre frames. Cette ontologie a été créée à partir de "Protégé 3.4.3" disponible sur <a href="http://protege.stanford.edu">http://protege.stanford.edu</a> . . . . .	135
6.10	Annotation de référence en frame basée sur la phrase de l'exemple 6.3.7	141
6.11	Etape de mise en hypothèses des fragments de frames à partir des concepts - 1 <sup>er</sup> partie de phrase . . . . .	143
6.12	Etape de mise en hypothèses des fragments de frames à partir des concepts - 2 <sup>ième</sup> partie de phrase . . . . .	143
6.13	Graphe des possibles associations à partir des fragments du tableau 6.8 .	144
6.14	Interprétation du tour de dialogue issu du système R.A.P. à partir des concepts et spécifieurs donnés dans le tableau 6.9 . . . . .	145
6.15	<i>Représentation d'une relation et de son support.</i> . . . . .	147
A.1	Représentation en frames d'une partie de la base de connaissance et sa projection de <i>FrameNet</i> vers <i>MEDIA</i> . . . . .	192
A.2	Représentation partielle de l'ontologie du domaine <i>MEDIA</i> , les frames sont en majuscules et les emplacements ou éléments de frames sont en minuscules, sur les flèches. Celles-ci représentent les liens entre frames. .	193
A.3	Représentation partielle de l'ontologie du domaine <i>MEDIA</i> , en particulier les actes de dialogues, les frames sont en majuscules et les emplacements ou éléments de frames sont en minuscules, sur les flèches. Celles-ci représentent les liens entre frames. . . . .	194
B.1	Exemple d'annotations en frame, illustrant le protocole d'évaluation . .	196
B.2	Evaluation de la métrique 1 : Reconnaissance de frames . . . . .	197
B.3	Evaluation de la métrique 2 : Détection des limites des arguments . . . .	197
B.4	Evaluation de la métrique 3 : Détection des étiquettes des arguments . .	197
B.5	Evaluation de la métrique 4 : Reconnaissance des arguments . . . . .	197
B.6	Evaluation de la métrique 5 : Réalisation des frames . . . . .	198
B.7	Evaluation de la métrique 6 : Composition des frames . . . . .	198
B.8	Evaluation de la métrique 7 : Analyse complète du tour . . . . .	198

# Liste des tableaux

4.1	Résultats de l'étiquetage en Concept à taux d'erreurs de concept ( C.E.R. ) obtenus avec différents modèles et méthodes sur le corpus français MEDIA . . . . .	102
6.1	Exemple d'annotation sémantique du corpus MEDIA . . . . .	117
6.2	Taux d'erreurs (ER) et Taux d'erreurs Oracle (OER) sur la $n$ -meilleure liste des interprétations pour les mots, les concepts, les concepts avec des spécifieurs et valeurs . . . . .	121
6.3	Tableau de correspondance entre concept et fragments de frames . . . . .	126
6.4	Pourcentage de tours correctement interprétés sur le corpus de test (2593 tours) . . . . .	133
6.5	Résultats du système à base de règles sur le corpus de test . . . . .	136
6.6	Résultats du système à base de C.R.F. sur le corpus de test . . . . .	136
6.7	Exemple de la transcription et des annotations manuelles d'un tour de dialogue du corpus MEDIA , en concepts, valeurs et spécifieurs . . . . .	139
6.8	Exemple de la transcription et d'annotations automatiques d'un tour de dialogue du corpus MEDIA , en concepts et valeurs . . . . .	140
6.9	Exemple de la transcription et d'annotations automatiques d'un tour de dialogue du corpus MEDIA , en concepts, valeurs et spécifieurs. . . . .	142
6.10	Liste des types d'expressions de références . . . . .	147
6.11	<i>Composition du corpus MEDIA avec le nombre de tours de dialogue exprimant les lambda-expression entre parenthèses.</i> . . . . .	150
6.12	Résultats expérimentaux en termes de précision, rappel et F-mesure pour le corpus de test obtenu avec différents types de fonctions C.R.F. par entraînement des C.R.F. en utilisant le corpus d'entraînement annoté manuellement. . . . .	150
6.13	Résultats expérimentaux en termes de précision, rappel et F-mesure pour le corpus de test obtenu avec différents types de fonctions C.R.F. par entraînement des C.R.F. en utilisant le corpus d'entraînement obtenu par un système R.A.P. . . . .	150

---

6.14	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de <b>développement</b> issu d'un système de R.A.P. , avec un corpus d' <b>entraînement issu d'un système de R.A.P.</b> . . . . .	153
6.15	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus d' <b>entraînement</b> issu d'un système de R.A.P. , avec un corpus d' <b>entraînement issu d'un système de R.A.P.</b> . . . . .	153
6.16	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de <b>test</b> issu d'un système de R.A.P. , avec un corpus d' <b>entraînement issu d'un système de R.A.P.</b> . . . . .	154
6.17	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de <b>développement</b> issu d'un système de R.A.P. , avec un <b>entraînement annoté manuellement</b> . . . . .	154
6.18	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus d' <b>entraînement</b> issu d'un système de R.A.P. , avec corpus d'un <b>entraînement annoté manuellement</b> . . . . .	155
6.19	Résultats sur les différentes variations des fonctions caractéristiques sur le corpus de <b>test</b> issu d'un système de R.A.P. , avec corpus d'un <b>entraînement annoté manuellement</b> . . . . .	155
6.20	Résultats des fonctions caractéristiques en <i>flot composé W,C,T</i> sur les corpus de <b>dev,train</b> et <b>test</b> issu d'un système de R.A.P. , avec un corpus d' <b>entraînement annoté manuellement</b> combiné à <b>entraînement issu d'un système de R.A.P.</b> . . . . .	156
A.1	Exemple de dialogue extrait du corpus <small>MEDIA</small> . . . . .	188
A.2	Nombre de dialogue par lot dans le corpus <small>MEDIA</small> . . . . .	188
A.3	Exemple de message annoté hors-contexte du corpus <small>MEDIA</small> . . . . .	189
A.4	Exemple de message annoté hors-contexte annoté en-contexte du corpus <small>MEDIA</small> sur le message « Est-il proche du centre-ville ? » énoncé après le prompt - « Je vous propose l'hôtel du Palais » . . . . .	190
A.5	Reprise de l'exemple de message annoté hors-contexte A.3, cette fois annoté en-contexte. . . . .	191
A.6	Exemple de facettes et d'éléments de descriptions des frames de l'ontologie <small>MEDIA</small> . . . . .	192
B.1	Les différentes métriques utilisées dans le protocole d'évaluation . . . . .	196

# Bibliographie

- (Abney, 1991) S. P. Abney, 1991. Parsing by chunks. Dans R. C. Berwick, S. P. Abney, et C. Tenny (Eds.), *Principle-Based Parsing : Computation and Psycholinguistics*, 257–278. Dordrecht : Kluwer Academic Publishers.
- (Abney et al., 1999) S. P. Abney, R. E. Schapire, et Y. Singer, 1999. Boosting applied to tagging and PP attachment. Dans les actes de *Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, Maryland, 38–45.
- (Ajdukiewicz, 1935) K. Ajdukiewicz, 1935. Die syntaktische konnexität. *Studia Philosophica* 1, 1–27.
- (Allen, 1988) J. F. Allen, 1988. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc.
- (Ang et al., 2005) J. Ang, Y. Liu, et E. Shriberg, 2005. Automatic dialog act segmentation and classification in multiparty meetings. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1061–1064.
- (Appelt et al., 1995) D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, et M. Tyson, 1995. SRI International FASTUS system MUC-6 test results and analysis. Dans les actes de *Message Understanding Conference*, San Francisco, 237–248. Morgan Kaufmann.
- (Appelt et Jackson, 1992) D. E. Appelt et E. Jackson, 1992. Sri international february 1992 atis benchmark test results. Dans les actes de *DARPA Workshop on Speech and Natural Language Processing*.
- (Argamon et al., 1998) S. Argamon, I. Dagan, et Y. Krymolowski, 1998. A memory-based approach to learning shallow natural language patterns. Dans les actes de *International Conference on Computational Linguistics*, 67–73. Association for Computational Linguistics.
- (Austin et al., 1991) S. Austin, R. Schwartz, et P. Placeway, 1991. The forward-backward search algorithm. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 1, 697–700. IEEE.
- (Baker et al., 1998) C. F. Baker, C. J. Fillmore, et J. B. Lowe, 1998. The Berkeley Frame-Net Project. Dans les actes de *International Conference on Computational Linguistics*, Morristown, NJ, USA, 86–90. Association for Computational Linguistics.

- (Bangalore et Johnston, 2004) S. Bangalore et M. Johnston, 2004. Balancing data-driven and rule-based approaches in the context of a multimodal conversational system. Dans D. M. Susan Dumais et S. Roukos (Eds.), *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Boston, Massachusetts, USA, 33–40. Association for Computational Linguistics.
- (Bangalore et Joshi, 1999) S. Bangalore et A. K. Joshi, 1999. Supertagging : an approach to almost parsing. *International Conference on Computational Linguistics* 25(2), 237–265.
- (Bar-Hillel, 1953) Y. Bar-Hillel, 1953. A quasi-arithmetical notation for syntactic description. *Language* 28, 47–58.
- (Barrault et al., 2008) L. Barrault, C. Servan, D. Matrouf, G. Linarès, et R. De Mori, 2008. Frame-based acoustic feature integration for speech understanding. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas.
- (Bates et al., 1993) M. Bates, R. Bobrow, P. Fung, R. Ingria, F. Kubala, J. Makhoul, L. Nguyen, R. Schwartz, et D. Stallard, 1993. The bbn/harc spoken language understanding system. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 2, Los Alamitos, CA, USA, 111–114. IEEE Computer Society.
- (Bates et al., 1994) M. Bates, R. Bobrow, R. Ingria, S. Peters, et D. Stallard, 1994. Advances in BBN’s Spoken Language System. Dans les actes de *IEEE workshop on Spoken Language Technology (SLT)*, 43–47.
- (Bellegarda, 2000) J. R. Bellegarda, 2000. Large vocabulary speech recognition with multispans statistical language models. *IEEE Transactions on Speech and Audio Processing* 8(1), 76–84.
- (Bender et al., 2003) O. Bender, K. Macherey, F. J. Och, et H. Ney, 2003. Comparison of alignment templates and maximum entropy models for natural language understanding. Dans N. U. Morristown (Ed.), *Annual Meeting of the European Association for Computational Linguistics (EACL)*, Budapest, Hungary, 11–18. Association for Computational Linguistics.
- (Berners-Lee et al., 2001) T. Berners-Lee, J. Hendler, et O. Lassila, 2001. The Semantic Web. *Scientific American* 284(5), 34–43.
- (Bikel et al., 1999) D. M. Bikel, R. Schwartz, et R. M. Weischedel, 1999. An algorithm that learns what’s in a name. *Machine Learning* 34(1-3), 211–231.
- (Bilmes et Kirchhoff, 2003) J. A. Bilmes et K. Kirchhoff, 2003. Factored language models and generalized parallel backoff. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Morristown, NJ, USA, 4–6. Association for Computational Linguistics.



