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

École doctorale : Sciences et Agronomie

THÈSE

INTÉGRATION DE SOURCES DE CONNAISSANCES POUR LA MODÉLISATION STOCHASTIQUE DU LANGAGE APPLIQUÉE A LA PAROLE CONTINUE DANS UN CONTEXTE DE DIALOGUE ORAL HOMME-MACHINE

Présentée et soutenue publiquement le 28 novembre 2002
pour obtenir le grade de Docteur en Sciences
de l'Université d'Avignon et des Pays de Vaucluse

SPÉCIALITÉ : INFORMATIQUE

par

Yannick ESTÈVE

Composition du jury :

M	Marc EL-BÈZE	Professeur, LIA, Avignon	Président
MM	Pietro LAFACE	Professeur, Politecnico, Turin (Italie)	Rapporteur
	Kamel SMAILI	Professeur, LORIA, Nancy	Rapporteur
M	Denis JOUVET	Ingénieur, FTRD, Lannion	Examinateur
MM	Renato DE MORI	Professeur, LIA, Avignon	Directeur de thèse
	Frédéric BÉCHET	Maître de Conférences, LIA, Avignon	Co-Directeur de thèse

Laboratoire d'Informatique d'Avignon

Résumé

Les modèles de langage sont utilisés dans un système de reconnaissance de la parole pour guider le décodage acoustique en apportant des contraintes linguistiques. Ces modèles de langage sont généralement des modèles de langage probabilistes. Les modèles de langage *n-grams*, qui constituent les modèles de langage de référence en reconnaissance de la parole, modélisent des contraintes sur n mots à partir d'événements observés dans un corpus d'apprentissage. Ces modèles donnent des résultats satisfaisants car ils profitent d'une caractéristique commune à plusieurs langues, dont le français, qui exercent des contraintes locales fortes sur l'ordre des mots. Ils arrivent ainsi à résumer simultanément une grande partie des connaissances syntaxiques et sémantiques issues de l'observation du corpus d'apprentissage.

Malheureusement, l'utilisation de ces modèles probabilistes est confrontée à plusieurs difficultés. Une faible quantité de données d'apprentissage est courante lors du développement de nouvelles applications de reconnaissance de la parole et entraîne l'estimation de modèles probabilistes peu robustes, dont le comportement est biaisé par manque d'informations statistiques. Une autre difficulté vient de la longueur des contraintes modélisées. Même si il existe des contraintes locales très fortes propices à l'utilisation de modèles *n-grams*, certaines contraintes linguistiques portent sur des distances supérieures à leurs capacités de modélisation.

Afin de pallier les difficultés des modèles *n-grams*, nous proposons d'utiliser plusieurs sources de connaissances *a priori*. Ces connaissances sont injectées à plusieurs niveaux. Nous proposons un modèle hybride qui combine un modèle de langage *n-gram* avec des grammaires régulières locales : les connaissances linguistiques apportées par ces grammaires sont directement intégrées dans le modèle. Des connaissances *a priori* sont également exploitées pour la création de modèles de langage *n-grams* spécialisés et pour leur utilisation au cours d'un dialogue oral homme-machine. De même, l'analyse des caractéristiques des hypothèses issues de différents systèmes de reconnaissance utilise diverses sources de connaissances. Cette analyse permet de choisir l'hypothèse de reconnaissance la plus pertinente ou de rejeter l'ensemble des hypothèses proposées. Enfin, des connaissances *a priori* sont prises en compte pour élaborer des critères de consistance linguistique. Ces critères permettent de détecter certains types d'erreurs qui peuvent être corrigées à l'aide de modèles de langage très spécifiques, appelés modèles stratégiques.

Remerciements

C'est un plaisir que de pouvoir signifier ici ma gratitude envers les personnes qui m'ont aidé à commettre cette thèse.

Je tiens à remercier encore Messieurs Pietro LAFACE et Kamel SMAÏLI d'avoir accepté d'être les rapporteurs de ce mémoire. C'est une tâche ardue, et je suis conscient de la charge de travail qu'elle a représenté. De la même manière, je remercie Messieurs Marc EL-BÈZE et Denis JOUVET qui en plus de leur participation au jury ont rigoureusement contribué à la correction et l'amélioration de ce mémoire.

Je rends hommage à Monsieur Frédéric BÉCHET pour sa présence à mes côtés tout au long de cette thèse. Sa disponibilité, même à (très longue) distance fut d'un très grand réconfort. Son aide fut précieuse, et travailler avec lui un réel plaisir.

J'ai été extrêmement heureux d'avoir Monsieur Renato DE MORI comme directeur de thèse. Pouvoir le côtoyer quotidiennement a été une chance pour moi. Je le remercie de m'avoir consacré autant de temps et de m'avoir fait profiter de son expérience et de son savoir.

De façon générale, je remercie tous les membres du personnel du Laboratoire d'Informatique d'Avignon.

Plus particulièrement, merci à Thierry SPRIET pour sa bonne humeur communicative et pour l'aide qu'il m'a apporté. Merci aux doctorants d'hier et d'aujourd'hui : Corinne FREDOUILLE, Brigitte BIGI, Sylvain MEIGNIER, David JANISZEK, Christian RAYMOND, Loïc LEFORT, Dominique MASSONIE, Christophe LÉVY ... Une mention spéciale pour Teva MERLIN dont j'ai pu apprécier le secours en de nombreuses circonstances (come get some !).

Je remercie France Télécom Recherche et Développement qui nous a fourni le matériau nécessaire à cette étude. Particulièrement, merci aux membres du laboratoire IPS et à toutes les personnes que j'ai eu la chance de rencontrer lors de mes brefs séjours à Lannion.

Aussi, je tiens à remercier toutes les personnes qui n'ont pas participé directement aux travaux exposés dans ce mémoire mais qui m'ont aidé à franchir les obstacles.

Toute mon affection va à Monsieur et Madame LALOUX, ainsi qu'à Rodolphe et sa grand-mère. Ils sont toujours présents dans mon coeur en excellente compagnie.

Je remercie bien sûr mes parents, mes soeurs, et le reste de ma famille sur laquelle il est très réconfortant de pouvoir s'appuyer.

Je remercie mes amis David, Jean-Mi, Serge, Sissi, Titi, Fanny, Régis, etc. Je remercie Tiroir Rouge, Ligeya, B'n'B, ainsi que Metallica, Iron Maiden, ZZ Top, AC/DC et Georges Brassens pour ne citer qu'eux.

Bien entendu, je n'oublie pas de remercier Antoinette pour son accueil et son hospitalité.

Enfin, un énorme merci à Stéphanie qui n'a jamais hésité à faire quelques sacrifices, et ne m'en a jamais tenu rigueur, afin que ce mémoire soit terminé dans des délais raisonnables.

À la mémoire d'Audrey

Table des matières

I Modélisation du langage pour la reconnaissance de la parole : état de l'art **7**

1 Modèles de langage probabilistes **9**

1.1 Approximations *n-grams* 10

1.1.1 Modèles *n-grams* 10

1.1.2 Modèles *n-classes* 11

1.1.3 Modèles à séquences de longueurs variables 13

1.1.4 Discussion 14

1.2 Estimation des paramètres 15

1.2.1 Distribution paramétrique 16

1.2.2 Estimation par Maximum de Vraisemblance 17

1.2.3 Estimation Bayésienne 17

1.2.4 Estimation par Maximum a Posteriori (MAP) 18

1.3 Lissage 18

1.3.1 Principe 19

1.3.2 Lissage par repli (*backing-off*) 20

1.3.3 Lissage par interpolation 20

1.3.4 Techniques de *discounting* 21

1.4 Évaluation d'un modèle de langage 24

1.4.1 Rappels sur la théorie de l'information 24

1.4.2 Définition de la perplexité 25

1.4.3 Perplexité et modèle de langage 25

2 Approche formelle et autres modèles de langage 27

2.1 Grammaires formelles 28

2.1.1 Classification de Chomsky 29

2.1.2 Grammaires et automates 30

2.1.3 Automates à états finis 30

2.1.4 Automates de type pushdown 31

2.2 Grammaires formelles vs. modèles de langage stochastiques 31

2.2.1 Couverture 32

2.2.2 Construction 32

2.2.3 Longueur des contraintes 32

2.3 Méthodes mixtes 33

2.3.1 Grammaires probabilistes 33

2.3.2 Contraintes syntaxiques intégrées dans un système d'étiquetage probabiliste 34

2.3.3 Modèle de langage basé sur des automates stochastiques à *n-grams* variables 35

3 Modèles de langage et moteurs de reconnaissance de la parole 39

3.1 Reconnaissance de la parole : notions de base 39

3.1.1 Combinaison des modèles acoustiques et des modèles de langage 40

3.1.2 Espace de recherche et graphe de mots 41

3.2 Exemples de modèles de langage utilisés en seconde passe 42

3.2.1 Modèles *n-grams* 43

3.2.2 Modèles de langage à base de classes syntaxiques 44

3.2.3 Modèles de langage structurés 44

3.3 Adaptation d'un modèle de langage 45

3.3.1 Motivations 45

3.3.2 Acquisition des données d'adaptation 46

3.3.3 Techniques d'adaptation 47

3.3.4 Discussion 51

3.4 Évaluation d'un système de reconnaissance de la parole 52

3.4.1 Taux d'erreurs sur les mots 52

3.4.2 Intervalle de confiance 53

3.5 Présentation générale d'un système de dialogue 53

3.5.1 Architecture modulaire 53

3.5.2 Spécificités de la reconnaissance de la parole dans une application de dialogue 55

II Contributions **57**

4 Intégration d'automates stochastiques à états finis dans un modèle *n-gram* **59**

4.1 Introduction 60

4.1.1 Motivations 60

4.1.2 Contexte de l'étude 61

4.1.3 Automates à états finis et classes de séquences de mots 61

4.2 Architecture du modèle 63

4.2.1 Composantes 63

4.2.2 Combinaison des composantes 63

4.3 Probabilité d'une phrase selon le modèle intégré 64

4.3.1 Intégration d'un automate stochastique dans un modèle *n-gram* 64

4.3.2 Généralisation 65

4.3.3 Exemple 68

4.4 Estimation des paramètres du modèle 69

4.4.1 Modèles *internes* 69

4.4.2 Modèle *externe* 74

4.5 Utilisation du modèle hybride dans un système de reconnaissance de la parole 76

4.5.1 Utilisation du modèle hybride 76

4.5.2 Algorithme de détection d'une séquence de mots spécifique . . . 76

4.5.3 Séquence de mots détectée : insertion d'une transition 77

4.6 Expérimentations 78

4.6.1 Description des données expérimentales 78

4.6.2 Evaluation des modèles de langage 82

5 Sélection dynamique de modèles de langage **91**

5.1 Motivations 91

5.1.1 Travaux existants 91

5.1.2 Particularités de l'étude et propositions 92

5.2 Construction de modèles spécifiques 93

5.2.1 Étiquetage des phrases à l'aide de connaissances *a priori* 94

5.2.2 Utilisation d'un arbre de classification sémantique 95

5.3 Sélection des modèles 102
5.3.1 Modèles construits par étiquetage des phrases 102
5.3.2 Modèles construits à partir d'un arbre de classification sémantique 103
5.4 Expérimentation 104
5.4.1 Apprentissage des modèles de langages spécifiques 105
5.4.2 Perplexité 106
5.4.3 Reconnaissance de la parole 107
5.5 Conclusions 112

6 Utilisation d'hypothèses de reconnaissance issues de systèmes différents : choix, validation, rejet 113
6.1 ROVER 114
6.1.1 Présentation générale 114
6.1.2 Alignement de plusieurs hypothèses 114
6.1.3 Vote 116
6.1.4 Quelques résultats 117
6.2 Proposition 117
6.2.1 Informations utilisées 118
6.2.2 Arbre de décision 119
6.3 Expérimentation 122
6.3.1 Utilisation de la technique dite de *leave-one-out* 122
6.3.2 Rappel des performances des différents modèles de langage présentés dans ce mémoire 123
6.3.3 Prise de décision pour deux systèmes de reconnaissance mis en concurrence (modèles *trigrams*) 124
6.3.4 Prise de décision pour les quatre systèmes de reconnaissance à base de modèles bigrams mis en concurrence 127
6.4 Discussion 132

7 Utilisation stratégique de modèles de langage 135
7.1 Stratégies 136
7.1.1 Raisonnement sur les résultats et détermination de la stratégie 136
7.1.2 Mesure de consistance 137
7.1.3 Apprentissage par l'exemple *(Explanation-based learning)* 138