- (Black et al., 1993) E. Black, F. Jelinek, J. Lafferty, D. M. Magerman, R. L. Mercer, et S. Roukos, 1993. Towards history-based grammars : using richer models for probabilistic parsing. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 31–37. Association for Computational Linguistics.
- (Bobrow et al., 1990) R. Bobrow, R. Ingria, et D. Stallard, 1990. Syntactic and semantic knowledge in the delphi unification grammar. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 230–236. Association for Computational Linguistics.
- (Bonneau-Maynard et al., 2005a) H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn, et D. Mostefa, 2005a. Semantic annotation of the french media dialog corpus. Dans les actes de *Interspeech*, Lisbon.
- (Bonneau-Maynard et al., 2005b) H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn, et D. Mostefa, 2005b. Semantic annotation of the french media dialog corpus. Dans les actes de *Eurospeech*, Lisbonne, Portugal.
- (Bottou, 1991a) L. Bottou, 1991a. Stochastic gradient learning in neural networks. Dans les actes de *Neuro-Nîmes 91*, Nîmes, France. EC2.
- (Boykov et al., 2001) Y. Boykov, O. Veksler, et R. Zabih, 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(11), 1222–1239.
- (Brachman et Schmolze, 1985) R. J. Brachman et J. G. Schmolze, 1985. An overview of the kl-one knowledge representation system. *Cognitive Science* 9(2), 171–216.
- (Brill, 1995) E. Brill, 1995. Transformation-based error-driven learning and natural language processing : a case study in part-of-speech tagging. *International Conference on Computational Linguistics* 21(4), 543–565.
- (Brown et al., 1993) P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, et R. L. Mercer, 1993. The mathematics of statistical machine translation : Parameter estimation. *International Conference on Computational Linguistics* 19(2), 263–311.
- (Camelin et al., 2010) N. Camelin, F. Béchet, G. Damnati, et R. De Mori, 2010. Detection and interpretation of opinion expressions in spoken surveys. *IEEE Transactions on Audio, Speech and Language Processing* 18(2), 369–381.
- (Camelin et al., 2006) N. Camelin, G. Damnati, F. Béchet, et R. De Mori, 2006. Opinion mining in a telephone survey corpus. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*.
- (Castellanos et al., 1998) A. Castellanos, E. Vidal, M. A. Varó, et J. Oncina, 1998. Language understanding and subsequential transducer learning. *Computer, Speech and Language* 12(3), 193–228.



- (Charniak, 2001) E. Charniak, 2001. Immediate-head parsing for language models. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 124–131. Association for Computational Linguistics.
- (Chelba et Jelinek, 2000) C. Chelba et F. Jelinek, 2000. Structured language modeling. *Computer, Speech and Language* 14(4), 283–332.
- (Chen et Rosenfeld, 1999) S. F. Chen et R. Rosenfeld, 1999. A gaussian prior for smoothing maximum entropy models. Rapport technique, CMU-CS-99-108, School of Computer Science Carnegie Mellon University, Pittsburgh, PA.
- (Chen et Rosenfeld, 2000) S. F. Chen et R. Rosenfeld, 2000. A survey of smoothing techniques for ME methods. *IEEE* 8(1), 37–50.
- (Chu-Carroll et Carpenter, 1999) J. Chu-Carroll et B. Carpenter, 1999. Vector-based natural language call routing. *International Conference on Computational Linguistics* 25(3), 361–388.
- (Cohen et Moore, 1994) M. Cohen et R. C. Moore, 1994. Combining linguistic and statistical technology for improved spoken language understanding. Dans les actes de *workshop on Human Language Technology (HLT)*, Morristown, NJ, USA, 472–472. Association for Computational Linguistics.
- (Collins et al., 2004) C. Collins, B. Carpenter, et G. Penn, 2004. Head-driven parsing for word lattices. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, 232–239.
- (Collins, 2002) M. Collins, 2002. Discriminative training methods for hidden markov models : theory and experiments with perceptron algorithms. Dans les actes de *conference on Empirical methods in natural language processing*, Morristown, NJ, USA, 1–8. Association for Computational Linguistics.
- (Collins, 2003) M. Collins, 2003. Head-driven statistical models for natural language parsing. *International Conference on Computational Linguistics* 29(4), 589–637.
- (Collobert et Bengio, 2001) R. Collobert et S. Bengio, 2001. Svmtorch : Support vector machines for large-scale regression problems. *Journal of Machine Learning Research* 1, 143–160.
- (Core et Allen, 1997) M. G. Core et J. F. Allen, 1997. Coding dialogues with the DAMSL annotation scheme. Dans D. Traum (Ed.), *Working Notes : AAAI Fall Symposium on Communicative Action in Humans and Machines*, Menlo Park, California, 28–35. American Association for Artificial Intelligence.
- (Curry et Feys, 1958) H. B. Curry et R. Feys, 1958. *Combinatory Logic*, Volume 1. North Holland.
- (Dahlgren, 1988) K. Dahlgren, 1988. *Naive Semantics for Natural Language Understanding*. Norwell, MA, USA : Kluwer Academic Publishers.

- (Damnati et al., 2007) G. Damnati, F. Béchet, et R. De Mori, 2007. Spoken language understanding strategies on the france telecom 3000 voice agency corpus. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaï.
- (Darroch et Ratcliff, 1972) J. N. Darroch et D. Ratcliff, 1972. Generalized iterative scaling for log-linear models. *The annals of Mathematical Statistics* 43(5), 1470–1480.
- (De Mori, 1998) R. De Mori, 1998. *Spoken Dialogues with computers*. Orlando, FL, USA : Academic Press, Inc.
- (De Mori et al., 2008) R. De Mori, F. Béchet, D. Hakkani-Tür, M. McTear, G. Riccardi, et G. Tür, 2008. Spoken language understanding : a survey. *IEEE Signal Processing Magazine* 25(3), 50–58.
- (De Mori et al., 1991) R. De Mori, R. Kuhn, et G. Lazzari, 1991. A probabilistic approach to person-robot dialogue. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 2, 797–800.
- (Della Pietra et al., 1997) S. A. Della Pietra, V. J. Della Pietra, et J. Lafferty, 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(4), 1–13.
- (Dowding et al., 1993) J. Dowding, J. M. Gawron, D. E. Appelt, J. Bear, L. Cherny, R. C. Moore, et D. B. Moran, 1993. GEMINI : A natural language system for spoken-language understanding. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, 54–61.
- (Dowty, 1979) D. Dowty, 1979. *Word Meaning and Montague Grammar : The Semantics of Verbs and Times in Generative Semantics and in Montague’s PTQ*. Dordrecht : D. Reidel.
- (Duffy et Helmbold, 1999) N. Duffy et D. Helmbold, 1999. Potential boosters ? Dans les actes de *Advances in Neural Information Processing Systems*.
- (Duvert et al., 2006) F. Duvert, C. Jonquet, P. Dugénie, et S. A. Cerri, 2006. Agent-grid integration ontology. Dans les actes de *2nd International Workshop on Agents, Web Services and Ontologies Merging, AWeSOMe’06*, Volume 4277(1), 136–146. Springer.
- (Duvert et al., 2008) F. Duvert, M.-J. Meurs, C. Servan, F. Béchet, F. Lefevre, et R. de Mori, 2008. Semantic composition process in a speech understanding system. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.
- (Epstein et al., 1996) M. Epstein, K. Papineni, S. Roukos, T. Ward, et S. A. Della Pietra, 1996. Statistical natural language understanding using hidden clumpings. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Washington, DC, USA, 176–179. IEEE Computer Society.
- (Erdogan et al., 2005) H. Erdogan, R. Sarikaya, S. F. Chen, Y. Gao, et M. Picheny, 2005. Using semantic analysis to improve speech recognition performance. *Computer, Speech and Language* 19(3), 321–343.

- (Erman et al., 1980) L. D. Erman, F. Hayes-Roth, V. R. Lesser, et D. R. Reddy, 1980. The hearsay-ii speech-understanding system : Integrating knowledge to resolve uncertainty. *ACM Comput. Surv.* 12(2), 213–253.
- (Fellbaum, 1998) C. Fellbaum, 1998. *WordNet An Electronic Lexical Database*. Cambridge, MA ; London : The MIT Press.
- (Fikes et Nilsson, 1971) R. E. Fikes et N. J. Nilsson, 1971. Strips : a new approach to the application of theorem proving to problem solving. Dans les actes de *International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 608–620. Morgan Kaufmann Publishers Inc.
- (Fillmore, 1968) C. J. Fillmore, 1968. *The case for case*, 1–88. New York : Holt, Rinehart, and Winston.
- (Freund et Schapire, 1996) Y. Freund et R. E. Schapire, 1996. Experiments with a new boosting algorithm. Dans les actes de *International Conference on Machine Learning*, 148–156.
- (Freund et Schapire, 1997) Y. Freund et R. E. Schapire, 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139.
- (Freund et Schapire, 1999) Y. Freund et R. E. Schapire, 1999. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence* 14(5), 771–780. Appearing in Japanese, translation by Naoki Abe.
- (Friedman et al., 2000) J. Friedman, T. Hastie, et R. Tibshirani, 2000. Additive logistic regression : A statistical view of boosting. *The Annals of Statistics* 38(2), 337–374.
- (Fu, 1982) J. D. Fu, 1982. Syntactic Pattern Recognition. Dans E. Cliffs (Ed.), *Theory and Application*. Prentice-Halls.
- (Gazdar et al., 1985) G. Gazdar, E. Klein, G. K. Pullum, et I. Sag, 1985. *Generalized Phrase Structure Grammar Cambridge*. Harvard University Press.
- (Ge et Mooney, 2005) R. Ge et R. J. Mooney, 2005. A statistical semantic parser that integrates syntax and semantics. Dans les actes de *Conference on Computational Natural Language Learning*, Morristown, NJ, USA, 9–16. Association for Computational Linguistics.
- (Gildea et Jurafsky, 2002) D. Gildea et D. Jurafsky, 2002. Automatic labeling of semantic roles. *International Conference on Computational Linguistics* 28(3), 245–288.
- (Glass et al., 1995) J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, et V. Zue, 1995. Multilingual spoken-language understanding in the mit voyager system. *Speech Communication* 17(1-2), 1–18.
- (Gorin, 1995) A. L. Gorin, 1995. On automated language acquisition. *Journal of the Acoustical Society of America* 97(6), 3441–3461.