7.1.4 Déclencheurs 139
7.2 Modèles stratégiques 139
7.2.1 Modèles *n-grams* adaptés par génération de *n-grams* plausibles 140
7.2.2 Une variante : la dépréciation des *n-grams* peu plausibles 141
7.2.3 Modèles *n-grams* adaptés par augmentation de données 141
7.2.4 Modèles *n-grams* à automates spécifiques 142
7.2.5 Combinaison d'un modèle *n-gram* et d'un arbre de classification sémantique 143
7.3 Expérimentations 145
7.3.1 Définition des contraintes de consistance 146
7.3.2 Bilan 147

8 Conclusions et Perspectives 151
8.1 Bilan 151
8.2 Perspectives 153

A Liste des étiquettes syntaxiques utilisées et leur signification 155

B Grammaire utilisée pour l'analyse grammaticale partielle du corpus d'apprentissage 157

Bibliographie 163

Références bibliographiques personnelles 171

Table des figures

2.1 *Représentation de la structure syntaxique d'une phrase sous forme d'arbre* . . 28

2.2 *Exemple d'automate à états finis (automates réguliers)* 31

2.3 *Représentation d'une partie d'un automate stochastique à n-grams variables du troisième ordre* 37

3.1 *Exemple simple de graphe de mots* 43

3.2 *Schéma général du processus d'adaptation d'un modèle de langage* 46

3.3 *Architecture général d'un système de dialogue* 54

4.1 *Classe de séquences de mots et exemple d'automate lexical à états finis associé.* 62

4.2 *Architecture du modèle de langage* 64

4.3 *Exemple de segmentations possibles d'une séquence de mots* 67

4.4 *Apprentissage des modèles internes (ou locaux)* 69

4.5 *Extraction des syntagmes* 71

4.6 *Fusion de classes* 72

4.7 *Processus détaillé d'apprentissage des modèles internes* 75

4.8 *Apprentissage du modèle externe* 76

4.9 *Répartition des phrases du corpus d'apprentissage en fonction du nombre de mots qui les composent* 79

4.10 *Répartition des phrases de référence du corpus de test en fonction du nombre de mots qui les composent* 80

4.11 *Comparaison des perplexités des différents modèles n-grams et modèles hybrides sur les corpus d'apprentissage et de test* 82

4.12 *Évolution du taux d'erreur sur les mots en fonction de la valeur du fudge factor (Modèle bigram, Graphes I)* 84

4.13 *Comparaison des taux d'erreurs sur les mots des différents modèles n-grams et modèles hybrides sur les deux ensembles de graphes* 85

5.1 *Architecture générale de l'utilisation de modèles de langage spécialisés* 93

5.2 *Schéma général de construction des modèles de langage spécialisés* 94

5.3 *Exemple d'arbre de décision construit manuellement à partir de connaissances a priori et de l'observation du corpus d'apprentissage afin de classer les phrases en quatre catégories prédéfinies* 96

5.4 *Exemple abrégé d'un arbre de classification sémantique* 100

5.5 *Exemple de parcours d'un arbre de classification sémantique par une liste d'hypothèses produites en première passe pour le choix d'un modèle de langage spécialisé* 104

5.6 *Évolution du taux d'erreurs sur les mots en fonction la taille de la liste de n-best utilisée pour la sélection des modèles spécialisés* 111

6.1 *Architecture du système ROVER* 114

6.2 *Exemple d'alignement de trois hypothèses issues de trois systèmes de reconnaissance de la parole.* 115

6.3 *Pourcentage d'ensembles d'hypothèses rejetés en fonction de la valeur du seuil d'acceptation utilisé lors de la construction de l'arbre de décision (modèles trigrams, Graphes I)* 125

6.4 *Comparaison du taux d'erreurs sur les mots des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'ensemble d'hypothèses rejetés (Modèles trigrams, Graphes I)* 126

6.5 *Comparaison du taux d'erreurs sur les phrases des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles trigrams, Graphes I)* 127

6.6 *Proportion d'hypothèses de niveau* 1 *et d'hypothèses de niveau* 2 *validées en fonction du pourcentage d'hypothèses rejetées (Modèles trigrams, Graphes I)* . 128

6.7 *Taux d'erreurs sur les mots des phrases de niveau* 1 *(phrases validées et toutes les hypothèses sont identiques) et des phrases de niveau* 2 *(phrases validées, mais un choix a du être fait sur l'hypothèse finale) en fonction du pourcentage de phrases rejetées (Modèles trigrams, Graphes I)* 129

6.8 *Pourcentage d'ensembles d'hypothèses rejetés en fonction de la valeur du seuil d'acceptation utilisé lors de la construction de l'arbre de décision (Modèles bigrams, Graphes I)* 130

6.9 *Comparaison du taux d'erreurs sur les mots des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)* 131

6.10 *Comparaison du taux d'erreurs sur les phrases des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)* 131

6.11 *Proportion d'hypothèses de niveau* 1 *et d'hypothèses de niveau* 2 *validées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)* . 132

6.12 *Taux d'erreurs sur les mots des phrases de niveau* 1 *(phrases validées et toutes les hypothèses sont identiques) et des phrases de niveau* 2 *(phrases validées, mais un choix a du être fait sur l'hypothèse finale) en fonction du pourcentage de phrases rejetées (Modèles bigrams, Graphes I)* 133

7.1 *Arbre logique de résolution du remplacement de la séquence de mots* m *par la séquence de mots* n *dans le contexte* (a, b) 139

7.2 *Exemple d'arbre de classification utilisé pour la désambiguisation de syntagmes quasi-homophones* 144

7.3 *Résultats des expériences sur l'application de contraintes de consistance et l'utilisation de divers types de modèles de langage* 148

Liste des tableaux

2.1 *Types d'automates acceptant les langages engendrés par les différentes grammaires* 30

4.1 *Phrases de test contenant au moins un mot hors-vocabulaire ou un mot non vu lors de l'apprentissage* 81

4.2 *Répartition des graphes du corpus de test en fonction de la présence ou de l'absence de la phrase de référence dans ces graphes* 81

4.3 *Répartition des mots, des phrases et des sessions du corpus de test en fonction du locuteur* 81

4.4 *Intervalles de confiance à 95% des différents taux d'erreurs observés sur le corpus de test* 86

4.5 *Taux d'erreurs sur les mots en fonction de la présence ou non de la phrase de référence dans les graphes pour l'ensemble des graphes I* 87

4.6 *Taux d'erreurs sur les mots en fonction de la présence ou non de la phrase de référence dans les graphes pour l'ensemble des graphes II* 87

4.7 *Comparaison du taux d'erreurs obtenu à l'aide de modèles de langage bigrams classiques ou avec automates en fonction de l'existence ou de l'absence de la phrase de référence dans les graphes I et II* 88

4.8 *Comparaison du taux d'erreurs obtenu à l'aide de modèles de langage trigrams classiques ou avec automates en fonction de l'existence ou de l'absence de la phrase de référence dans les graphes I et II* 88

5.1 *Répartition des phrases du corpus d'apprentissage en fonction de l'étiquetage effectué par l'arbre de décision construit manuellement* 105

5.2 *Répartition des phrases de référence du corpus de test en fonction de l'étiquetage effectué par l'arbre de décision utilisé pour scinder le corpus d'apprentissage* 105

5.3 *Comparaison de la perplexité obtenue par les différents modèles bigrams sur les corpora d'apprentissage et de test.* 106

5.4 *Taux d'étiquetage correct des phrases reconnues en première passe à l'aide d'un modèle généraliste bigram en fonction de l'ensemble de graphes utilisé pour le décodage* 107

5.5 *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé* 108

5.6 *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi à partir de l'étiquette attribuée à la phrase de référence* 108

5.7 *Comparaison des taux d'erreurs sur les mots obtenus par le système de reconnaissance avec le modèle bigram général ou avec un modèle bigram spécialisé choisi en fonction de la catégorie attribuée à l'hypothèse de première passe H_1 pour les graphes de l'ensemble I* 109

5.8 *Comparaison des taux d'erreurs sur les mots obtenus par le système de reconnaissance avec le modèle bigram général ou un modèle bigram spécialisé choisi en fonction de la catégorie de la phrase de référence pour les graphes de l'ensemble I* 109

5.9 *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi par la méthode $Select_{Pattern}$* 110

5.10 *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi par la méthode $Select_{PP}$* 111

6.1 *Tableau récapitulatif des taux d'erreurs (sur les mots) obtenus sur le corpus de test selon le modèle bigram utilisé présenté dans ce mémoire* 124

6.2 *Tableau récapitulatif des taux d'erreurs (sur les mots) obtenus sur le corpus de test selon le modèle trigram utilisé présenté dans ce mémoire* 124

7.1 *Taux d'erreurs sur les mots et nombre de phrases en fonction de la mesure de consistance $CONS(LM)$* 138

Glossaire et notations

Glossaire

AGS : Audiotel Guide des Services

FTRD : France Télécom Recherche & Développement

Graphe de type I : graphe de mots fortement élagué

Graphe de type II : graphe de mots moins élagué

LIA : Laboratoire d'Informatique d'Avignon

MAP : maximum *a posteriori*

MDI : information de discrimination minimale (*minimum discrimination information*)

ME : maximum d'entropie (*maximum entropy*)

MV : maximum de vraisemblance

ROVER : *recognizer output voting error reduction*

RTN : réseau récursif de transitions (*recursive transition network*)

SCT : arbre de classification sémantique (*semantic classification tree*)

VNSA : automate stochastique à *n-grams* variables (*variable n-gram stochastic automaton*)

Notations

$A_k(W_i^j)$: séquence de mots W_i^j acceptée par l'automate A_k

$G(t_i)$: index de Gini du noeud t_i (mesure d'impureté)

$H(s)$: entropie de la source d'émissions s

$H(t_i)$: entropie du noeud t_i (mesure d'impureté)

$I(t_i)$: impureté du noeud t_i

$I(\sigma_t)$: quantité d'information liée à l'émission de σ_t

LM_{Cat} : modèles de langage spécialisés estimés à partir de scission du corpus d'apprentissage par connaissances *a priori*

LM_{Stat} : modèles de langage spécialisés estimés à partir de scission du corpus d'apprentissage par informations statistiques$N_C(S_i)$: nombre d'occurrences du syntagme S_i dans la classe C observées sur le corpus d'apprentissage

$P(W, \varsigma_z)$: probabilité de la séquence de mot W sous la forme segmentée ς_z

PP : perplexité

W_1^k : séquence de k mots indexés de 1 à k

X : séquence d'observations acoustiques

$c(w)$: nombre d'occurrences du mot w dans le corpus d'apprentissage

$c^*(hw)$: fréquence corrigée de la séquence de mots hw

$c'(hw)$: fréquence modifiée de la séquence de mots hw

$d(C_i, C_j)$: mesure de distance entre les classes de syntagmes C_i et C_j

$fr(w|h)$: fréquence conditionnelle relative du mot w selon l'historique h dans le corpus d'apprentissage

$fr^*(w|h)$: fréquence conditionnelle décomptée du mot w selon l'historique h dans le corpus d'apprentissage

h_i : historique du i^{me} élément d'une séquence

lw : poids linguistique (*fudge factor*)

lp : pénalite linguistique

$s.e.r$: taux d'erreurs sur les phrases (*sentence error rate*)

w_i : i^{me} mot d'une séquence de mots

$w.e.r$: taux d'erreurs sur les mots (*word error rate*)

$\lambda(h)$: probabilité de fréquence nulle pour l'historique h

Δ_I : réduction d'impureté

Θ : espace de paramètres

θ_w : vecteur de paramètres du mot w

θ^{MAP} : estimateur par maximum *a posteriori* du paramètre θ

θ^{MV} : estimateur par maximum de vraisemblance du paramètre θ

θ^B : estimateur bayésien du paramètre θ

Ξ : ensemble de séquence d'états d'un automate

$< UNK >$: mot inconnu

Introduction

Parler avec une machine. Cette idée folle très répandue chez les auteurs de science-fiction des années 50 devient réalité avec l'entrée de nos sociétés modernes dans l'ère de l'informatique. La capacité pour l'homme de diriger une machine par la voix et dans sa langue naturelle est rendue possible grâce aux technologies de reconnaissance automatique de la parole. La reconnaissance de la parole a pour objectif la transformation des sons émis par un locuteur en une séquence de mots qui correspond à celle qu'il a prononcée. Les applications qui en découlent sont nombreuses, et certaines déjà utilisées : commande vocale de machines (fauteuils roulants par exemple), dictée vocale, indexation de discours, d'enregistrements audiovisuels, etc.

Il est d'autant plus facile de constater les limites actuelles des technologies de reconnaissance de la parole que celles-ci s'insèrent dans notre quotidien : toutes les personnes qui utilisent (ou ont essayé d'utiliser) un système de dictée vocale ont une idée du chemin qu'il reste à parcourir avant que ce type d'applications ne donne entière satisfaction à ses utilisateurs.

Les technologies de reconnaissance de la parole restent encore limitées mais dans certaines circonstances (milieu non bruité, vocabulaire réduit, thème restreint) leurs performances deviennent exploitables. Dans ce contexte, les applications de dialogues oraux homme-machine donnent des résultats très intéressants.

Modélisation du langage

Les modèles de langage sont un des nombreux constituants d'un système de reconnaissance de la parole. La modélisation du langage a pour objectif de résumer les connaissances générales liées à un langage naturel. En reconnaissance de la parole, la modélisation du langage guide la recherche de l'hypothèse optimale en exerçant des contraintes linguistiques sur les hypothèses acoustiques.

Deux types de modèles de langage se distinguent : les modèles à base de connaissance et les modèles probabilistes.

Les premiers nécessitent une expertise linguistique pour leur construction, et se composent généralement de grammaires formelles. Le principal inconvénient de leur utilisation dans un système de reconnaissance de la parole est leur manque

de couverture du langage : ces modèles sont confrontés aux phénomènes d'agrammaticalité qui touchent régulièrement les phrases prononcées en parole spontanée. Malgré l'ampleur de leurs connaissances linguistiques, ces modèles manquent de souplesse et peuvent rejeter à tort des hypothèses de reconnaissance correctes lorsque le locuteur s'exprime en commettant quelques fautes de grammaire : seules les phrases grammaticalement correctes, *i.e.* conformes à la grammaire formelle qui définit le modèle, peuvent être acceptées.

Les modèles probabilistes présentent plusieurs avantages par rapport aux modèles à base de connaissance. Par essence, ils apportent une information quantifiée sur la validité d'une hypothèse au contraire des modèles précédents dont l'apport est limité à une information de type " rejet/acceptation ". Et, surtout, les modèles de langage stochastiques offrent une couverture complète du langage visé. Ils n'excluent aucune construction syntaxique et autorisent la reconnaissance de phrases ne respectant pas les règles de grammaire régissant le langage. Le principal défaut des modèles de langage statistiques est la quantité importante de données nécessaires à leur construction, au contraire des modèles de langage à base de connaissance. De plus, les modèles à base de connaissance intègrent des informations syntaxiques, voire sémantiques, inexistantes de manière explicite dans les modèles probabilistes.

Les modèles de langage stochastiques *n-grams* sont les modèles les plus utilisés dans les systèmes de reconnaissance de la parole. En 1991, (Jelinek, 1991) notait avec surprise que quinze ans après leur première utilisation dans un système de reconnaissance (Bahl *et al.*, 1978), les modèles *trigrams* restaient toujours les modèles de référence.

Onze ans plus tard, soit vingt-six ans après leurs premières expérimentations, ces modèles n'ont toujours pas été détrônés. Pourtant, de nouveaux modèles ont été proposés, comme par exemple les modèles basés sur des arbres de décision (Bahl *et al.*, 1990), les modèles de langage structurés (Chelba et Jelinek, 2000,), les modèles *n-multigrams* (Deligne et Bimbot, 1995,) ou les variantes des modèles *n-grams* que sont les modèles à mémoire cache (Kuhn et De Mori, 1990,). Cependant, les modèles *n-grams* restent très majoritairement les modèles de langage les plus répandus dans les systèmes de reconnaissance. La raison principale est certainement la grande simplicité d'utilisation des modèles *n-grams* que n'offrent pas ces nouveaux modèles. Les faibles améliorations des performances des systèmes de reconnaissance apportées par les nouveaux modèles ne justifient pas l'abandon de la simplicité d'utilisation des modèles *n-grams*. Pourtant, au niveau de la modélisation du langage, il est envisageable d'obtenir des résultats substantiellement supérieurs à ceux obtenus avec les *trigrams*, comme le décrit (Jelinek, 1991) : un être humain est capable de détecter et de corriger un grand nombre d'erreurs d'une hypothèse de reconnaissance sans détenir aucune autre information que la transcription écrite de cette hypothèse.

Une des tendances actuelles des travaux de recherche en modélisation du langage concerne la combinaison des deux approches formelles et statistiques. L'objectif de ces travaux est de réunir les avantages de la modélisation statistique (couverture, souplesse) avec ceux de la modélisation à base de connaissance (informations *a*

priori, peu de données d'apprentissage).

Apports de la thèse

Les travaux proposés dans cette thèse se situent au niveau de la modélisation probabiliste du langage appliquée à la reconnaissance de la parole. Plus précisément, ils se situent dans un contexte de dialogue oral homme-machine.

Nous avons cherché à intégrer des connaissances issues de diverses sources d'information (connaissances grammaticales ou sémantiques, connaissances concernant le dialogue) à plusieurs niveaux de la modélisation statistique.

Notre première contribution consiste en la présentation d'un nouveau modèle de langage d'essence statistique. Ce modèle intègre des grammaires régulières locales au sein d'un modèle stochastique de type *n-gram*. L'intégration de grammaires régulières locales permet de modéliser des événements non vus dans le corpus d'apprentissage, ce qui est un problème récurrent en modélisation statistique. De plus, l'intégration de ces grammaires, sous la forme d'automates stochastiques à états finis, permet d'injecter de l'information linguistique *a priori*. Elle permet également d'étendre la taille des contraintes entre les mots, généralement limitée à *n* mots pour un modèle *n-gram*.

Nous avons aussi travaillé sur la construction de modèles de langage *n-grams* spécialisés dans plusieurs types de phrases. Ces types de phrases sont définis soit à l'aide de connaissances *a priori*, soit de manière probabiliste. Ces phrases peuvent également correspondre à un état particulier du dialogue. Une question importante concerne l'utilisation de ce type de modèles spécifiques : il s'agit de sélectionner dynamiquement le modèle de langage spécialisé le plus approprié tout au long du dialogue.

L'utilisation de plusieurs modèles de langage dans un système de reconnaissance de la parole permet d'obtenir plusieurs hypothèses de reconnaissance. Nous proposons un système de décision qui permet de choisir l'hypothèse la plus pertinente, ou de rejeter l'ensemble de ces hypothèses. La décision est prise après analyse des caractéristiques de chacune des hypothèses. Ces caractéristiques dépassent le cadre des informations statistiques puisqu'elles peuvent prendre en compte des informations concernant la structure syntaxique de l'hypothèse, son type de phrase, etc.

Enfin, nous ouvrons de nouvelles perspectives sur l'utilisation des modèles de langage pour la reconnaissance de la parole. Partant du constat que chaque modèle a des qualités propres et que les modèles *n-grams* sont performants mais qu'il est possible d'aller au-delà de leurs limites, nous utilisons des modèles de langage construits dans le but précis de répondre à des problèmes spécifiques. Nous proposons la création de modèles de langage dont le champ d'action est limité, et ne répond qu'à un problème particulier. L'utilisation de tels modèles de langage ne concerne que les hypothèses de reconnaissance pour lesquelles des irrégularités ou

des risques d'irrégularités ont été détectées. Ainsi, la qualité de ces modèles de langage stratégiques doit être mesurée non pas sur l'ensemble des phrases reconnues, mais uniquement lorsque ces modèles ont été utilisés.

Organisation du mémoire

Le mémoire est divisé en deux grandes parties.

La première partie propose un survol de l'état de l'art des techniques de modélisation du langage. Trois chapitres composent cette partie :

1. Le premier chapitre traite de la modélisation statistique du langage, et en particulier des modèles *n-grams*. Les techniques d'estimation des paramètres de ces modèles sont présentées, ainsi que les techniques les plus courantes de lissage qui permettent de modéliser des événements non vus pendant l'apprentissage. La mesure de la qualité d'un modèle de langage est également abordée.
2. Le second chapitre aborde les modèles de langage à base de connaissance et discute des qualités comparées de ces modèles et des modèles probabilistes. Quelques approches mixtes sont alors évoquées.
3. Dans le chapitre trois, nous nous intéressons à l'intégration d'un modèle de langage probabiliste au sein d'un système de reconnaissance de la parole. Il sera également question de modèles de langage utilisés en phase de *rescoring* et de l'adaptation des modèles de langage. L'évaluation d'un système de reconnaissance sera aussi abordée. Enfin, l'architecture générale d'un système de dialogue sera présenté.

La seconde partie du mémoire concernent les travaux proposés par cette thèse, déjà évoqués précédemment. Elle est composée de quatre chapitres :

1. Le chapitre quatre présente un modèle de langage hybride : il combine des automates stochastiques à états finis, représentant des grammaires régulières locales, au sein d'un modèle de langage *n-gram*.
2. Le chapitre cinq concerne la création et l'utilisation de modèles de langage spécialisés. Ces modèles sont spécifiques à certains types de phrases qui apparaissent au cours du dialogue. Une sélection dynamique de ces modèles est donc nécessaire.
3. Le système de décision présenté dans le chapitre six permet de choisir une hypothèse de reconnaissance parmi les diverses hypothèses présentées par un ensemble de modules de reconnaissance différents. Le système de décision peut également rejeter en bloc l'ensemble des hypothèses.
4. Enfin, le chapitre sept ouvre de nouvelles perspectives sur l'utilisation des modèles de langage. Il s'agit de créer des modèles de langage de différentes natures qui permettent d'agir sur certains problèmes spécifiques. A l'aide d'un autre système de décision, basé sur des critères de consistance linguistique

prédéfinis, il est possible de déterminer les hypothèses qui semblent présenter des irrégularités, ou qui risquent d'en présenter. Différents types d'irrégularités sont analysés, et si c'est possible un modèle de langage adéquat est utilisé pour une dernière phase de *rescoring*.

Pour chacun des chapitres de la seconde partie du mémoire, des résultats d'expériences menées sur une application de dialogue oral homme-machine sont présentés. L'application de dialogue en question est le démonstrateur AGS (Audiotel Guide des Services) créé par France Télécom Recherche & Développement.

Première partie

Modélisation du langage pour la reconnaissance de la parole : état de l'art

Chapitre 1

Modèles de langage probabilistes

Sommaire

1.1 Approximations *n-grams* 10
1.1.1 Modèles *n-grams* 10
1.1.2 Modèles *n-classes* 11
1.1.3 Modèles à séquences de longueurs variables 13
1.1.4 Discussion 14
1.2 Estimation des paramètres 15
1.2.1 Distribution paramétrique 16
1.2.2 Estimation par Maximum de Vraisemblance 17
1.2.3 Estimation Bayésienne 17
1.2.4 Estimation par Maximum a Posteriori (MAP) 18
1.3 Lissage 18
1.3.1 Principe 19
1.3.2 Lissage par repli (*backing-off*) 20
1.3.3 Lissage par interpolation 20
1.3.4 Techniques de *discounting* 21
1.4 Évaluation d'un modèle de langage 24
1.4.1 Rappels sur la théorie de l'information 24
1.4.2 Définition de la perplexité 25
1.4.3 Perplexité et modèle de langage 25

Les modèles de langage probabilistes ont pour objet d'attribuer une probabilité à une séquence de mots.

De manière générale, la probabilité de la séquence de mots W_1^k s'exprime :

$$P(W_1^k) = P(w_1) \prod_{i=2}^{k} P(w_i|w_1, ..., w_{i-1}) = P(w_1) \prod_{i=2}^{k} P(w_i|h_i) \tag{1.1}$$

Remarques :

– ceci peut être vu comme la génération d'une séquence de mots par un modèle de Markov, pour lequel la séquence de mots $w_1, ..., w_{i-1}$ correspond à un état, et où w_i correspond à l'étiquette de la transition vers l'état $w_1, ..., w_i$
– h_i est l'*historique* du mot w_i
– $h_i = w_1, ..., w_{i-1}$

1.1 Approximations *n-grams*

En fait, ce type de probabilité ne peut être calculé de manière satisfaisante, car il nécessite un corpus d'apprentissage introuvable : ce corpus devrait être composé de toutes les phrases pouvant être énoncées dans le langage visé avec un nombre d'occurrences statistiquement significatif pour chaque phrase. Il est donc nécessaire de procéder à une approximation en remplaçant l'historique h par $S(h)$, une de ses classes d'équivalence. Dès lors, nous obtenons :

$$P(W_1^k) \simeq P(w_1) \prod_{i=2}^{k} P(w_i|S(w_1, ..., w_{i-1})) = P(w_1) \prod_{i=2}^{k} P(w_i|S(h_i)) \tag{1.2}$$

Cette approximation doit aboutir à l'obtention d'un modèle de langage suffisamment complet et précis pour être exploitable et utilisable d'un point de vue technique : par exemple, le corpus d'apprentissage nécessaire à la construction du modèle doit être disponible, ainsi que la technologie permettant d'intégrer ce modèle dans un système de reconnaissance de la parole.