- (Gorin et al., 1997) A. L. Gorin, G. Riccardi, et J. H. Wright, 1997. How may i help you? *Speech Communication* 23(1-2), 113–127.
- (Gu et al., 2006) L. Gu, Y. Gao, F.-H. Liu, et M. Picheny, 2006. Concept-based speech-to-speech translation using maximum entropy models for statistical natural concept generation. *IEEE Transactions on Audio, Speech and Language Processing* 14(2), 377–392.
- (Hacioglu, 2004) K. Hacioglu, 2004. A lightweight semantic chunker based on tagging. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Boston, Massachusetts, USA, 145–148.
- (Hacioglu et Ward, 2003) K. Hacioglu et W. H. Ward, 2003. Target word detection and semantic role chunking using support vector machines. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Morristown, NJ, USA, 25–27. Association for Computational Linguistics.
- (Hakkani-Tür et al., 2006) D. Hakkani-Tür, G. Riccardi, et G. Tür, 2006. An active approach to spoken language processing. *ACM Transaction on Speech Language Processing* 3(3), 1–31.  
English
- (Hall, 1973) P. A. Hall, 1973. Equivalence between AND/OR graphs and context-free grammars. *Communications of the Association for Computing Machines* 16, 444–445.
- (Hammersley et Clifford, 1971) J. M. Hammersley et P. Clifford, 1971. Markov field on finite graphs and lattices. *Unpublished manuscript* -, -.
- (He et Young, 2003b) Y. He et S. Young, 2003b. Hidden vector state model for hierarchical semantic parsing. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, 268–271.
- (Hendler et McGuinness, 2000) J. Hendler et D. L. McGuinness, 2000. The DARPA Agent Markup Language. *IEEE Intelligent Systems* 15(6), 67–73.
- (Hobbs et al., 1996) J. R. Hobbs, D. E. Appelt, J. Bear, D. J. Israel, M. Kameyama, A. Kehler, M. E. Stickel, et M. Tyson, 1996. Sri’s tipster ii project. Dans les actes de *ACL, Advances in Text Processing*, Morristown, NJ, USA, 201–208. Association for Computational Linguistics.
- (Hobbs et Israel, 1994) J. R. Hobbs et D. J. Israel, 1994. Principles of template design. Dans les actes de *workshop on Human Language Technology (HLT)*.
- (Jackendoff, 1990) R. Jackendoff, 1990. *Semantic Structures*. The MIT Press, Cambridge Mass.
- (Jackendoff, 2002) R. Jackendoff, 2002. *Foundations of Language : Brain, Meaning, Grammar, Evolution*. Oxford University Press.

- (Jackson et al., 1991) E. Jackson, D. E. Appelt, J. Bear, R. C. Moore, et A. Podlozny, 1991. A template matcher for robust nl interpretation. Dans les actes de *workshop on Human Language Technology (HLT)*.
- (Ji et Billes, 2005) G. Ji et J. A. Billes, 2005. Dialog act tagging using graphical models. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- (Ji et Billes, 2006) G. Ji et J. A. Billes, 2006. Backoff model training using partially observed data : application to dialog act tagging. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Morristown, NJ, USA, 280–287. Association for Computational Linguistics.
- (Jones et Warren, 1982) M. A. Jones et D. S. Warren, 1982. Conceptual dependency and montage grammar : A step toward conciliation. Dans les actes de *National Conference on Artificial Intelligence*, 79–83.
- (Jonquet et Cerri, 2005) C. Jonquet et S. A. Cerri, 2005. i-dialogue : modeling agent conversation by streams and lazy evaluation. Dans les actes de *International Lisp Conference, ILC'05*, 219–228.
- (Joshi, 2001) A. K. Joshi, 2001. The xtag project at penn. Dans les actes de *IWPT*. Tsinghua University Press.
- (Kasper et Hovy, 1990) R. T. Kasper et E. H. Hovy, 1990. Performing integrated syntactic and semantic parsing using classification. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 54–59. Association for Computational Linguistics.
- (Kawahara et al., 1999) T. Kawahara, K. Tanaka, et S. Doshira, 1999. Virtual fitting room with spoken dialogue interface. Dans les actes de *ESCA Workshop on Interactive Dialog in Multi-Modal Systems*, Irsee, Germany, 5–8.
- (Kifer et al., 1995) M. Kifer, G. Lausen, et J. Wu, 1995. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM* 42, 741–843.
- (Kipper et al., 2000) K. Kipper, H. T. Dang, et M. Palmer, 2000. Class-based construction of a verb lexicon. Dans les actes de *National Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence*, 691–696. AAAI Press / The MIT Press.
- (Klatt, 1977) D. H. Klatt, 1977. Review of the ARPA speech understanding project. *Journal of the Acoustical Society of America* 62(6), 1345–1366.
- (Koppelman et al., 1995) J. Koppelman, S. A. Della Pietra, M. Epstein, et S. Roukos, 1995. A statistical approach to language modeling for the atis task. Dans les actes de *Spoken Language Systems Workshop*, Madrid, Spain., 1785–1788.



- (Kudo et Matsumoto, 2001) T. Kudo et Y. Matsumoto, 2001. Chunking with support vector machines. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 1–8.
- (Kuhn et De Mori, 1995) R. Kuhn et R. De Mori, 1995. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 449–460.
- (Kupiec, 1992) J. Kupiec, 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer, Speech and Language* 6, 225–242.
- (Lafferty et al., 2001) J. Lafferty, A. McCallum, et F. Pereira, 2001. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. Dans les actes de *International Conference on Machine Learning*, 282–289. Morgan Kaufmann, San Francisco, CA.
- (Lambek, 1958) J. Lambek, 1958. The mathematics of sentence structure. *American Mathematical Monthly* 65(3), 154–170.
- (Lefèvre, 2006) F. Lefèvre, 2006. A dbn-based multi-level stochastic spoken language understanding system. Dans les actes de *IEEE workshop on Spoken Language Technology (SLT)*.
- (Lefèvre, 2007) F. Lefèvre, 2007. Dynamic bayesian networks and discriminative classifiers for multi-stage semantic interpretation. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hawaii, USA.
- (Levesque et Brachman, 1985) H. J. Levesque et R. J. Brachman, 1985. A fundamental tradeoff in knowledge representation and reasoning (revised version). Dans R. J. Brachman et H. J. Levesque (Eds.), *Readings in Knowledge Representation*, 41–70. Los Altos, CA : Kaufmann.
- (Levin, 1993) B. Levin, 1993. *English Verb Classes and Alternations : A Preliminary Investigation*. Chicago : Chicago University Press.
- (Li et Chou, 2002) L. Li et W. Chou, 2002. Improving latent semantic indexing based classifier with information gain. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, 1141–1144.
- (Liu et al., 2005) Y. Liu, A. Stolcke, E. Shriberg, et M. P. Harper, 2005. Using conditional random fields for sentence boundary detection in speech. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 451–458. Association for Computational Linguistics.
- (Lowe et al., 1997) J. B. Lowe, C. F. Baker, et C. J. Fillmore, 1997. A frame-semantic approach to semantic annotation. Dans les actes de *SIGLEX Workshop on Tagging Text with Lexical Semantics : Why, What, and How ?*, Washington D.C., USA.

- (LUNA, 2009) LUNA, 2009. spoken Language UNderstanding in multilinguAl communications systems, project luna : <http://www.ist-luna.eu>. European Community, Sixth framework programme priority 2, information society technology, project n 33549. Final report.
- (Macherey et al., 2001) K. Macherey, F. J. Och, et H. Ney, 2001. Natural language understanding using statistical machine translation. Dans les actes de *Eurospeech*, Aalborg, Denmark, 2205–2208.
- (Mast et al., 1996) M. Mast, R. Kompe, S. Harbeck, A. Kießling, H. Niemann, E. Nöth, E. Schukat-Talamazzini, et V. Warnke, 1996. Dialog act classification with the help of prosody. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, 1732–1735.
- (McCallum et al., 2000) A. McCallum, D. Freitag, et F. Pereira, 2000. Maximum entropy markov models for information extraction and segmentation. Dans les actes de *International Conference on Machine Learning*, San Francisco, CA, USA, 591–598.
- (McCallum et Wellner, 2003) A. McCallum et B. Wellner, 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. Dans les actes de *Advances in Neural Information Processing Systems*, 905–912. MIT Press.
- (McDonald et al., 2005) R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, et P. White, 2005. Simple algorithms for complex relation extraction with applications to biomedical ie. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 491–498. Association for Computational Linguistics.
- (McRoy et Hirst, 1990) S. W. McRoy et G. Hirst, 1990. Race-based parsing and syntactic disambiguation. *Cognitive Science* 14(3), 313–353.
- (Meng et al., 1999) H. M. Meng, W. Lam, et C. Wai, 1999. To believe is to understand. Dans les actes de *European Conference on Speech Communication and Technology (Eurospeech)*, 2015–2018.
- (Meurs et al., 2008) M.-J. Meurs, F. Lefèvre, et R. De Mori, 2008. A bayesian approach to semantic composition for spoken language interpretation. Dans les actes de *Interspeech*, Brisbane, Australia.
- (Miller, 1995) G. A. Miller, 1995. Wordnet : a lexical database for english. *Communications of the Association for Computing Machines* 38(11), 39–41.
- (Miller et al., 1990) 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.
- (Miller et al., 1995) S. Miller, M. Bates, R. Bobrow, R. Ingria, J. Makhoul, et R. Schwartz, 1995. Recent progress in hidden understanding models. Dans les actes de *Spoken Language Systems Workshop*, Austin TX, 276–279.