Le choix de la classe d'équivalence de l'historique a bien entendu une influence directe sur les qualités d'un modèle de langage probabiliste.

L'approximation la plus fréquente considère comme équivalents deux historiques dont les $n-1$ derniers mots sont identiques : il s'agit de l'approximation *n-gram*.

1.1.1 Modèles *n-grams*

Le modèle de type *n-gram* constitue la base actuelle de la modélisation stochastique du langage : pour ce genre de modèle, l'historique d'un mot est représenté par les $n-1$ mots qui le précèdent. Ainsi, la formule permettant de calculer la probabilité $P(W_i^k)$ de la séquence de mots W_i^k devient :

$$P(W_1^k) = P(w_1) \prod_{i=2}^{n-1} P(w_i|w_1, ..., w_{i-1}) \prod_{i=n}^{k} P(w_i|w_{i-n+1}, ..., w_{i-1}) \tag{1.3}$$

Dans la pratique, la valeur de n dépasse rarement 3 :

– Si $n = 1$: modèle *unigram*. Ce type de modèle est particulier car il ne prend en compte aucun historique :

$$P(W_1^k) = \prod_{i=1}^{k} P(w_i) \tag{1.4}$$

– Si $n = 2$: modèle *bigram*. Ce type de modèle ne prend en compte que le mot précédent :

$$P(W_1^k) = P(w_1) \prod_{i=2}^{k} P(w_i|w_{i-1}) \tag{1.5}$$

– Si $n = 3$: modèle *trigram*. Ce type de modèle prend en compte les deux mots précédents :

$$P(W_1^k) = P(w_1)P(w_2|w_1) \prod_{i=3}^{k} P(w_i|w_{i-2}w_{i-1}) \tag{1.6}$$

Même si ce genre de modèle semble particulièrement réducteur en ne modélisant que des contraintes lexicales courtes, il contient suffisamment d'informations pour améliorer très fortement un système de reconnaissance de la parole basé uniquement sur la modélisation acoustique. Sa simplicité rend son emploi aisé et n'engendre pas un coût élevé en terme de calcul lors de la phase de reconnaissance. Enfin, une qualité fondamentale des modèles *n-grams* est la couverture totale des phrases pouvant être exprimées dans un langage. En contrepartie, sa précision est limitée puisque ce type de modèle accepte aussi des phrases n'appartenant pas au langage visé, mais la probabilité donnée à ces phrases par le modèle est en principe plus faible que la probabilité associée à une phrase correcte.

1.1.2 Modèles *n-classes*

La quantité de données nécessaire à l'apprentissage d'un modèle de langage robuste et performant, malgré l'approximation *n-gram*, reste importante. En partant du constat que certains mots ont un comportement similaire, leur regroupement en classes est envisageable. La modélisation du comportement comporte au moins deux avantages :

1. Le nombre d'événements à modéliser est moindre, il nécessite donc moins de données d'apprentissage.
2. L'utilisation des classes permet d'établir une généralisation : certains événements non vus au niveau des mots dans le corpus d'apprentissage peuvent être modélisés au niveau des classes.

Trois types de classes sont couramment utilisés : les classes syntaxiques qui regroupent les mots de même catégorie grammaticale[1], les classes morphologiques qui regroupent les mots selon leur racine morphologique (lemme), et les classes obtenues par classification automatique.

[1] Par exemple, la classe NMS peut regrouper tous les noms communs au masculin singulier.

Selon le type de modèle utilisé, une entité lexicale peut appartenir à plusieurs classes. Par exemple, dans le cas de classes syntaxiques, le mot “été” peut appartenir à la classe des verbes au participe passé, ainsi qu'à la classe des noms communs.

1.1.2.1 Modèles basés sur les classes syntaxiques

Ce type de modèle est particulièrement intéressant pour les langues à fort taux de flexion, comme le français, l'allemand ou l'italien. Ces langues demandent un grand nombre de données pour un apprentissage robuste. L'utilisation des classes syntaxiques permet de réduire le volume des données d'apprentissage, tout en intégrant de l'information sur les structures grammaticales.

Soient $g(w_t) = g_t$ la classe syntaxique du mot apparaissant à l'instant t et G l'ensemble des classes syntaxiques. Une approximation d'un modèle *trigram* basée sur des classes syntaxiques est donnée par la formule suivante :

$$P(w_t|g_{t-2}g_{t-1}) \simeq \sum_{g_t \in G} P(w_t|g_t)P(g_t|g_{t-2}g_{t-1}) \tag{1.7}$$

Remarque : les probabilités $P(g_t|g_{t-2}g_{t-1})$ sont estimées de la même manière que pour un modèle *n-gram* classique.

L'utilisation des modèles de langage de type *n-gram* à base de classes syntaxiques est décrite dans (Cerf-Danon et El-Bèze, 1991,).

1.1.2.2 Modèles basés sur les classes morphologiques et syntaxiques

Les modèles morpho-syntaxiques (El-Bèze et Derouault, 1990,) complètent les modèles à base de classes syntaxiques : ils ajoutent aux connaissances syntaxiques des connaissances sémantiques fournies par les lemmes des mots. Par exemple, les mots “aller”, “vais”, “irai”, “irais”, “allons”, ... sont regroupés dans la même classe morphologique, puisqu'ils dérivent du même lemme. D'un point de vue syntaxique, ces mots appartiennent à la classe des verbes[2], mais la classe morphologique dans laquelle ils sont regroupés introduit une information sémantique supplémentaire.

Soient $g(w_t) = g_t$ et $l(w_t) = l_t$ respectivement la classe syntaxique et la classe morphologique du mot apparaissant à l'instant t et G et L respectivement l'ensemble des classes syntaxiques et l'ensemble des classes morphologiques. Une approximation d'un modèle morpho-syntaxique *trigram* est donnée par la formule suivante :

$$P(w_t|h_t) \simeq \sum_{g_t \in G} P(g_t|g_{t-2}g_{t-1})(\lambda(g_t)P(w_t|g_t) + (1 - \lambda(g_t))P_m(w_t|g_t, h_t)) \tag{1.8}$$

[2] Selon le choix des classes syntaxiques utilisées, il est tout à fait possible d'avoir plusieurs classes pour représenter les verbes. Par exemple, il peut y avoir la classe des verbes au participe passé, la classe des verbes à l'infinitif, la classe des verbes conjugués, ... Dans ce paragraphe, nous parlons de la classe des verbes uniquement dans un souci de simplification et de clarté du discours.

où la composante morphologique P_m est formulée de la manière suivante :

$$P_m(w_t|g_t, h_t) = \sum_{l_t \in L} P(l_t|l_{t-2}l_{t-1})P(w_t|g_t, l_t) \tag{1.9}$$

Le paramètre d'interpolation λ dépend de la classe syntaxique du mot prédit : si il s'agit d'un mot outil (conjonction, article, ...) λ vaut 1, sinon, dans le cas de mots porteurs de sens, λ prend une valeur comprise entre 0 et 1.

L'utilisation de ce type de modèles est décrite dans (Cerf-Danon et El-Bèze, 1991,).

1.1.2.3 Modèles basés sur la classification automatique des mots

Il est également possible de créer des classes de mots sans utiliser de connaissances *a priori* de type syntaxique ou morphologique.

Plusieurs algorithmes ont été proposés qui permettent de construire des classes de mots en optimisant certains critères, en particulier dans (Jelinek, 1990) et (Brown *et al.*, 1992). Ceux-ci peuvent être des critères de maximum de vraisemblance, de *cross-validation*, ou encore d'information mutuelle entre deux classes de mots. D'autres algorithmes utilisent directement la notion de similarité entre les mots pour opérer des regroupements : pour (Farhat *et al.*, 1996), deux mots sont similaires si ils peuvent se substituer dans le corpus d'apprentissage, c'est-à-dire s'ils apparaissent dans le même contexte. La notion de contexte peut varier en fonction de l'approche choisie. Par exemple, pour (Ries *et al.*, 1995), le contexte d'un mot englobe plusieurs mots qui le précèdent et plusieurs mots qui le suivent.

Des connaissances *a priori* peuvent être utilisées : (Damnati, 1999) propose un algorithme de classification automatique intégrant des informations syntaxiques ou conceptuelles.

1.1.3 Modèles à séquences de longueurs variables

Une des caractéristiques communes des modèles présentés jusqu'ici est l'utilisation d'une taille d'historique fixe.

D'autres travaux ont proposé des modèles de type *n-gram* modélisant des contraintes sur des historiques de longueur variable. (Beaujard et Jardino, 1999,) proposent de regrouper des mots en une seule unité lexicale (*compound word*) : par exemple, les trois mots “il”, “y” et “a” sont regroupés en une unité lexicale “*il_y_a*”. Dans ce cas, la taille de l'historique est augmentée en n'intervenant que sur le lexique. D'autre part, (Deligne et Bimbot, 1995,) présente une variante des modèles *n-grams* : le modèle de langage *n-multigram*. Pour un modèle 3-*multigram*, la vraisemblance donnée à la séquence de mots $w_1w_2w_3w_4$ est formulée de la façon suivante :

$$P(w_1 w_2 w_3 w_4) = \sum \begin{cases} p([w_1])p([w_2 w_3 w_4]) \\ p([w_1 w_2 w_3])p([w_4]) \\ p([w_1 w_2])p([w_3 w_4]) \\ p([w_1 w_2])p([w_3])p([w_4]) \\ p([w_1])p([w_2 w_3])p([w_4]) \\ p([w_1])p([w_2])p([w_3 w_4]) \\ p([w_1])p([w_2])p([w_3])p([w_4]) \end{cases} \tag{1.10}$$

où $p([w_i ... w_j])$ est la probabilité d'apparition de la séquence de mots $w_i ... w_j$ dans le corpus d'apprentissage.

A la différence d'un modèle *n-gram*, un modèle *n-multigram* ne comporte aucune probabilité conditionnelle, mais fait intervenir des probabilités sur les différentes segmentations possibles d'une séquence de mots. Ces segmentations donnent au modèle *n-multigram* le pouvoir de capter des contraintes locales fortes que ne peut pas saisir un modèle *n-gram* en raison de sa taille d'historique fixe.

Les modèles *n-multiclasses* (Zitouni *et al.*, 1998), qui constituent une variante des modèles *n-multigram*s, ont été proposés dans (Zitouni *et al.*, 2001) sous une forme plus élaborée basée sur la hiérarchisation des séquences de classes syntaxiques de longueur variable. Ce type de modèle, appelé modèle MC_η^ν, où ν est le nombre maximum de niveaux hiérarchiques et η la longueur maximale d'une séquence de classes syntaxiques, apporte aux modèles de type *n-multigram* la prise en compte de dépendances entre les séquences de classes syntaxiques de longueur variable.

Un modèle *n-gram* basé sur les syntagmes (groupes nominaux, groupes verbaux, ...) plutôt que sur les mots a été proposé dans (Béchet *et al.*, 1999). Ce type de modèles, qui permet de modéliser des contraintes à longue distance est particulièrement efficace pour la gestion du nombre, surtout pour les mots homophones très courants en français. Par exemple pour des phrases du type "les problèmes d'environnement ne constituent pas une priorité actuelle", le mot "constituent" sera bien orthographié avec un modèle *trigram* basé sur les syntagmes, alors qu'avec un modèle *trigram* sur les mots, la marque du pluriel ne sera probablement pas retenue.