- (Miller et al., 1994) S. Miller, R. Schwartz, R. Bobrow, et R. Ingria, 1994. Statistical language processing using hidden understanding models. Dans les actes de *IEEE workshop on Spoken Language Technology (SLT)*, Plainsboro, New Jersey, USA, 48–52. Morgan Kaufmann, Los Altos, CA.
- (Mitchell, 1997) T. M. Mitchell, 1997. *Machine Learning*. McGraw-Hill Higher Education.
- (Molina et Pla, 2002) A. Molina et F. Pla, 2002. Shallow parsing using specialized hmms. *Journal of Machine Learning Research* 2, 595–613.
- (Montague, 1974) R. Montague, 1974. *Formal Philosophy : Selected Papers of Richard Montague*. New Haven : Yale University Press.
- (Moore et al., 1995) R. C. Moore, D. E. Appelt, J. Dowding, J. M. Gawron, et D. B. Moran, 1995. Combining linguistic and statistical knowledge sources in natural-language processing for ATIS. Dans les actes de *ARPA Workshop on Spoken Language Technology*, Austin, TX, 261–264. Morgan Kaufmann.
- (Mori, 2011) R. D. Mori, April 2011. *Roots and history of SLU systems*. Spoken Language Understanding. J. Wiley : Gokhan Tur and Renato De Mori Eds.
- (Moschitti, 2004) A. Moschitti, 2004. A study on convolution kernels for shallow semantic parsing. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 335. Association for Computational Linguistics.
- (Nagai et al., 1994) A. Nagai, Y. Ishikawa, et K. Nakajima, 1994. A semantic interpretation based on detecting concepts for spontaneous speech understanding. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*.
- (Nasr et al., 1999) A. Nasr, Y. Estève, F. Béchet, T. Spriet, et R. De Mori, 1999. A language model combining n-grams and stochastic finite state automata. Dans les actes de *Eurospeech*, Volume 5, Budapest, Hungary, 2175–2178.
- (Newman, 2007) P. S. Newman, 2007. Rh : a retro hybrid parser. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Morristown, NJ, USA, 121–124. Association for Computational Linguistics.
- (Nilsson, 1980) N. J. Nilsson, 1980. *Principles of artificial intelligence*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.
- (Nocera et al., 2002) P. Nocera, G. Linarès, et D. Massonié, 2002. Principes et performances du décodeur parole continue speeral. Dans les actes de *Journées d'étude de la parole (JEP)*, Nancy.
- (Norvig, 1983) P. Norvig, 1983. Frame activated inferences in a story understanding program. Dans les actes de *International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 624–626. Morgan Kaufmann Publishers Inc.



- (Ostendorf et al., 1991) M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz, et J. R. Rohlicek, 1991. Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 83–87. Association for Computational Linguistics.
- (Palmer et al., 2005) M. Palmer, D. Gildea, et P. Kingsbury, 2005. The proposition bank : An annotated corpus of semantic roles. *International Conference on Computational Linguistics* 31(1), 71–106.
- (Papineni et al., 1998) K. Papineni, S. Roukos, et R. Ward, 1998. Maximum likelihood and discriminative training of direct translation models. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 1, 189–192 vol.1.
- (Partee, 1976) B. Partee, 1976. *Montague Grammar*. New York : Academic Press.
- (Patel-Schneider et al., 2003) P. F. Patel-Schneider, P. Hayes, et I. Horrocks, 2003. Owl web ontology language semantics and abstract syntax. Rapport technique, W3C working Draft.
- (Pazzani et Billsus, 1997) M. Pazzani et D. Billsus, 1997. Learning and revising user profiles : The identification of interesting web sites. *Machine Learning* 27, 313–331.
- (Pereira, 1990) F. Pereira, 1990. Finite-state approximations of grammars. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 20–25. Association for Computational Linguistics.
- (Potamianos et al., 2005) A. Potamianos, S. Narayanan, et G. Riccardi, 2005. Adaptive categorical understanding for spoken dialogue systems. *IEEE* 13(3), 321–329.
- (Pradhan et al., 2004) S. Pradhan, W. H. Ward, K. Hacioglu, J. H. Martin, et D. Jurafsky, 2004. Shallow semantic parsing using support vector machines. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Boston, Massachusetts, USA, 233–240.
- (Pradhan et al., 2007) S. Pradhan, W. H. Ward, et J. H. Martin, 2007. Towards robust semantic role labeling. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, 556–563.
- (Prieto et al., 1994) N. Prieto, E. Sanchis, et L. Palmero, 1994. Continuous speech understanding based on automatic learning of acoustic and semantic models. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, Yokohama, Japan.
- (Punyakanok et Roth, 2000) V. Punyakanok et D. Roth, 2000. The use of classifiers in sequential inference. Dans les actes de *Advances in Neural Information Processing Systems*, 995–1001.

- (Punyakankok et al., 2004) V. Punyakankok, D. Roth, W. tau Yih, et D. Zimak, 2004. Learning via inference over structurally constrained. Dans les actes de *Workshop on Learning Structured with Output, NPIS*.
- (Rambow et al., 2002) O. Rambow, S. Bangalore, T. Butt, A. Nasr, et R. Sproat, 2002. Creating a finite-state parser with application semantics. Dans les actes de *International Conference on Computational Linguistics*, Morristown, NJ, USA, 1–5. Association for Computational Linguistics.
- (Ramshaw et Marcus, 1995) L. Ramshaw et M. Marcus, 1995. Text chunking using transformation-based learning. Dans D. Yarovsky et K. Church (Eds.), *Workshop on Very Large Corpora*, Somerset, New Jersey, 82–94. Association for Computational Linguistics.
- (Ratnaparkhi, 1996) A. Ratnaparkhi, 1996. A maximum entropy part-of-speech tagger. Dans les actes de *conference on Empirical methods in natural language processing*, University of Pennsylvania, 133–142. Annual Meeting of the Association for Computational Linguistics (ACL).
- (Raymond, 2005) C. Raymond, 2005. *Décodage conceptuel : co-articulation des processus de transcription et compréhension dans les systèmes de dialogue*. Thèse de Doctorat, Université d'Avignon et des Pays du Vaucluse.
- (Raymond et al., 2006) C. Raymond, F. Béchet, R. De Mori, et G. Damnati, 2006. On the use of finite state transducers for semantic interpretation. *Speech Communication* 48,3-4, 288–304.
- (Raymond et Riccardi, 2007) C. Raymond et G. Riccardi, 2007. Generative and discriminative algorithms for spoken language understanding. Dans les actes de *Inter-speech*, Antwerp, Belgium.
- (Rayner et Hockey, 2003) M. Rayner et B. A. Hockey, 2003. Transparent combination of rule-based and data-driven approaches in a speech understanding architecture. Dans les actes de *Annual Meeting of the European Association for Computational Linguistics (EACL)*, Morristown, NJ, USA, 299–306. Association for Computational Linguistics.
- (Resnik, 1995) P. Resnik, 1995. Disambiguating noun groupings with respect to word-net senses. Dans les actes de *Workshop on Very Large Corpora*, 54–68.
- (Riccardi et al., 1997) G. Riccardi, A. L. Gorin, A. Ljolje, et M. Riley, 1997. Spoken language understanding for automated call routing. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1143–1146.
- (Richardson et al., 1998) S. D. Richardson, W. B. Dolan, et L. Vanderwende, 1998. Mind-net : Acquiring and structuring semantic information from text. Dans les actes de *International Conference on Computational linguistics (COLING/ACL)*, 1098–1102.
- (Riedmiller, 1994) M. Riedmiller, 1994. Rprop - description and implementation details. Rapport technique, Institut fur Logik, Komplexitat und Deduktionssysteme, Karlsruhe.

- (Rivest, 1987) R. L. Rivest, 1987. Learning decision lists. *Machine Learning* 2(3), 229–246.
- (Roark, 2001) B. Roark, 2001. Probabilistic top-down parsing and language modeling. *International Conference on Computational Linguistics* 27(2), 249–276.
- (Roark, 2002) B. Roark, 2002. Markov parsing : lattice rescoring with a statistical parser. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA.
- (Roark et al., 2006) B. Roark, Y. Liu, M. P. Harper, R. Stewart, M. Lease, M. Snover, Z. Shafran, B. Dorr, J. Hale, A. Krasnyanskaya, et L. Young, 2006. Reranking for sentence boundary detection in conversational speech. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 1, Toulouse, France, 545–548.
- (Samuel et al., 1998) K. Samuel, S. Carberry, et K. Vijay-Shanker, 1998. Dialogue act tagging with transformation-based learning. Dans les actes de *International Conference on Computational Linguistics*, Morristown, NJ, USA, 1150–1156. Association for Computational Linguistics.
- (Sang et Buchholz, 2000) E. T. K. Sang et S. Buchholz, 2000. Introduction to the conll-2000 shared task : chunking. Dans les actes de *2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, Morristown, NJ, USA, 127–132. Association for Computational Linguistics.
- (Sarikaya et al., 2004) R. Sarikaya, Y. Gao, et M. Picheny, 2004. A comparison of rule-based and statistical methods for semantic language modeling and confidence measurement. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*, Boston, Massachusetts, USA, 65–68.
- (Schapire, 2002) R. E. Schapire, 2002. The boosting approach to machine learning : An overview. Dans les actes de *MSRI Workshop on Nonlinear Estimation and Classification*.
- (Schapire et Singer, 1999) R. E. Schapire et Y. Singer, 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3), 297–336.
- (Schapire et Singer, 2000) R. E. Schapire et Y. Singer, 2000. BoosTexter : A boosting-based system for text categorization. *Machine Learning* 39, 135–168. Available online at <http://www.cs.princeton.edu/~schapire/boostexter.html>.
- (Schwartz et al., 1992) R. Schwartz, S. Austin, F. Kubala, J. Makhoul, L. Nguyen, P. Placeway, et G. Zavaliagos, 1992. New uses for the n-best sentence hypotheses within the byblos speech recognition system. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 1, San Francisco, California, USA, 1–4.

- (Schwartz et al., 1997) R. Schwartz, S. Miller, D. Stallard, et J. Makhoul, 1997. Hidden understanding models for statistical sentence understanding. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 2, Washington, DC, USA, 1479. IEEE Computer Society.
- (Seneff, 1989) S. Seneff, 1989. TINA : A probabilistic syntactic parser for speech understanding systems. Dans les actes de *Speech and Natural Language Workshop*, Philadelphia, PA, USA, 168–178.
- (Seneff, 1992a) S. Seneff, 1992a. A relaxation method for understanding spontaneous speech utterances. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 299–304. Association for Computational Linguistics.
- (Seneff, 1992b) S. Seneff, 1992b. TINA : A natural language system for spoken language applications. *International Conference on Computational Linguistics 18(1)*, 61–86.
- (Sha et Pereira, 2003) F. Sha et F. Pereira, 2003. Shallow parsing with conditional random fields. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) on Human Language Technology (HLT)*.
- (Shank, 1975) R. C. Shank, 1975. *The Structure of Episodes in Memory*, 273–272. Academic Press, New York.
- (Shapiro et al., 2005) R. Shapiro, M. Rochery, M. Rahin, et N. Gupta, 2005. Boosting with prior knowledge for call classification. *IEEE Transactions on Speech and Audio Processing 13(2)*, 174–182.
- (Shriberg et al., 2000) E. Shriberg, A. Stolcke, D. Hakkani-Tür, et G. Tür, 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication 32(1-2)*, 127–154.
- (Stahl et Müller, 1995) H. Stahl et J. Müller, 1995. A stochastic grammar for isolated representation of syntactic and semantic knowledge. Dans les actes de *Eurospeech*.
- (Stallard et Bobrow, 1993) D. Stallard et R. Bobrow, 1993. The semantic linker : a new fragment combining method. Dans les actes de *workshop on Human Language Technology (HLT)*, Morristown, NJ, USA, 37–42. Association for Computational Linguistics.
- (Steedman, 1996) M. Steedman, 1996. *Surface Structure and Interpretation*. MIT Press.
- (Steedman, 2000) M. Steedman, 2000. *The Syntactic Process*. Cambridge, Massachusetts : MIT Press.
- (Stolcke et al., 1998) A. Stolcke, E. Shriberg, S. International, N. Coccaro, D. Jurafsky, R. Martin, et R. Bates, 1998. Dialog act modeling for conversational speech. Dans les actes de *National Conference on Artificial Intelligence*, 98–105. AAAI Press.
- (Sudoh et Tsukada, 2005) K. Sudoh et H. Tsukada, 2005. Tightly integrated spoken language understanding using word-to-concept translation. Dans les actes de *Inter-speech*.