1.1.4 Discussion

La majorité des systèmes de reconnaissance de la parole actuels exploitent les modèles *n-grams* car ils peuvent facilement être utilisés avec les modèles acoustiques pour un gain en performance très important. De nombreux travaux visent à améliorer ces modèles. Les modèles *n-classes* peuvent pallier le problème du manque de données d'apprentissage, car ils ont moins de paramètres à estimer. Ces modèles basés sur des classes intègrent des connaissances qui sont très mal modélisées par les modèles *n-grams* (connaissances syntaxiques, "sémantiques", informations statistiques). Les modèles *n-grams* et les modèles *n-classes* peuvent être combinés par interpolation linéaire :

$$P(W_1^n) = \prod_{i=1}^{n} \lambda P_{mot}(w_i|h_i) + (1-\lambda) P_{classe}(w_i|h_i) \tag{1.11}$$

où λ est un réel compris entre 0 et 1, P_{mot} est la probabilité donnée par le modèle *n-gram* et P_{classe} la probabilité donnée par le modèle *n-gram* à base de classes.

(Maltese *et al.*, 2001) montrent que l'interpolation linéaire d'un modèle *n-classe* avec un modèle *n-gram* permet d'obtenir des gains significatifs en perplexité[3], et de réduire le nombre d'erreurs commises par un système de reconnaissance ; les expériences de cette étude ont été menées avec de gros corpora d'apprentissage, et dans trois langues différentes (anglais britannique, italien, et français). Ces résultats attestent qu'au-delà de leur pouvoir de généralisation, les modèles *n-grams* à base de classes ajoutent de l'information pertinente quand ils sont combinés à des modèles *n-grams*, même quand les données d'apprentissage semblent suffisantes pour estimer un modèle *n-gram* robuste.

Les regroupements de mots (*"il_y_a"*) autorisent la modélisation de contraintes à plus longue portée que peut offrir un modèle *n-gram* classique. Quant aux modèles *n-multigrams*, ils arrivent à capter des contraintes locales fortes alors que les modèles *n-grams* sont pris en défaut du fait de l'invariance de la taille de l'historique utilisé. Les modèles *n-multigrams* classiques ne permettent pas d'améliorer les performances d'un système de reconnaissance utilisé avec un modèle *n-gram*. Cependant, l'utilisation du système MAUD (Fohr *et al.*, 1997) avec le modèle MC_η^ν pour le rescoring d'une liste de *n*-best montre une amélioration du taux de reconnaissance par rapport aux résultats obtenus à l'aide d'un modèle *n-gram* classique (Zitouni *et al.*, 2001).

Le modèle de langage hybride que nous présentons dans le chapitre 4 de ce mémoire se situe entre ces différentes approches : notre objectif est de proposer un modèle qui associe les avantages de l'utilisation des classes (généralisation et intégration de connaissances) aux bénéfices de la modélisation sur de longues distances (contraintes sur des séquences de mots de longueurs variables) et à la simplicité des modèles de type *n-gram*. De plus, à l'instar des modèles *n-multigrams*, ce modèle de langage peut également décrire des contraintes locales fortes.

1.2 Estimation des paramètres

L'apprentissage d'un modèle de langage *n-gram* consiste à estimer un ensemble de probabilités à partir d'un corpus d'apprentissage.

Notons V le vocabulaire du langage. Supposons que V contient k mots $\{w_1, w_2, ..., w_k\}$. Construire un modèle de langage *n-gram* pour ce vocabulaire revient à estimer les N_n probabilités $P(w_i|w_{i-1}w_{i-2}...w_{i-n+1})$, avec :

$$N_n(k) = k^n \tag{1.12}$$

[3] Une définition de la perplexité est proposée en section 1.4.2

Par exemple, $N_2 = k^2$ pour un modèle *bigram*, et $N_3 = k^3$ pour un modèle *trigram*.

Mais cette formule néglige les probabilités utilisées pour les mots se trouvant en début de phrase et dont l'historique est plus court que $n-1$ mots. Calculer un modèle de langage de type *n-gram* pour ce vocabulaire revient en fait à estimer les N_n probabilités $P(w_i|w_{i-1}w_{i-2}...w_{i-n+1})$, plus les N_{n-1} probabilités $P(w_i|w_{i-1}w_{i-2}...w_{i-n})$, ..., plus les N_1 probabilités $P(w_i)$. Il y a donc N probabilités à estimer pour calculer un modèle de langage de type *n-gram* dont le vocabulaire est fixé à k mots, avec :

$$N(n,k) = \sum_{i=1}^{n} N_i = \sum_{i=1}^{n} k^i \tag{1.13}$$

1.2.1 Distribution paramétrique

Pour chaque historique $h_i = w_i...w_{i+n-1}$ (avec $w_i...w_{i+n-1} \in V$), la distribution des probabilités $P(w|h_i)$ est une distribution discrète. La probabilité $P(w|h_i)$ est notée $\theta_{i,w}$, paramètre de cette distribution. Dans un souci de simplification, nous supposerons par la suite que l'historique h est fixé : chaque mot w est associé à un vecteur de paramètres θ_w, avec $P(w;\theta) = \theta_w$.

L'espace des paramètres, noté Θ, est caractérisé par les contraintes suivantes :

$$\Theta = \left\{ \theta = [\theta_w]_{w \in V} : \forall w \in V, 0 \leq \theta_w \leq 1, \sum_{w \in V} \theta_w = 1 \right\} \tag{1.14}$$

L'estimation des paramètres s'effectue à partir d'un échantillon du langage visé. Cet échantillon, noté S, est de la forme :

$$S = w_1 w_2 ... w_m$$

Cet échantillon est en fait constitué de variables aléatoires indépendamment distribuées, avec m le nombre d'événements apparaissant dans l'échantillon S.

La vraisemblance $P(S;\theta)$ du corpus est définie ainsi :

$$P(S;\theta) = \frac{m!}{\prod_{w \in V} c(w)!} \prod_{w \in V} \theta_w^{c(w)} \tag{1.15}$$

avec

$$c(w) = \sum_{i=1}^{m} \delta(w_i = w)$$

et $\delta(e) = 1$ si e est vrai et $\delta(e) = 0$ sinon.

Il existe plusieurs méthodes pour procéder à l'estimation des paramètres du modèle de langage.

1.2.2 Estimation par Maximum de Vraisemblance

Ce critère d'estimation considère le paramètre θ comme une quantité inconnue à déterminer. La valeur choisie est celle qui maximise la vraisemblance de l'observation de l'échantillon S. Ainsi, la valeur donnée à θ par le critère du maximum de vraisemblance (MV) est :

$$\theta^{MV} = \underset{\theta \in \Theta}{argmax}\, P(S;\theta) = \underset{\theta \in \Theta}{argmax} \prod_{i=1}^{k} P(w_i;\theta) \tag{1.16}$$

Ainsi, en tenant compte des contraintes de Θ, en considérant le logarithme de la vraisemblance au lieu de la vraisemblance elle-même, et en utilisant le facteur lagrangien pour prendre en compte la contrainte de sommation à 1, on obtient (Federico et De Mori, 1998b,) :

$$\theta_w^{MV} = \frac{c(w)}{m} \tag{1.17}$$

La valeur de θ_w^{MV} correspond à la fréquence d'apparition de l'événement w dans l'échantillon S.

Pour des probabilités conditionnelles de type *trigram*, cette valeur devient :

$$\theta_{i,j,w}^{MV} = \frac{c(w_i w_j w)}{c(w_i w_j)}$$

et donc la probabilité d'apparition du mot w précédé des mots $w_i w_j$ donnée par un modèle *trigram* estimé par le critère de maximum de vraisemblance est

$$P^{MV}(w|w_i w_j) = \frac{c(w_i w_j w)}{c(w_i w_j)} \tag{1.18}$$

où $c(w_i w_j w)$ correspond au nombre d'occurrences de la suite de mots $w_i w_j w$ dans le corpus d'apprentissage S et $c(w_i w_j)$ au nombre d'occurrences de la suite de mots $w_i w_j$.

1.2.3 Estimation Bayésienne

Dans le cadre de l'estimation bayésienne (Duda et Hart, 1973,), le vecteur de paramètres θ est considéré comme une variable aléatoire d'une distribution *a priori* connue. L'objectif est alors de donner une estimation de θ à partir de l'échantillon S et de la distribution *a priori* $P(\theta)$ du paramètre θ.

L'application de la règle de Bayes permet d'obtenir la distribution *a posteriori* de θ :

$$P(\theta|S) = \frac{P(S|\theta)P(\theta)}{P(S)} \tag{1.19}$$

Remarque : la distinction de notation entre $P(S|\theta)$ et $P(S;\theta)$ porte uniquement sur la nature de θ. Lorsque θ est une variable aléatoire, la notation $P(S|\theta)$ est utilisée, alors que la notation $P(S;\theta)$ est préférée lorsque θ est une quantité inconnue.

L'estimateur bayésien de θ à partir de l'échantillon S et de la distribution *a posteriori* de θ, s'écrit :

$$\theta^B = E[\theta|S] = \int_\Theta \theta P(\theta|S)d\theta = \int_\Theta \theta \frac{P(S|\theta)P(\theta)}{P(S)}d\theta$$

$$\theta^B = \frac{\int_\Theta \theta P(S|\theta)P(\theta)d\theta}{\int_\Theta P(S|\theta)P(\theta)d\theta}$$

Dans le cas d'une distribution de probabilité discrète, l'estimateur bayésien de θ s'écrit (Vapnik, 1982) :

$$\theta_w^B = \frac{\int_\Theta \theta_w \prod_{z\in V} \theta_z^{c(z)} d\theta}{\int_\Theta \prod_{z\in V} \theta_z^{c(z)} d\theta} = \frac{c(w)+1}{m+k}, \forall w \in V \tag{1.20}$$

où k est le nombre d'événements existant dans V (par exemple, k peut être la taille du vocabulaire V).

Ainsi la probabilité d'apparition du mot w précédé des mots $w_i w_j$ donnée par un modèle *trigram* estimé par le critère bayésien s'écrit :

$$P^B(w|w_i w_j) = \frac{c(w_i w_j w)+1}{c(w_i w_j)+k^2} \tag{1.21}$$

où k est la taille du vocabulaire V.

1.2.4 Estimation par Maximum a Posteriori (MAP)

Le paramètre θ est ici aussi considéré comme une variable aléatoire d'une distribution *a priori* connue $P(\theta)$. Pour respecter le critère du *maximum a posteriori*, le paramètre θ est contraint de maximiser la probabilité *a posteriori* $P(\theta|S)$. Or, en utilisant la règle de Bayes (voir la formule (1.19)) et en constatant que le dénominateur de la fraction est indépendant du paramètre θ, la contrainte exercée sur θ se réduit à maximiser la valeur de $P(S|\theta)P(\theta)$. Ainsi :

$$\theta^{MAP} = \underset{\theta\in\Theta}{argmax}\, P(\theta|S) = \underset{\theta\in\Theta}{argmax}\, P(S|\theta)P(\theta) \tag{1.22}$$

Il est à noter que le critère du maximum *a posteriori* est équivalent au critère de maximum de vraisemblance dans le cas où la distribution *a priori* connue $P(\theta)$ est uniforme.

1.3 Lissage

C'est à partir de la valeur des fréquences d'apparition des *n-grams* dans les données d'apprentissage que sont estimés les paramètres d'un modèle de langage. Malheureusement, la quantité de données est en général insuffisante et certains *n-grams*,

voire même certains mots du lexique, n'apparaissent jamais dans le corpus d'apprentissage.

Les techniques de lissage tentent de compenser cette carence : elles peuvent être vues comme une sorte de généralisation qui permet d'attribuer une probabilité non nulle à un événement non vu dans le corpus d'apprentissage. Les principales techniques de lissage sont décrites dans (Chen, 1996) où est également présentée une discussion sur leurs performances respectives.

1.3.1 Principe

L'estimation des paramètres d'un modèle de langage de type *n-gram* est le plus souvent obtenue par la combinaison de deux composants : un modèle de *discounting* (décompte) et un modèle de redistribution. Le principe général est de prélever une quantité à la masse des probabilités issue des événements observés, et de la redistribuer aux probabilités associées aux événements non vus.

La probabilité d'un mot jamais vu en présence d'un historique donné est, sans lissage, nulle. Au contraire, les méthodes de lissage présentées ici lui attribuent une valeur non nulle calculée à partir d'un historique réduit.

Il existe principalement deux schémas d'utilisation du lissage : les techniques de repli et les techniques d'interpolation.

1.3.1.1 *Discounting*

La fréquence conditionnelle relative fr d'un mot w selon un historique h s'écrit :

$$\begin{cases} fr(w|h) = \frac{c(hw)}{c(h)} & \text{si } c(h) > 0 \\ fr(w|h) = 0 & \text{si } c(h) = 0 \end{cases} \tag{1.23}$$

Toutes les méthodes de *discounting* introduisent une fréquence conditionnelle décomptée $fr^*(w|h)$ telle que :

$$0 \leq fr^*(w|h) \leq fr(w|h) \quad \forall hw \in V^n \tag{1.24}$$

1.3.1.2 Redistribution

Pour un historique h donné, la redistribution de la masse de probabilités ôtée de fr s'effectue à l'aide d'une composante appelée la probabilité de fréquence nulle (*zero-frequency probability*), calculée à partir de fr^*.

La probabilité de fréquence nulle, notée λ, est définie comme suit :

$$\lambda(h) = 1 - \sum_{w \in V} fr^*(w|h) \tag{1.25}$$

Cette définition implique que pour un historique jamais observé ($c(h) = 0$), alors $\lambda(h) = 1$.

Pour un mot w jamais rencontré après l'historique h, la probabilité de fréquence nulle associée à h est utilisée pour pondérer la valeur de $P(w|h')$, où h' est un historique moins restrictif que h et pour lequel on suppose que l'événement $h'w$ a plus de chance d'avoir été observé que hw.

1.3.2 Lissage par repli (*backing-off*)

Le lissage par repli (Katz, 1987) est un lissage de type hiérarchique. Le principe de cette technique consiste à utiliser un modèle de langage plus général lorsque un modèle spécifique ne détient pas suffisamment d'information pour un contexte donné.

Par exemple, lorsque pour un *n-gram* hw, où h correspond aux $n-1$ mots précédant le mot w, aucune observation n'a été obtenue sur le corpus d'apprentissage, le modèle *n-gram* se tourne vers un modèle de niveau inférieur *(n-1)-gram* : ce processus peut bien sûr être réitéré jusqu'au niveau le plus bas, le *zéro-gram*, qui consiste en l'attribution d'une constante indépendante du mot w.

La probabilité d'un *n-gram* est donc estimée à partir du lissage de l'approximation la plus significative (du point de vue de la quantité d'observations) :

$$P(w|h) = \begin{cases} fr^*(w|h) & \text{si } fr^*(w|h) > 0 \\ \alpha_h \lambda(h) P(w|h') & \text{sinon} \end{cases} \tag{1.26}$$

avec

$$\alpha_h = \left(\sum_{w: fr^*(w|h)=0)} P(w|h') \right)^{-1}$$

qui permet à la distribution $P(w|h)$ de respecter la contrainte de sommation à 1.

1.3.3 Lissage par interpolation

Au contraire du lissage par repli, le lissage par interpolation est également effectué pour des *n-grams* qui ont été observés dans le corpus d'apprentissage. Cette méthode est une mixture de modèles qui garantit qu'au moins un des termes est non nul.

La probabilité d'un *n-gram*, lissée par interpolation, s'écrit :

$$P(w|h) = fr^*(w|h) + \lambda(h) P(w|h') \tag{1.27}$$

1.3.4 Techniques de *discounting*

Il existe plusieurs techniques de *discounting* et de redistribution pour le lissage de modèles de langage utilisés en reconnaissance de la parole. Une grande partie est présentée dans (Federico et De Mori, 1998a,) et (Chen et Rosenfeld, 2000,). Nous ne détaillerons ici que trois techniques de *discounting* : la technique dite de Good-Turing, la méthode d'*absolute discounting*, et la méthode de *linear discounting*.

1.3.4.1 Méthode de Good-Turing

La technique de *discounting* dite de Good-Turing a été introduite dans (Katz, 1987) associée à la technique de lissage par repli.

Pour cette méthode, la fréquence relative décomptée fr^* s'écrit :

$$fr^*(w|h) = \begin{cases} \frac{c^*(hw)}{c(h)} & \text{si } c(hw) > 0 \\ 0 & \text{sinon} \end{cases} \tag{1.28}$$

où $c^*(hw)$est appelée la fréquence corrigée :

$$c^*(hw) = (c(hw)+1)\frac{n_{c(hw)+1}}{n_{c(hw)}} \tag{1.29}$$

où n_x représente le nombre des différents *n-grams* apparaissant x fois dans le corpus d'apprentissage. $c^*(hw)$ découle de la formule de Good-Turing (Good, 1953) .

D'après les formules (1.25) et (1.28), la probabilité de fréquence nulle, avec la technique de Good-Turing, s'écrit :

$$\lambda(h) = 1 - \sum_{w:c(hw)>0} \frac{c^*(hw)}{c(h)} \tag{1.30}$$

En pratique, nous pouvons remarquer que n_x a des valeurs positives pour x assez petit (i.e. pour de petites fréquences), mais que n_x est souvent nul pour des valeurs de x plus élevées (i.e. pour de grandes fréquence). Pour pallier ce problème, plusieurs variantes ont été proposées pour lisser les valeurs de n_x afin que celles-ci soient toujours strictement positives, en particulier dans (Church et Gale, 1991,) et (Katz, 1987).

Le modèle présenté par (Katz, 1987) pour un modèle de langage *trigram*, définit la fréquence modifiée c' à partir de la fréquence corrigée c^* , de la fréquence c et d'un seuil l. Nous avons :

$$c'(hw) = \begin{cases} \frac{c^*(hw)-\alpha c(hw)}{1-\alpha} & \text{si } 0 \leq c(hw) \leq l \\ c(hw) & \text{si } c(hw) > l \end{cases} \tag{1.31}$$

avec :

$$\alpha = (l+1)\frac{n_{l+1}}{n_1}$$

Ainsi, seules les fréquences de mots dont la valeur est supérieure à l seront modifiées. Une des valeurs utilisées en pratique est $l = 5$.

La fréquence modifiée c' est alors utilisée à la place de c^* pour la définition (1.30) de la probabilité de fréquence nulle λ.

Dès lors, le modèle *trigram* est calculé de la manière suivante :

$$P(w_3|w_1w_2) = \begin{cases} \frac{c'(w_1w_2w_3)}{c(w_1w_2)} & \text{si } c(w_1w_2w_3) > 0 \\ \lambda(w_1w_2)\alpha_{w_1w_2}P(w_3|w_2) & \text{si } c(w_1w_2w_3) = 0 \quad \text{et } c(w_1w_2) > 0 \\ P(w_3|w_2) & \text{si } c(w_1w_2) = 0 \end{cases} \tag{1.32}$$

où

$$\alpha_{w_1w_2} = \left(\sum_{w_3 : c(w_1w_2w_3)=0} P(w_3|w_2) \right)^{-1}$$

Le modèle *bigram* peut être calculé selon le même principe.

1.3.4.2 *Absolute discounting*

La technique de l'*absolute discounting* (Witten et Bell, 1991,) consiste à retrancher une quantité constante aux fréquences d'apparition d'événements observés, et à la redistribuer aux valeurs des fréquences d'apparition des événements non vus.

La fréquence décomptée fr^* et la probabilité λ de fréquence nulle sont définies ainsi :

$$fr^*(w|h) = max\left(\frac{c(hw) - \beta}{c(h)}, 0\right) \tag{1.33}$$

et

$$\lambda(h) = \beta\frac{n(h)}{c(h)} \tag{1.34}$$

où β est une petite valeur constante positive, et $n(h)$ est le nombre de mots différents apparaissant après l'historique h dans le corpus d'apprentissage.

En utilisant la technique d'*absolute discounting* associée à une stratégie de repli, un modèle *trigram* se calcule de la manière suivante :

$$P(w_3|w_1w_2) = \begin{cases} \frac{c(w_1w_2w_3)-\beta}{c(w_1w_2)} & \text{si } c(w_1w_2w_3)-\beta > 0 \\ \lambda(w_1w_2)\alpha_{w_1w_2}P(w_3|w_2) & \text{si } c(w_1w_2w_3)-\beta \leq 0 \quad \text{et } c(w_1w_2) > 0 \\ P(w_3|w_2) & \text{si } c(w_1w_2) = 0 \end{cases} \tag{1.35}$$

où

$$\alpha_{w_1w_2} = \left(\sum_{w_3 : c(w_1w_2w_3)=0} P(w_3|w_2) \right)^{-1}$$

1.3.4.3 *Linear discounting*

Pour cette méthode de *discounting*, la quantité enlevée à une fréquence conditionnelle est proportionnelle à la valeur de cette fréquence :

$$fr^*(w|h) = (1-\lambda(h))fr(w|h) \tag{1.36}$$

avec, bien sûr : $0 \leq \lambda(h) \leq 1$.

Plusieurs solutions pour estimer la probabilité de fréquence nulle λ ont été proposées, en particulier dans (Witten et Bell, 1991,) et (Ney et Essen, 1991,). Par exemple, (Witten et Bell, 1991,) propose de rendre $\lambda(h)$ proportionnelle au nombre d'événements observés en présence de l'historique h :

$$\lambda(h) = \frac{n(h)}{c(h)+n(h)} \tag{1.37}$$

et :

$$fr^*(w|h) = \frac{c(hw)}{c(h)+n(h)} \tag{1.38}$$

L'utilisation de la technique d'*absolute discounting* quelle que soit la méthode utilisée pour estimer λ, associée à une stratégie de repli, permet de calculer un modèle *trigram* de la manière suivante :

$$P(w_3|w_1w_2) = \begin{cases} (1-\lambda(h))fr(w|h) & \text{si } c(w_1w_2w_3) > 0 \\ \lambda(h)\alpha_{w_1w_2}P(w_3|w_2) & \text{si } c(w_1w_2w_3) = 0 \quad \text{et } c(w_1w_2) > 0 \\ P(w_3|w_2) & \text{si } c(w_1w_2) = 0 \end{cases} \tag{1.39}$$

avec toujours :

$$\alpha_{w_1w_2} = \left(\sum_{w_3 : c(w_1w_2w_3)=0} P(w_3|w_2) \right)^{-1}$$

1.4 Évaluation d'un modèle de langage

La qualité d'un modèle de langage intégré dans un système de reconnaissance de la parole se mesure à son influence sur les performances du système. Par exemple, pour une application de dictée vocale, un modèle de langage est jugé meilleur qu'un autre si son utilisation permet un plus petit nombre d'erreurs sur les mots. Par contre, dans le cas d'une application de dialogue homme-machine, la qualité d'un modèle de langage se mesure à son influence positive sur la reconnaissance des mots clés : puisque l'application ne se base que sur ces mots pour effectuer l'interprétation de la phrase reconnue, les erreurs sur les mots non porteurs de sens peuvent être ici négligées.

Un modèle de langage peut aussi être considéré comme une entité autonome, indépendante de tout système de reconnaissance. Dans ce cas, il est aussi possible d'évaluer sa qualité.

1.4.1 Rappels sur la théorie de l'information

Considérons une source s produisant des émissions σ_t indépendantes les unes des autres, pouvant prendre M valeurs possibles, avec pour chacune d'entre elles une probabilité d'émission P. On définit la quantité d'information I liée à l'émission de σ_t comme :

$$I(\sigma_t) = -log_2 P(\sigma_t) \tag{1.40}$$

La quantité d'information est une valeur positive ou nulle. Elle caractérise la diminution de l'incertitude que l'émission de σ_t retire au destinataire : la réalisation d'une émission peu probable est plus informative que la réalisation d'une émission probable. Ainsi, à un événement certain correspond une quantité d'information nulle.

La valeur moyenne de la quantité d'information émise par une source s se nomme entropie, par analogie à la notion de désordre utilisée en thermo-dynamique, et s'écrit (Cover et Thomas, 1991,) :

$$H(s) = -\sum_{t=1}^{M} P(\sigma_t) log_2 P(\sigma_t) \tag{1.41}$$

Considérons maintenant $P(\sigma_1^n)$ comme étant la probabilité d'émission de la séquence de mots $\sigma_1^n = \sigma_1 ... \sigma_n$ par la source s, et faisons tendre n vers l'infini.

L'entropie H d'une source s permettant d'émettre la séquence σ_1^n, s'écrit alors :

$$H(s) = \lim_{n \to \infty} -\frac{1}{n} \sum_{t=1}^{n} P(\sigma_1^n) log P(\sigma_1^n) \tag{1.42}$$

(Jelinek, 1997) présente une étude détaillée sur l'entropie.

En pratique, il n'est pas possible d'observer toutes les réalisations de séquences σ_1^n pouvant être produites par une source donnée. Il est donc nécessaire d'introduire la notion de source ergodique. Une source est dite ergodique si toute séquence d'émissions suffisamment longue de cette source est représentative de cette source, et permet d'en étudier la structure statistique.