- (Tait, 1985) J. I. Tait, 1985. Semantic parsing and syntactic constraints (mark iv). Dans K. Sparck Jones et Y. Wilks (Eds.), *Automatic Natural Language Parsing*, 169–177. Chichester : Ellis Horwood.
- (Thomae et al., 2005) M. Thomae, T. Fabian, R. Lieb, et G. Ruske, 2005. Hierarchical language models for one-stage speech interpretation. Dans les actes de *European Conference on Speech Communication and Technology (Eurospeech)*, Lisbon, Portugal, 1149–1152.
- (Toutanova et al., 2005) K. Toutanova, A. Haghighi, et C. D. Manning, 2005. Joint learning improves semantic role labeling. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Morristown, NJ, USA, 589–596. Association for Computational Linguistics.
- (Vidal et al., 1993) E. Vidal, R. Pieraccini, et E. Levin, 1993. Learning associations between grammars : a new approach to natural language understanding. Dans les actes de *European Conference on Speech Communication and Technology (Eurospeech)*, Berlin, Germany.
- (Walker, 1975) D. Walker, 1975. The sri speech understanding system. *IEEE Transactions on Acoustics, Speech, And Signal Proceeding ASSP-23, NO-5*, 397–416.
- (Walker et al., 2002) M. A. Walker, A. I. Rudnicky, R. Prasad, J. Aberdeen, E. O. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, S. Greg, S. Seneff, et D. Stallard, 2002. Darpa communicator : Cross-system results for the 2001 evaluation. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, 269–272.
- (Waltz, 1981) D. L. Waltz, 1981. Toward a detailed model of processing for language describing the physical world. Dans les actes de *International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 1–6. Morgan Kaufmann Publishers Inc.
- (Wang et al., 1999) H. Wang, W. Gao, et S. Li, 1999. Dialog act analysis of spoken chinese based on neural networks. *Chinese Journal of Computers* 22(10), 1014–1018.
- (Wang, 2003) K. Wang, 2003. Semantics synchronous understanding for robust spoken language applications. Dans les actes de *Automatic Speech Recognition and Understanding workshop (ASRU)*.
- (Wang et al., 2002) W. Wang, Y. Liu, et M. P. Harper, 2002. Rescoring effectiveness of language models using different levels of knowledge and their integration. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 785–788.
- (Wang et Acero, 2001) Y. Wang et A. Acero, 2001. Grammar learning for spoken language understanding. Dans les actes de *Automatic Speech Recognition and Understanding workshop (ASRU)*, 1229–1244.



- (Wang et Acero, 2002) Y. Wang et A. Acero, 2002. Evaluation of spoken language grammar learning in atis domain. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- (Wang et Acero, 2003) Y. Wang et A. Acero, 2003. Combination of CFG and n-gram modeling in semantic grammar learning. Dans les actes de *European Conference on Speech Communication and Technology (Eurospeech)*, Geneva, Switzerland, 2809–2812.
- (Wang et Acero, 2006) Y. Wang et A. Acero, 2006. Discriminative models for spoken language understanding. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*.
- (Wang et al., 2002) Y. Wang, A. Acero, C. Chelba, B. Frey, et L. Wong, 2002. Combination of statistical and rule-based approaches for spoken language understanding. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, USA, 609–613.
- (Wang et al., 2000) Y. Wang, M. Mahajan, et X. Huang, 2000. A unified context-free grammar and N-gram model for spoken language processing. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey.
- (Ward, 1990) W. H. Ward, 1990. The cmu air travel information service : understanding spontaneous speech. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 127–129. Association for Computational Linguistics.
- (Wong et Mooney, 2007) Y. W. Wong et R. J. Mooney, 2007. Learning synchronous grammars for semantic parsing with lambda calculus. Dans les actes de *Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, 960–967. Association for Computational Linguistics.
- (Woods, 1970) W. A. Woods, 1970. Transition network grammars for natural language analysis. *Communications of the Association for Computing Machines* 13(10), 591–606.
- (Woods, 1975) W. A. Woods, 1975. *What's in a Link : Foundations for Semantic Networks*. Bolt, Beranek and Newman.
- (Woods et al., 1976) W. A. Woods, M. Bates, G. Brown, C. C. Cook., et B. C. Bruce, 1976. Speech understanding systems, final technical progress report, volumes i-v. Rapport technique, Cambridge, MA.
- (Wu et al., 2002) C.-H. Wu, G.-L. Yan, et C.-L. Lin, 2002. Speech act modeling in a spoken dialog system using a fuzzy fragment-class markov model. *Speech Communication* 38(1), 183–199.
- (Young et al., 1989) S. R. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, et P. Werner, 1989. High level knowledge sources in usable speech recognition systems. *Communications of the Association for Computing Machines* 32(2), 183–194.

- (Zechner, 1998) K. Zechner, 1998. Automatic construction of frame representations for spontaneous speech in unrestricted domains. Dans les actes de *International Conference on Computational Linguistics*, Morristown, NJ, USA, 1448–1452. Association for Computational Linguistics.
- (Zettlemoyer et Collins, 2005) L. S. Zettlemoyer et M. Collins, 2005. Learning to map sentences to logical form : Structured classification with probabilistic categorial grammars. Dans les actes de *Conference on Uncertainty in Artificial Intelligence*, 658–666.
- (Zettlemoyer et Collins, 2009) L. S. Zettlemoyer et M. Collins, 2009. Learning context-dependent mappings from sentences to logical form. Dans les actes de *Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, Morristown, NJ, USA, 976–984. Association for Computational Linguistics.
- (Zhang et Rudnicky, 2002) R. Zhang et A. I. Rudnicky, 2002. Improve latent semantic analysis based language model by integrating multiple level knowledge. Dans les actes de *International Conference on Spoken Language Processing (ICSLP)*, 893–896.
- (Zimmermann, 2005) M. Zimmermann, 2005. Toward joint segmentation and classification of dialog acts in multiparty meetings. Dans les actes de *Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI-05)*, 187. Springer.
- (Zue et al., 1992) V. Zue, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, M. Phillips, J. Polifroni, et S. Seneff, 1992. The mit atis system : February 1992 progress report. Dans les actes de *Human Language Technology, workshop on Speech and Natural Language*, Morristown, NJ, USA, 84–88. Association for Computational Linguistics.
- (Zue et al., 1993) V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goddeau, J. Glass, et E. Brill, 1993. The mit atis system : December 1993 progress report. Dans les actes de *IEEE workshop on Spoken Language Technology (SLT)*, Morristown, NJ, USA, 84–88. Association for Computational Linguistics.

# Publications Personnelles

- (Béchet et al., 2010) F. Béchet, C. Raymon, F. Duvert, et R. de Mori, 2010. Frame based interpretation of conversational speech. Dans les actes de *IEEE workshop on Spoken Language Technology (SLT)*, Berkeley, California, U.S.A.
- (Duvert et de Mori, 2010) F. Duvert et R. de Mori, 2010. Conditional models for detecting lambda-functions in a spoken language understanding system, à paraître. Dans les actes de *INTERSPEECH*, Makuhari, Tokyo, Japan.
- (Duvert et de Mori, 2011) F. Duvert et R. de Mori, 2011. A conditional model for triggering understanding actions in speech understanding system. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic. IEEE.
- (Duvert et al., 2006) F. Duvert, C. Jonquet, P. Dugenie, et S. A. Cerri, 2006. Agent grid integration ontology. Dans les actes de *AWeSOME'2006 Workshop, Springer, LNCS*, 136–146.
- (Duvert et al., 2008a) F. Duvert, M.-J. Meurs, C. Servan, F. Béchet, F. Lefèvre, et R. de Mori, 2008a. Composition sémantique pour la compréhension de la parole dans le cadre de dialogue. Dans les actes de *JEP (Journées d'étude de la parole)*, Avignon, France.
- (Duvert et al., 2008b) F. Duvert, M.-J. Meurs, C. Servan, F. Béchet, F. Lefèvre, et R. de Mori, 2008b. Semantic composition process in a speech understanding system. Dans les actes de *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, U.S.A. IEEE.
- (Meurs et al., 2008a) M.-J. Meurs, F. Duvert, F. Béchet, F. Lefèvre, et R. de Mori, 2008a. Annotation en frames sémantiques du corpus de dialogue media. Dans les actes de *Traitement Automatique du Langage Naturel*, Avignon, France.
- (Meurs et al., 2008b) M.-J. Meurs, F. Duvert, F. Béchet, F. Lefèvre, et R. de Mori, 2008b. Semantic frame annotation on the french media corpus. Dans les actes de *International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association.



- (Meurs et al., 2008c) M.-J. Meurs, F. Duvert, F. Lefèvre, et R. de Mori, 2008c. Markov logic networks for spoken language interpretation. Dans les actes de *Intelligent Information Systems International Conference, Zakopane, Zakopane, Poland*.
- (Meurs et al., 2008d) M.-J. Meurs, F. Duvert, C. Servan, F. Béchet, F. Lefèvre, et R. de Mori, 2008d. Semantic composition process in a spoken understanding system. Dans les actes de *Intelligent Information Systems International Conference, Zakopane, Zakopane, Poland*.
- (Salle et al., 2007) P. Salle, F. Duvert, D. Herin, et S. A. Cerri, 2007. Dynamic workflow in grid-mas integration context. Dans les actes de *SYNASC '07 : Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Washington, DC, USA, 242–245*. IEEE Computer Society.

**Troisième partie**

**Appendices**



## Annexe A

# Corpus utilisé et Ontologie du domaine

### Sommaire

---

<b>A.1 Le Corpus MEDIA</b> . . . . .	<b>188</b>
A.1.1 Dictionnaire sémantique MEDIA . . . . .	189
A.1.2 Annotation . . . . .	190
<b>A.2 Ontologie du domaine lié à MEDIA</b> . . . . .	<b>191</b>

---

## A.1 Le Corpus MEDIA

Les applications de dialogue Homme-Machine considérées dans ce manuscrit peuvent être considérées comme une interface entre des utilisateurs et une base de données. Le but du dialogue est de remplir tous les champs d'une requête qui va permettre d'atteindre l'objectif fixé par l'utilisateur et qui va être adressée à la base de données.

Dans ce cadre et afin de comprendre ce qui est dit dans un dialogue Homme-Machine, nous avons utilisé le corpus de dialogue MEDIA, [Bonneau-Maynard et al. \(2005a\)](#) (programme Technolange/Evalda). La campagne d'évaluation MEDIA se place dans ce cadre applicatif. MEDIA est une simulation d'un système de réservation d'hôtels et d'informations touristiques. C'est un corpus de 1250 dialogues enregistrés par ELDA, selon un protocole de *Magicien d'Oz* (WOZ), simulation du locuteur machine par un opérateur humain. 250 locuteurs différents ont effectué 5 scénarios de réservation d'hôtel avec un système de dialogue. Ce corpus a été entièrement transcrit manuellement, puis annoté sémantiquement selon un dictionnaire sémantique de concepts mis au point par les partenaires du projet MEDIA.

Un exemple de dialogue est donné entre un locuteur (**spk**) et compère (**woz**) dans le tableau A.1.

<b>woz</b>	Bienvenue sur le serveur MEDIA système d'informations touristiques et de réservation d'hôtel. Quelle information désirez-vous ?
<b>spk</b>	Je voudrais réserver pour le troisième week-end de janvier.
<b>woz</b>	Dans quelle ville souhaitez-vous réserver un hôtel pour le troisième week-end de janvier ?
<b>spk</b>	Dans un hôtel à Evreux dans le centre-ville.
<b>woz</b>	Quel type de chambres désirez-vous réserver ?
<b>spk</b>	Alors euh il me faudrait deux chambres doubles et une simple.

TABLE A.1 – Exemple de dialogue extrait du corpus MEDIA

Le corpus a été divisé en 3 sous-corpus, le corpus d'apprentissage (Train) contenant 720 dialogues, environ 12000 messages d'utilisateurs, le corpus de développement (Dev), contenant 200 dialogues, environ 3000 messages d'utilisateurs et le corpus de tests (Test), contenant 200 dialogues, environ 3000 messages d'utilisateurs (résumé dans le tableau A.2).

Lot	nombre de dialogues
Train	720
Dev	200
Test	200

TABLE A.2 – Nombre de dialogue par lot dans le corpus MEDIA

### A.1.1 Dictionnaire sémantique MEDIA

Le dictionnaire sémantique de MEDIA, permet d'associer trois types d'informations à un groupe de mots [Bonneau-Maynard et al. \(2005a\)](#). Ces informations sont des annotations de la transcription manuelle. On trouve :

- les paires attributs-valeurs, correspondant à une représentation sémantique à plat d'un énoncé.
- des spécifieurs qui définissent des relations entre les attributs, et peuvent définir une représentation hiérarchique de l'interprétation d'un énoncé,
- une information sur la modalité de l'attribut de l'énoncé, (affirmatif, négatif, interrogatif ou optionnel)

$n$	$W^{c_n}$	$c_n$	mode	spécifieur	valeur
16	pour un couple	sejour-nbCouple	+		1
17	et un enfant	sejour-nbEnfant	+		1
18	comptez au niveau	null	+		
19	des prix	objet	+	chambre	paiement-montant-entier
20	soixante	paiement-montant-entier	+	chambre	60
21	euros	paiement-monnaie	+		euro
22	par chambre	null	+		

TABLE A.3 – Exemple de message annoté hors-contexte du corpus MEDIA

Le tableau A.3 présente un exemple de message annoté du corpus MEDIA. La première colonne correspond au numéro du segment dans le message, la deuxième colonne à la chaîne de mots  $W^{c_n}$  porteuse du concept  $c_n$  contenu dans la troisième colonne. Les colonnes 4, 5 et 6 contiennent le mode, le spécifieur et la valeur du concept  $c_n$  dans la chaîne  $W^{c_n}$ . Le dictionnaire sémantique MEDIA contient 83 attributs, auxquels peuvent s'ajouter 19 spécifieurs de relations entre attributs.

Les attributs sont dérivés de la base de données associée à l'application MEDIA. Par exemple, *nom-hotel* dérive de la table *hotel* contenue dans la base de données et du champ associé au nom de l'hôtel.

Les spécifieurs sont dépendants de l'attribut et permettent de modifier ou de préciser la signification associée à l'attribut suivant le cas. Dans l'exemple A.3, le spécifieur *chambre* associé à l'attribut *paiement-montant-entier* permet de préciser que la chaîne  $W^{c_n}$  annotée *paiement-montant-entier-chambre* se rapporte au montant de la réservation en cours - *soixante*.

Enfin, des valeurs normalisées ont été adjointes au couple attribut-spécifieur. Elles ont été définies dans un dictionnaire sémantique avec trois configurations possibles :

- une liste de valeurs (par exemple *singulier*, *pluriel*, etc.) ;
- des expressions régulières (pour les dates, par exemple) ;
- des valeurs ouvertes (pour les entités nommées principalement).

Le corpus collecté a été découpé en plusieurs lots. Dans ce chapitre, les 4 premiers lots composent le corpus d'apprentissage (soit 720 dialogues contenant environ 12 000 messages d'utilisateurs) et les lots 5 et 6 forment respectivement les corpus de développement et de test contenant 200 dialogues avec 3 000 messages d'utilisateurs chacun.

### A.1.2 Annotation

La première partie de la campagne d'évaluation MEDIA consistait à analyser des tours de dialogues sans tenir compte du contexte du dialogue. De ce fait, les références ne sont pas résolues. L'exemple proposé dans le tableau A.4 illustre parfaitement le cadre *hors-contexte*, en donnant aux liens référentiels annotés *lienRef* les valeurs *singulier* ou *pluriel*. Ce type d'annotation vise principalement l'attribut *lienRef* et n'influence pas la résolution des autres attributs  $c_n$  pour chaque  $W^{c_n}$ .

Par opposition au *hors-contexte*, la résolution *en-contexte* tient compte du contexte du dialogue. Ceci a pour conséquence la résolution des références, qui passe d'une annotation simplifiée (les valeurs *singulier* ou *pluriel*) à une annotation plus complète, avec pour valeur, à présent, les identifiants  $n$  des  $W^{c_n}$  associés. De plus, nous pouvons avoir plusieurs identifiants pour une seule référence. Les références sont associées aux attributs *lienRef* et les spécifieurs associés sont :

- *coRef* : désigne le référent par référence directe ou coréférence et regroupe les pronoms définis et démonstratifs ;
- *coDom* : désigne le référent grâce à une marque linguistique d'altérité ;
- *elsEns* : désigne le référent d'après des propriétés sémantiques qui l'opposent à d'autres entités dans un ensemble ; on retrouve, dans ce cas là, les ordinaux, les superlatifs, les expressions relatives et certains pronoms démonstratifs.

Voici un exemple d'annotation *hors-contexte* annoté *en-contexte* dans le tableau A.4

Annotation <i>hors-contexte</i>						
$n$	$W^{c_n}$	$c_n$		<i>mode</i>	<i>spécifieur</i>	<i>valeur</i>
0	est	null		+		null
1	il	lienRef		+	coRef	singulier
2	proche du	loc-distanceRelative		?		proche
3	centre-ville	loc-lieuRelatif		?	general	centreville
Annotation <i>en-contexte</i>						
$n$	$W^{c_n}$	$c_n$	référence	<i>mode</i>	<i>spécifieur</i>	<i>valeur</i>
0	est	null		+		null
1	il	lienRef	palais	+	coRef	singulier
2	proche du	loc-distanceRelative		?		proche
3	centre-ville	loc-lieuRelatif		?	general	centreville

TABLE A.4 – Exemple de message annoté *hors-contexte* annoté *en-contexte* du corpus MEDIA sur le message « Est-il proche du centre-ville ? » énoncé après le prompt - « Je vous propose l'hôtel du Palais »

Enfin, le passage entre les deux cadres est aussi marqué par un ajout de spécifieurs, relatifs aux événements apparus précédemment dans le dialogue. Ainsi, au tableau A.3 annoté *hors-contexte*, il nous faut ajouter les spécifieurs comme présenté dans le tableau A.5 :

- *reservation* aux concepts *sejour-nbCouple* et *sejour-nbEnfant* ;
- *conserver chambre* aux concepts *objet* et *paiement-montant-entier*.

On retrouve des spécifieurs faisant référence à des contextes énoncés dans les tours précédents.



Le travail réalisé lors de la campagne MEDIA s'est limité à la représentation *hors-contexte*. Ainsi tous les travaux présentés dans ce chapitre sont basés sur l'évaluation *hors-contexte*.

$n$	$W^{c_n}$	$c_n$	$mode$	$spécifieur$	$valeur$
16	pour un couple	sejour-nbCouple	+	reservation	1
17	et un enfant	sejour-nbEnfant	+	reservation	1
18	comptez au niveau	null	+		
19	des prix	objet	+	chambre	paiement-montant -entier
20	soixante	paiement-montant-entier	+	chambre	60
21	euros	paiement-monnaie	+		euro
22	par chambre	null	+		

TABLE A.5 – Reprise de l'exemple de message annoté hors-contexte A.3, cette fois annoté en-contexte.

Dans les expériences décrites plus loin, nous n'utiliserons que le corpus de test *hors-contexte* pour mesurer l'efficacité de notre système, conformément aux expérimentations menées lors de la campagne MEDIA.

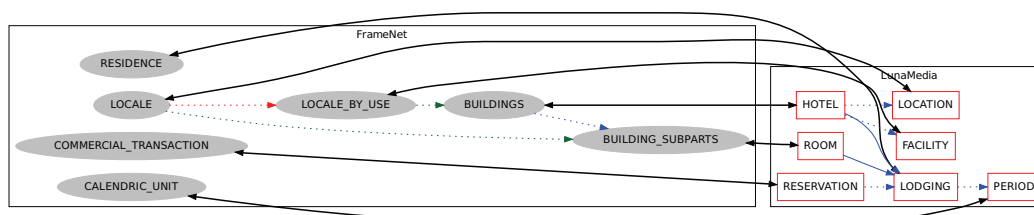
## A.2 Ontologie du domaine lié à MEDIA

Une ontologie du domaine d'application de MEDIA a été développée, afin de pouvoir générer des structures de frames cohérentes. Cette ontologie a été créée à partir de "Protégé 3.4.3" disponible sur <http://protege.stanford.edu/>. Cette ontologie se décompose en trois parties :

- un ensemble de 23 frames du domaine, comme par exemple « ADDRESS », « PERIOD », « HOTEL »...
- un ensemble de 6 frames de négociation : « RESERVATION », « INFORMATION », « TRANSACTION »...
- un ensemble de 22 frames liées aux actes de dialogues : « ACCEPT », « QUERY », « KNOW », "CORRECT\_MODIFY"...