1.4.2 Définition de la perplexité

Considérons maintenant l'émission σ_t comme le mot w_t, et la source d'émission s comme le langage L.

Définissons la quantité appelée *logprob* et notée LP :

$$LP = -\frac{1}{n} log_2 P(W_1^n) \tag{1.43}$$

où $W_1^n = w_1 ... w_n$ est une séquence de mots suffisamment longue, et $P(W_1^n)$ la probabilité d'apparition de W_1^n donnée par un modèle de langage Γ.

La définition (1.43) de LP est une approximation de l'entropie définie par (1.42).

La perplexité du modèle de langage Γ sur la séquence de mots W_1^n du langage L s'exprime alors :

$$PP = 2^{LP} \tag{1.44}$$

Ainsi, pour un modèle de langage de type *n-gram*, nous avons :

$$PP = 2^{-\frac{1}{n}\Sigma_{t=1}^{n} log_2 P(w_t|h)} \tag{1.45}$$

où $P(w_t|h)$ est une probabilité donnée par le modèle *n-gram*.

1.4.3 Perplexité et modèle de langage

La perplexité d'un modèle d'un langage est un indicateur de sa capacité de prédiction. La perplexité est assimilable à un facteur de branchement moyen : on interprète une perplexité de valeur K comme le fait que, pour un historique donné, le modèle doit choisir entre K mots équiprobables pour déterminer le prochain mot émis. Plus la valeur de la perplexité est petite, plus le pouvoir de prédiction du modèle de langage est grand.

On distingue généralement la perplexité d'un modèle calculée sur un corpus de test, et la perplexité calculée sur le corpus d'apprentissage. La perplexité calculée sur le corpus d'apprentissage permet de mesurer la qualité des approximations utilisées pour définir le modèle (choix de la classe d'équivalence de l'historique par exemple), et permet de juger de la pertinence du critère choisi pour l'estimation des paramètres. La perplexité calculée sur un corpus de test permet de mesurer le

degré de généralisation du modèle, et peut, par exemple, être utilisée pour comparer diverses techniques de lissage.

Enfin, même si la perplexité est un bon estimateur de la qualité d'un modèle de langage, l'expérience montre que sa corrélation avec les performances d'un système de reconnaissance de la parole est loin d'être parfaite : il arrive souvent qu'un modèle ayant une perplexité bien meilleure qu'un autre ne permette pas, une fois intégré dans un système de reconnaissance, d'améliorer ses performances.

En fait, la perplexité permet d'évaluer le pouvoir de discrimination d'un modèle entre l'ensemble des mots de son lexique, alors que dans son utilisation, l'objectif du modèle de langage est d'établir une discrimination entre un sous-ensemble de ce lexique. Par exemple, dans un système de reconnaissance de la parole, le modèle intervient uniquement dans le choix d'un mot parmi un ensemble de mots phonétiquement proches.

Chapitre 2

Approche formelle et autres modèles de langage

Sommaire

2.1 Grammaires formelles 28
2.1.1 Classification de Chomsky 29
2.1.2 Grammaires et automates 30
2.1.3 Automates à états finis 30
2.1.4 Automates de type pushdown 31
2.2 Grammaires formelles vs. modèles de langage stochastiques . . . 31
2.2.1 Couverture 32
2.2.2 Construction 32
2.2.3 Longueur des contraintes 32
2.3 Méthodes mixtes 33
2.3.1 Grammaires probabilistes 33
2.3.2 Contraintes syntaxiques intégrées dans un système d'étiquetage probabiliste 34
2.3.3 Modèle de langage basé sur des automates stochastiques à *n-grams* variables 35

Il est courant, dans le cadre du traitement automatique du langage naturel, d'opposer deux approches. Du point de vue historique, la première est basée sur des règles formelles, essentiellement des grammaires construites par des experts en linguistique. La seconde (la plus répandue actuellement en reconnaissance de la parole) est l'approche statistique basée sur les modèles *n-grams*, qui est la seule à avoir été évoquée jusqu'ici dans ce mémoire. Le fossé qui sépare ces deux écoles tend à se réduire. Plusieurs travaux présentent de nouvelles voies intermédiaires dans le but de combiner le meilleur des deux approches. C'est dans cette mouvance que s'inscrit notre proposition de modèle de langage hybride, s'appuyant sur l'intégration de connaissances linguistiques *a priori* (grammaires locales) dans un modèle d'essence statistique.

Avant la présentation de ce modèle, une vue d'ensemble sur l'approche basée sur des règles s'impose, ainsi que le survol de quelques techniques intermédiaires qui ont un lien étroit avec notre proposition.

2.1 Grammaires formelles

Dans le cadre défini par la théorie chomskienne dans (Chomsky, 1957) et (Chomsky, 1965), une grammaire G est définie par $G = (V, T, P, S)$, tel que :

- V est un ensemble fini de symboles terminaux,
- T est un ensemble fini de symboles non-terminaux,
- P est un ensemble fini de règles de production (appelées aussi règles de réécriture), de type $\alpha \rightarrow \beta$, où α et β sont des chaînes de symboles appartenant à V ou T, et où α est non vide et contient au moins un élément de T,
- S est un symbole non terminal particulier : il est le symbole de départ.

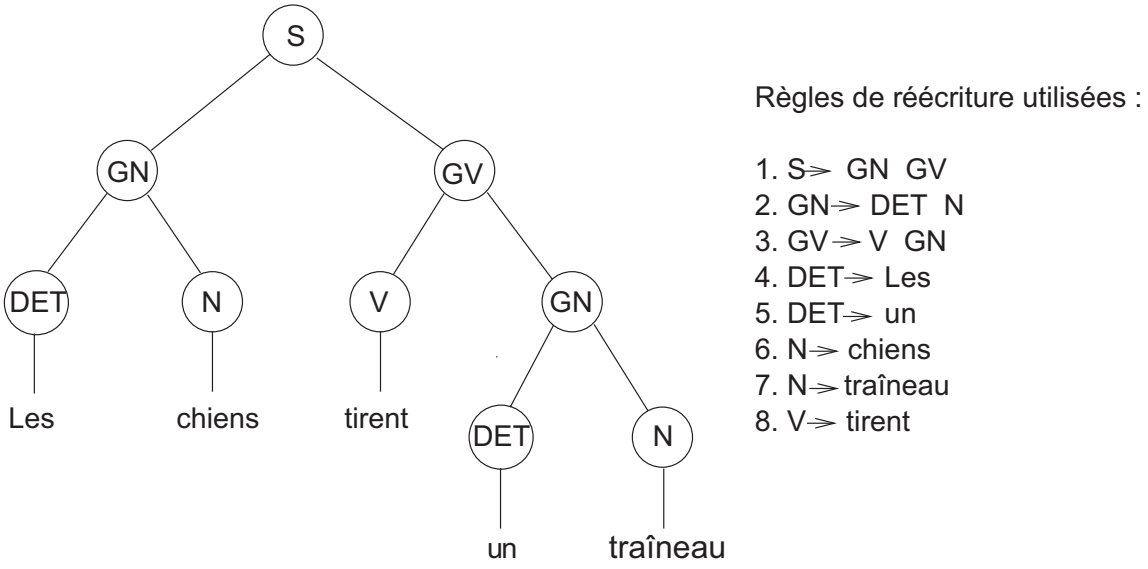


FIG. 2.1 – *Représentation de la structure syntaxique d'une phrase sous forme d'arbre*

Le langage engendré par une grammaire G, noté $L(G)$ est l'ensemble des suites de symboles terminaux que permet de produire la grammaire G. La grammaire est généralement utilisée comme outil de description d'un langage plutôt que comme outil de production.

La structure syntaxique d'une phrase est couramment représentée sous la forme d'un arbre de dérivations. La figure 2.1 en donne un exemple pour la phrase "Les chiens tirent un traîneau". Les règles de réécriture indiquées dans cett figure sont des règles qui appartiennent à la grammaire qui permet de produire la phrase

énoncée dans le langage visé. L'utilisation d'une règle de réécriture se nomme une dérivation. La notation suivante peut s'utiliser pour exprimer les dérivations représentées par l'arbre de la figure :

$$S = \alpha_0 \xrightarrow{r_1} \alpha_1 \xrightarrow{r_2} \alpha_2 \xrightarrow{r_4} \alpha_3 \xrightarrow{r_6} \alpha_4 \xrightarrow{r_3} \alpha_5 \xrightarrow{r_8} \alpha_6 \xrightarrow{r_2} \alpha_7 \xrightarrow{r_5} \alpha_8 \xrightarrow{r_7} \alpha_9$$

ou, plus simplement :

$$S \xrightarrow{r_1 r_2 r_4 r_6 r_3 r_8 r_2 r_5 r_7} \alpha$$

2.1.1 Classification de Chomsky

Chomsky a établi une hiérarchie entre quatre grands types de grammaires, selon les contraintes imposées sur les règles de réécriture.

1. Type 0 : une grammaire G est de ce type si les règles de réécriture sont de la forme $\alpha \to \beta$, avec α non vide et contenant au moins un élément de T. Aucune autre contrainte n'existe sur les règles de réécriture.
2. Type 1 : ce genre de grammaires, appelées grammaires contextuelles, est un sous-ensemble des grammaires de type 0, dont les règles de réécriture sont soumises à la contrainte $|\alpha| \leq |\beta|$, où $|.|$ représente la longueur de la chaîne de symboles.
3. Type 2 : grammaires dites hors-contexte (CFG : *context free grammar*s, en anglais), sous-ensemble des grammaires de type 1. Les règles de réécriture sont de la forme $A \to \beta$, où A est un non-terminal ($A \in T$). Il existe, entre autres, un cas particulier de grammaire de type 2 : les grammaires sous la forme normale de Chomsky, pour lesquelles les règles de réécriture s'écrivent $A \to w$ ou $A \to BC$, où w est un symbole terminal et B et C des symboles non-terminaux. (Aho et Ullman, 1972,) a montré que toutes les grammaires hors-contexte peuvent être ramenées à des grammaires sous forme normale de Chomsky.
4. Type 3 : ces grammaires sont appelées grammaires régulières et sont un sous-ensemble des grammaires de type 2. On distingue deux sortes de grammaires régulières : les grammaires "régulières-gauche" dont les règles de réécriture sont de la forme $A \to w$ et $A \to wB$, et les grammaires "régulières-droite", dont les règles de réécriture sont de la forme $A \to w$ et $A \to Bw$.

Les grammaires hors contexte sont les plus utilisées pour le traitement du langage naturel, bien qu'il ait été prouvé que celui-ci n'est pas engendré par une grammaire de ce type (Pullum et Gazdar, 1982,). Cette utilisation répandue vient du bon compromis existant entre la capacité de description des grammaires hors-contexte, et les restrictions qu'elles induisent au niveau de l'analyse grammaticale : ces restrictions permettent une analyse efficace, et la puissance de description des

grammaires hors-contexte permet de décrire une grande partie de la structure d'un langage.

Pour des applications restreintes concernant le traitement du langage naturel, les grammaires régulières sont préférées aux grammaires hors-contexte : puisque la partie visée du langage est déterminée, la capacité de description des grammaires régulières s'avère suffisante. De plus, leur analyse grammaticale est plus efficace, en terme de rapidité, que celle des grammaires hors-contexte.

2.1.2 Grammaires et automates

Un automate peut être défini comme un appareil permettant d'accomplir certaines tâches sans intervention humaine. Chacune des grammaires présentées dans la classification de Chomsky peut être associée à un type d'automates qui acceptent le langage engendré par cette grammaire. Ces automates permettent de signifier l'appartenance ou non d'une phrase à un langage.

Le tableau 2.1 établit la correspondance entre les quatre grands types de grammaires présentées précédemment et quatre sortes d'automates.

Grammaires	Automates
Grammaires non restreintes	Machine de Turing
Grammaires dépendantes du contexte	Automates linéaires bornés
Grammaires hors-contexte	Automates de type “push-down“
Grammaires régulières	Automates à états finis

TAB. 2.1 – *Types d'automates acceptant les langages engendrés par les différentes grammaires*

Les deux types de grammaires très majoritairement utilisées pour le traitement automatique du langage naturel sont les grammaires hors-contexte et les grammaires régulières. Il est donc intéressant de se focaliser sur les deux types d'automates associés à ces grammaires.