On a défini pour chaque frame, des emplacements. Ces emplacements définissent la frame, son comportement et les relations avec les autres frames grâce aux facettes. Si un emplacement correspond à une valeur, cet emplacement est un attribut, il ne permet pas le lien avec un autre frame. Si un emplacement admet un instance d'une frame c'est alors une facette. Cet emplacement peut recevoir plusieurs types de frame, si le cas arrive, l'emplacement est alors copié autant de fois que nécessaire.

Cette ontologie contient des composants sémantiques qui ne sont pas exprimés nécessairement par un seul mot. Cette représentation conceptuelle diffère d'une approche basée sur FrameNet Baker et al. (1998). La figure A.1 montre la transposition de quelques frames de FrameNet vers l'ontologie proposée pour MEDIA .



**FIGURE A.1** – Représentation en frames d’une partie de la base de connaissance et sa projection de FrameNet vers MEDIA

En ajout à cette ontologie, nous avons créé des règles de correspondances entre des concepts et des fragments de frames. Pour un concept  $C$  on lui associe une frame  $F_C$  telle que  $F_C$  est un fragment minimal représentant le concept.

Les figures suivantes proposent une vue partielle de l’ontologie. La figure A.2, montre l’ontologie pour la partie spécifique au domaine ainsi que les relations et les frames qui sont spécifiques au domaine de la réservation d’hôtel, la figure A.3 montre des frames liées au actes de dialogues. Le tableau A.6 montre les notions de frames, leurs emplacements et les facettes liées aux emplacements.

Frame	Emplacement	Facette
ROOM	establishment r_facilities room_type	HOTEL FACILITY OCCUPANCY
REQUEST	request_theme	RESERVATION KNOW
HOTEL	room_availability  h_facility h_rank	NUMBER OCCUPANCY FACILITY NUMBER
PRICE	amount modality	NUMBER PAYMENT

**TABLE A.6** – Exemple de facettes et d’éléments de descriptions des frames de l’ontologie MEDIA



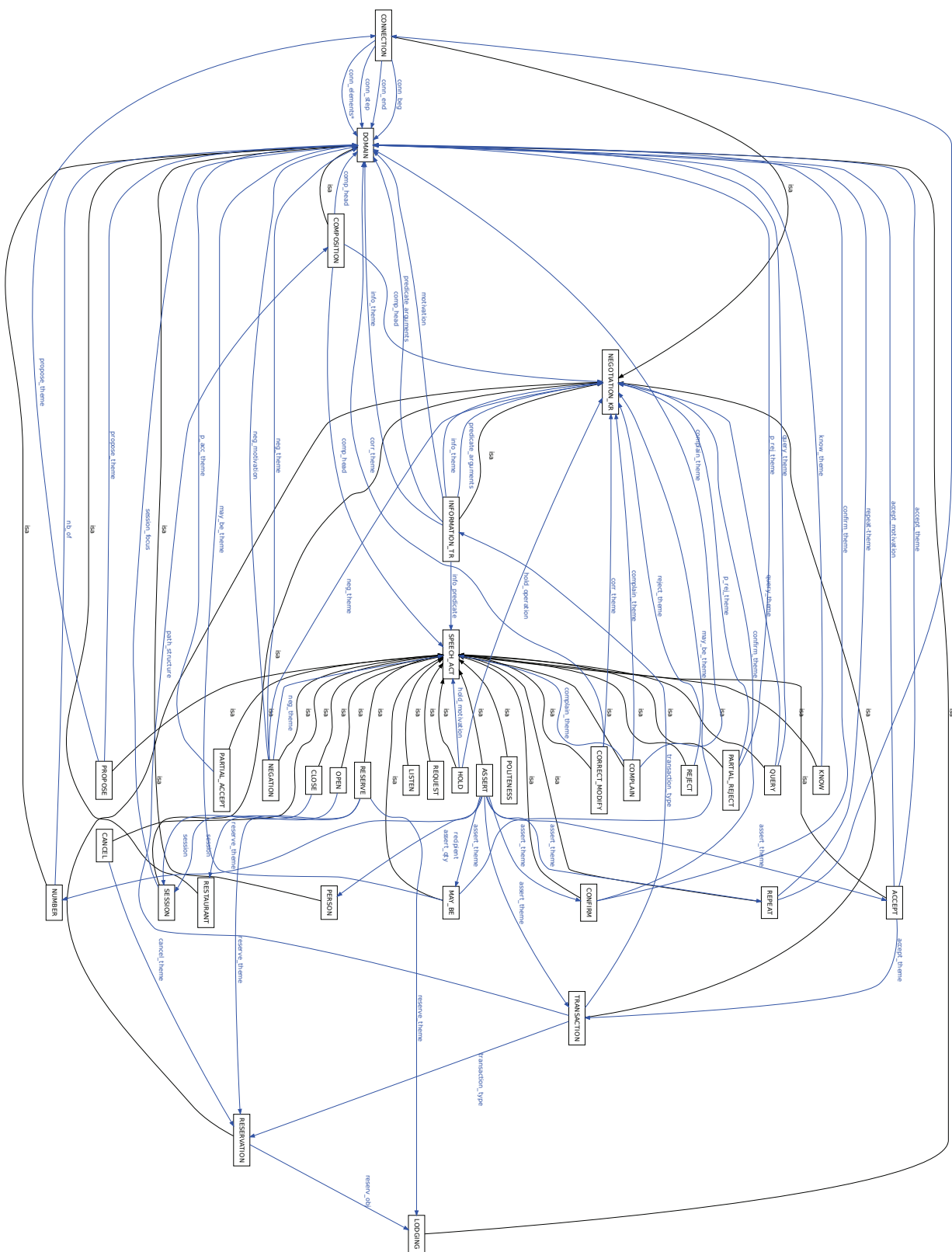


FIGURE A.3 – Représentation partielle de l'ontologie du domaine MEDIA, en particulier les actes de dialogues, les frames sont en majuscules et les emplacements ou éléments de frames sont en minuscules, sur les flèches. Celles-ci représentent les liens entre frames.

## **Annexe B**

# **Protocole d'évaluation**

Pour comparer les différentes approches, notamment celle de la section 6.3.1, nous utilisons la Précision, le Rappel et la F-mesure sur un ensemble de 7 métriques.

La figure B.1 montre une représentation en frame avec l'éditeur de frames SALTO<sup>1</sup>, de la phrase en anglais "good morning I would like to know how much a double room in Ibis costs", équivalent à un tour de dialogue. A partir de cet exemple, nous présentons dans les différentes figures suivantes (B.2 à B.8), les parties de l'annotation qui sont évaluées. Ces évaluations sont présentées dans le tableau suivant B.1.

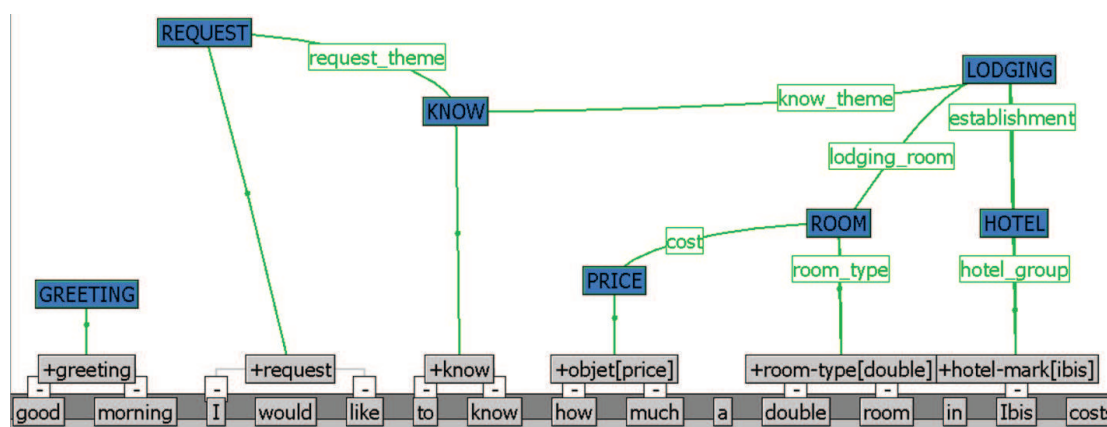


FIGURE B.1 – Exemple d'annotations en frame, illustrant le protocole d'évaluation

N°	Evaluation	Détail	Figure
1	Reconnaissance de frames	Taux de frames prises individuellement, correctes. On ne tient pas compte des arguments	B.2
2	Détection des limites des arguments	Limites des arguments au niveau des mots, sans tenir compte des étiquettes d'arguments	B.3
3	Détection des étiquettes des arguments	Étiquettes correctes des arguments sans tenir comptes des limites des arguments	B.4
4	Reconnaissance des arguments	Évaluation de la composition des tâches définies par les métriques 2 et 3	B.5
5	Réalisation des frames	Composition des métriques 1 + 2 + 3	B.6
6	Composition des frames	Liens et relations de frames à frames	B.7
7	Analyse complète du tour	Composition des métriques 5 + 6	B.8

TABLE B.1 – Les différentes métriques utilisées dans le protocole d'évaluation

1. <http://www.coli.uni-saarland.de/projects/salsa>

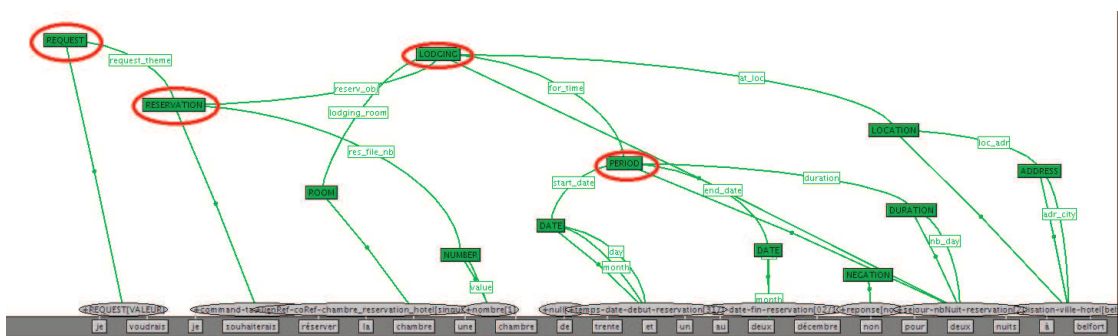


FIGURE B.2 – Evaluation de la métrique 1 : Reconnaissance de frames

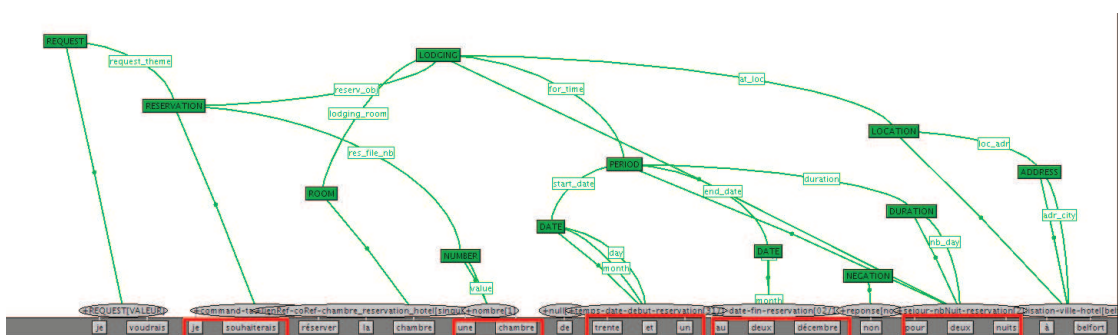


FIGURE B.3 – Evaluation de la métrique 2 : Détection des limites des arguments

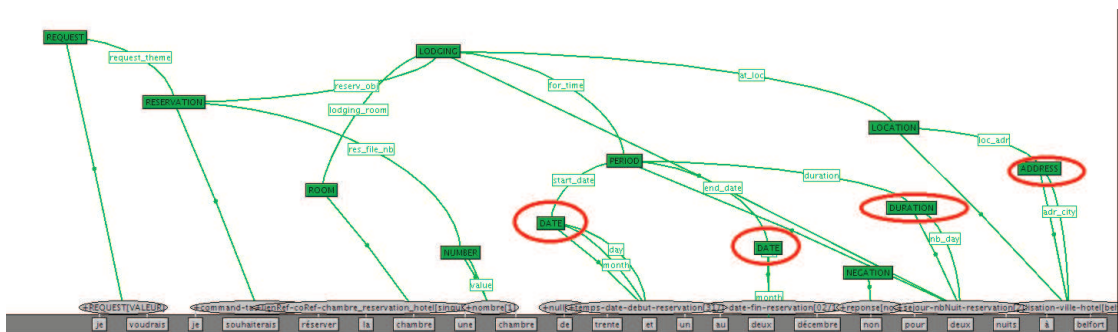


FIGURE B.4 – Evaluation de la métrique 3 : Détection des étiquettes des arguments

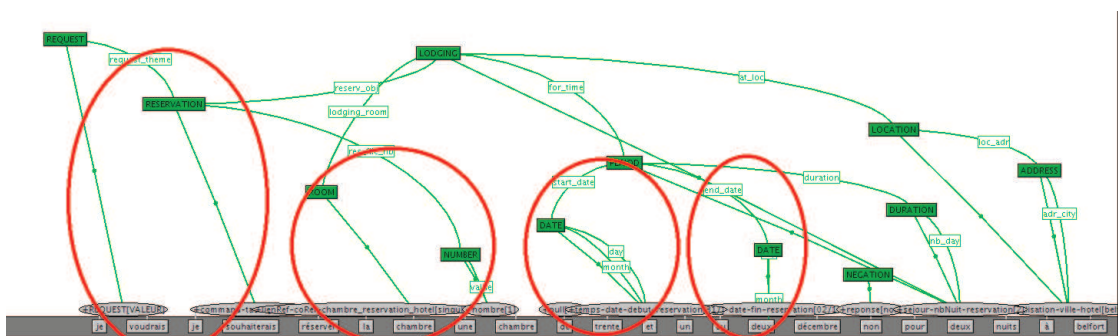


FIGURE B.5 – Evaluation de la métrique 4 : Reconnaissance des arguments



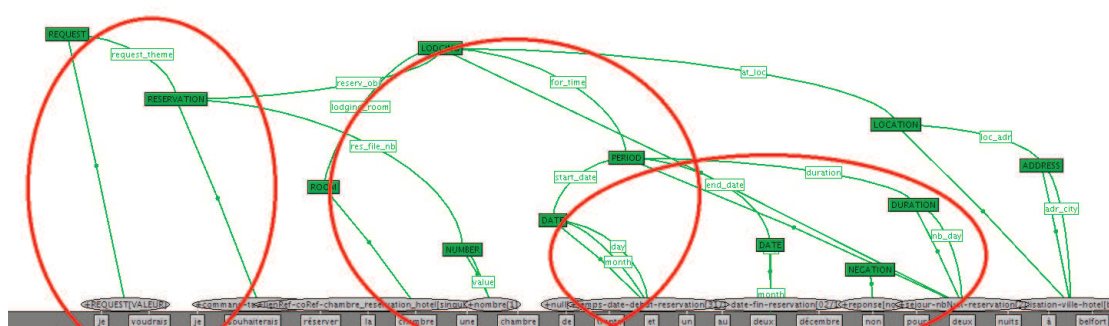


FIGURE B.6 – Evaluation de la métrique 5 : Réalisation des frames

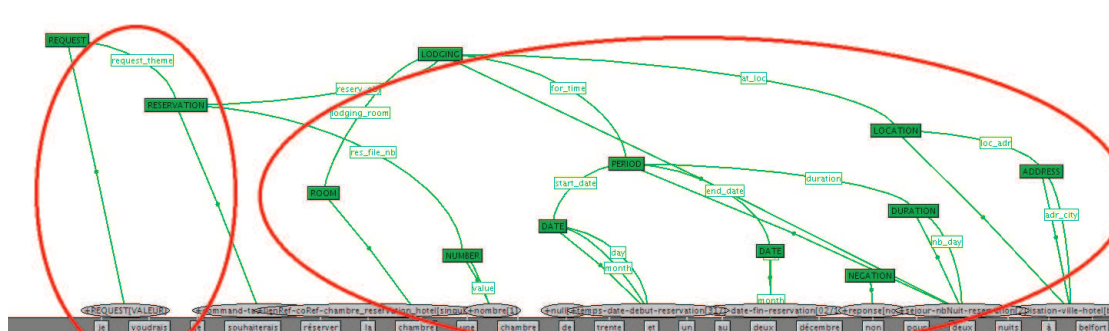


FIGURE B.7 – Evaluation de la métrique 6 : Composition des frames

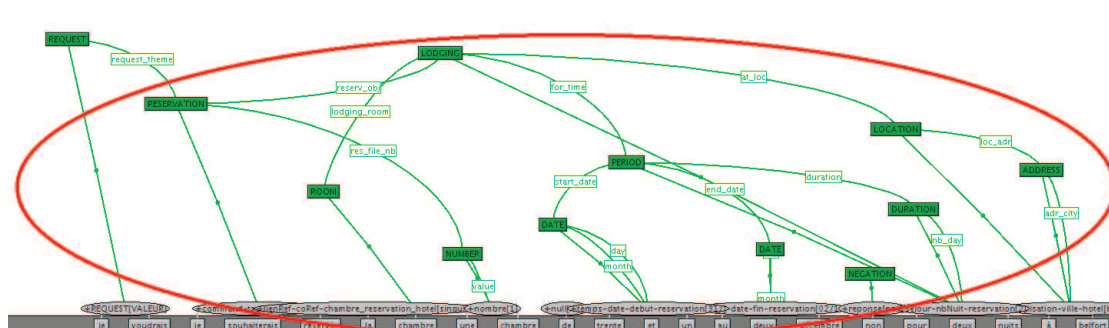


FIGURE B.8 – Evaluation de la métrique 7 : Analyse complète du tour



# Glossaire

- A.T.I.S.** Air Travel Information System.
- A.T.N.G.** Augmented Transition Network Grammars, cf. [G.R.T.A.](#) .
- C.E.R.** Taux d'Erreurs Concepts (Concept Error Rate).
- C.F.G.** Grammaire indépendante du contexte.
- C.L.P.** Compréhension du Langage Parlé.
- C.R.F.** Conditional Random Fields.
- D.B.N.** Réseaux Bayésiens Dynamiques (Dynamic Bayesian Networks).
- D.C.G.** Definite Clause Grammar, cf [G.C.D.](#) .
- E.G.** Grammaire sémantique indépendante Embarquée.
- E.M.** Entropie Maximum.
- F.L.M.** Modèle de Langage Factorisé.
- F.S.M.** Machine à états finis (Final State Machine).
- F.S.T.** Transducteur à états finis.
- G.C.C.** Grammaire Catégorielle Combinatoire.
- G.C.D.** Grammaires à Clauses Définies.
- G.R.T.** Grammaire à Réseaux de Transitions.
- G.R.T.A.** Grammaire à Réseaux de Transitions Augmentées.
- GEN.L.M.** Modèles de langage générique.
- H.M.M.** Modèles de Markov Cachés, (Hidden Markov Model).
- L.H.C.** Langage Hors-Contexte.
- L.R.S.** Langage de Représentation Sémantique.
- L.S.A.** analyse sémantique latente (Latent Semantic Analysis).
- L.T.A.G.** Grammaire lexicalisée d'arbres adjoints (Lexicalised Tree Adjoining Grammar).

**M.C.E.** Erreurs de Classification Minimum.

**M.E.L.M.** Modèle de Langage d'Entropie Maximum.

**M.L.** Modèle de Langage.

**M.L.C.** Modèle de Langage Conceptuel.

**M.L.M.** Modèles à Grandes Marges.

**M.P.** Modèle de Phrase.

**P.O.S.** Partie du discours (Part of Speech).

**R.A.P.** Reconnaissance Automatique de la Parole.

**S.C.F.G.** Grammaire stochastique indépendante du contexte.

**S.C.T.** Arbre de Classification Sémantique (Semantic Classification Tree).

**S.F.S.M.** Machine à états finis stochastique (Statistical Final State Machine).

**S.F.S.T.** Transducteur stochastique à états finis.

**S.M.T.** Traduction automatique statistique (Statistical Machine Translation).

**S.T.D.S.** Schéma de Traduction Dirigé par la Syntaxe.

**S.V.M.** Machine à Vecteurs de Supports (Support Vector Machine).

**T.N.G.** Transition Network Grammars, cf. [G.R.T.](#) .

**W.E.R.** Taux d'Erreurs Mots (Word Error Rate).