2.1.3 Automates à états finis

Ce type d'automates est associé à des grammaires régulières. Un automate à états finis est constitué de noeuds et d'arcs étiquetés, comme le montre la figure 2.2. Un ou plusieurs noeuds sont considérés comme étant des états initiaux S_i. Il existe également au moins un noeud considéré comme un état final. En partant d'un noeud initial, un arc est traversé si le mot courant de la phrase analysée correspond à l'étiquette de l'arc (dans l'automate de la figure 2.2, ce sont les classes syntaxiques des mots de la phrase analysée qui sont comparées aux étiquettes des transitions de l'automate). Si le mot courant correspond à une étiquette d'un arc provenant du noeud courant, le mot suivant de la phrase devient le mot analysé courant, et le processus recommence à partir du noeud de l'automate auquel aboutit l'arc

traversé. Dans ce type d'automates, la récursivité est autorisée : il est possible, lors du processus d'analyse d'une phrase, de passer plusieurs fois par le même noeud.

Une phrase est acceptée par l'automate (et appartient donc au langage engendré par la grammaire associée à cet automate) si l'analyse de cette phrase a produit un chemin dans cet automate commençant sur un état initial et se terminant sur un état final.

Il est à noter qu'une même phrase peut mener à la production de plusieurs chemins différents, tout comme l'analyse grammaticale de cette phrase peut mener à l'utilisation de plusieurs combinaisons différentes de règles de réécriture. Cela est dû à l'impossibilité des grammaires régulières de lever certaines ambiguïtés. Malheureusement, la présence d'une ambiguïté syntaxique signifie généralement la présence d'une ambiguïté sémantique. De plus, l'absence d'ambiguïté syntaxique ne suffit pas à éviter une ambiguïté sémantique.

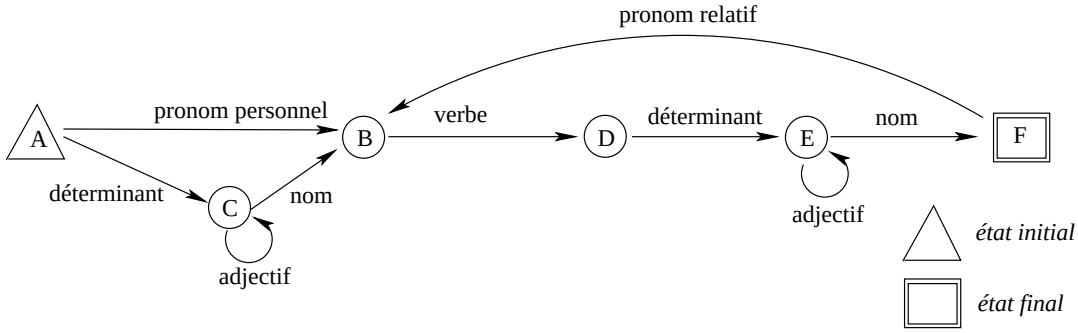


FIG. 2.2 – *Exemple d'automate à états finis (automates réguliers)*

2.1.4 Automates de type pushdown

Ce type d'automates ressemblent aux automates à états finis. Cependant, ils intègrent un mécanisme de stockage de l'information (une mémoire de type "dernier dedans-premier dehors"). Cette particularité leur permet de pouvoir être associés à des grammaires hors-contexte, ce qui est impossible aux automates à états finis.

2.2 Grammaires formelles vs. modèles de langage stochastiques

Les deux grandes approches du traitement automatique du langage naturel que sont l'approche par grammaires formelles et l'approche statistique ont des qualités, et des limites, différentes.

2.2.1 Couverture

Les grammaires, aussi complètes soient elles, ne permettent pas de décrire un langage naturel dans son intégralité. Cet aspect du problème est encore plus flagrant pour le traitement du langage parlé puisque de nombreuses tournures de phrases grammaticalement incorrectes peuvent être employées lors d'une conversation orale. Les modèles stochastiques n'ont pas ce problème de couverture : ils acceptent toutes les phrases d'un langage. Même les phrases incorrectes sont acceptées. Les modèles stochastiques sont plus permissifs que les grammaires formelles, ce qui est intéressant pour traiter la parole spontanée malgré l'acceptation d'hypothèses de reconnaissance erronées.

Pour pallier le problème de couverture des grammaires, plusieurs solutions ont été proposées :

– L'utilisation d'analyseurs correcteurs d'erreurs (Aho et Peterson, 1972,). Cette technique consiste à étudier les erreurs de grammaires les plus fréquentes et à intégrer un mécanisme de correction de ces erreurs.
– L'analyse d'îlots (Corazza *et al.*, 1991). Cette méthode propose d'utiliser des grammaires stochastiques uniquement sur les parties d'hypothèses de reconnaissance qui présentent de fortes vraisemblances acoustiques. En ne travaillant que sur des sous-séquences de mots, l'utilisation des analyseurs d'îlots permet de s'affranchir du problème de couverture des grammaires. Cette technique est toutefois difficile à mettre en place : l'intégration d'automates dans un modèle *n-gram* que nous proposons dans cette thèse se rapproche quelque peu de cette technique, tout en étant beaucoup plus simple d'utilisation.

2.2.2 Construction

Au niveau de la construction, les deux approches sont également très différentes : l'approche formelle se base sur une expertise linguistique, c'est-à-dire sur des travaux généralement effectués par des linguistes, travaux longs et en perpétuelle évolution. L'approche statistique, elle, est en principe totalement automatisée. Cependant, il faut noter que la quantité de corpus nécessaire à l'apprentissage d'un modèle de langage stochastique robuste n'est pas toujours disponible.

2.2.3 Longueur des contraintes

Les modèles de type *n-grams* apportent, en reconnaissance de la parole, des résultats vraiment surprenants compte tenu des contraintes à courte distance qu'ils utilisent : un modèle de ce type ne peut pas intégrer les contraintes sur de longues distances, alors qu'une grammaire formelle a cette capacité.

Les faiblesses et qualités des deux approches semblent complémentaires : cette constatation est le point de départ de travaux combinant des approches formelles et statistiques afin d'améliorer encore les performances des systèmes de traitement automatique du langage naturel.

2.3 Méthodes mixtes

Il existe plusieurs études combinant approche formelle et approche statistique. Quelques approches mixtes sont abordées ici.

2.3.1 Grammaires probabilistes

Les grammaires probabilistes sont un raffinement des grammaires formelles, généralement des grammaires hors-contexte (Salomaa, 1969). Chaque règle de production d'une grammaire probabiliste est associée à une probabilité. Cette information supplémentaire a pour but de réduire les ambiguïtés syntaxiques qui peuvent apparaître lors de l'analyse grammaticale d'une phrase. L'intérêt de cette information statistique augmente avec le nombre de règles de production qui constituent la grammaire : plus il y a de règles, plus le nombre d'ambiguïtés syntaxiques est susceptible d'augmenter et plus la capacité de discrimination des analyses grammaticales concurrentes est nécessaire.

La probabilité d'une dérivation (c'est-à-dire d'une suite d'applications de règles r_i de production) peut être écrite de la manière suivante :

$$P(S \xrightarrow{r_1 r_2 ... r_m} x) = P(r_1)P(r_2|r_1)P(r_3|r_1 r_2)...P(r_m|r_1 r_2 ... r_{m-1}) \tag{2.1}$$

où x est une chaîne de symboles terminaux de la grammaire G et $P(r_i|r_1 r_2 ... r_{i-1})$, avec $1 < i \leq m$ est la probabilité conditionnelle que la règle r_i est appliquée si les règles $r_1 r_2 ... r_{i-1}$ ont été appliquées précédemment. L'approximation suivante est cependant très répandue (Gonzales et Thomason, 1978,) :

$$P(r_i|r_1 r_2 ... r_{i-1}) \simeq P(r_i) \tag{2.2}$$

En appliquant cette hypothèse, l'équation (2.1) devient :

$$P(S \xrightarrow{r_1 r_2 ... r_m} x) = P(r_1)P(r_2)...P(r_m) \tag{2.3}$$

Il s'agit ici de l'hypothèse d'approximation la plus simple. D'autres hypothèses peuvent être proposées, comme celles présentées dans (Charniak et Carrol, 1994,), où la probabilité d'application d'une règle de réécriture dépend du non-terminal du membre gauche de la règle. De même, des approximations *bigrams* ou *trigrams* sur les historiques d'application des règles peuvent être proposées.

Les grammaires probabilistes sont une extension des grammaires formelles. Leur construction s'effectue en deux phases. Tout d'abord, il faut retenir un ensemble de règles de production, comme pour une grammaire formelle. A partir d'un corpus contenant des phrases déjà analysées, l'approche la plus simple pour calculer les probabilités d'apparition des règles de réécriture est de compter le nombre de fois où chaque règle est utilisée. La probabilité d'application d'une règle r_i de la grammaire G de type $A \rightarrow \alpha$ peut être notée $P(A \rightarrow \alpha|G)$, ou $P(r_i|G)$.

Ainsi, si il existe m règles de production dans la grammaire G contenant (uniquement) le symbole non-terminal A telles que $A \rightarrow \alpha_1, A \rightarrow \alpha_2, ..., A \rightarrow \alpha_m$, alors la probabilité de ces règles s'expriment :

$$P(A \rightarrow \alpha_j | G) = \frac{c(A \rightarrow \alpha_j)}{\sum_{i=1}^{m} c(A \rightarrow \alpha_i)} \tag{2.4}$$

où $c(.)$ correspond au nombre d'utilisations d'une règle.

Des hypothèses d'indépendance peuvent être définies sur l'usage des règles : par exemple, il est possible de considérer que la probabilité de la règle de production d'un groupe nominal ne prend pas en compte le fait que ce groupe nominal est dérivé d'un sujet ou d'un groupe verbal. Considérons la séquence de mots $W = w_1 w_2 ... w_k$ générée par une grammaire probabiliste hors-contexte, dont les règles sont sous la forme normale de Chomsky : $A_i \rightarrow A_m A_n$ et $A_i \rightarrow w_u$ où A_m et A_n sont deux symboles non terminaux qui permettent de ré-écrire A_i dans certains conditions. La probabilité de chacune des règles de G doit alors satisfaire la contrainte suivante :

$$\sum_{m,n} P(A_i \rightarrow A_m A_n | G) + \sum_u P(A_i \rightarrow w_u | G) \leq 1, \quad \text{pour tout } i \tag{2.5}$$

Si l'égalité est toujours atteinte, la grammaire est dite *propre*.

2.3.2 Contraintes syntaxiques intégrées dans un système d'étiquetage probabiliste

Dans le cas des grammaires probabilistes, l'idée était de partir de l'approche formelle et d'y intégrer des informations statistiques. La plupart des travaux cherchant à combiner les approches formelles et probabilistes abordent le problème de la manière inverse : c'est le cas, par exemple pour le système ECSta mis au point au Laboratoire d'Informatique d'Avignon, qui consiste en un système d'étiquetage probabiliste utilisant des contraintes syntaxiques à base de règles. Cette approche est décrite dans (El-Bèze et Spriet, 1995,) et (Spriet et El-Bèze, 1998,).

Après étude des performances d'un système d'étiquetage morpho-syntaxique[1] purement probabiliste, (El-Bèze et Spriet, 1995,) montre qu'il est possible de réduire sensiblement le nombre d'erreurs d'étiquetage en utilisant un ensemble très réduit de règles simples. Une particularité de cette méthode provient de la nature des règles syntaxiques, puisqu'il s'agit de règles d'agrammaticalité : ces règles sont utilisées afin de supprimer (ou plutôt de pénaliser très fortement) les hypothèses de l'espace de recherche qui contiennent des structures syntaxiques déclarées interdites.

[1] L'étiquetage des mots d'une phrase consiste généralement à associer chacun de ces mots à une classe particulière. Dans le cas d'un étiquetage syntaxique, il s'agit d'associer chaque mot à sa classe syntaxique (nom commun, déterminant, etc.). Comme certains mots peuvent appartenir à plusieurs classes, un système d'étiquetage a pour objectif de lever l'ambiguïté en associant chacun de ces mots à la classe syntaxique attendue en fonction du contexte.

Par exemple, lors de l'analyse des erreurs produites à la suite d'un étiquetage purement stochastique, il a été montré que l'une des erreurs les plus fréquentes provenait du mot *été* étiqueté à tort comme *nom masculin singulier* après l'auxiliaire *avoir* au lieu de *participe passé*. L'utilisation d'une règle d'agrammaticalité pénalisant l'hypothèse aboutissant à cette erreur élimine cette faiblesse du système probabiliste.

L'intégration des contraintes syntaxiques s'effectue, dans le système ECSta, par l'intermédiaire d'un l'algorithme de type $A*$ (Hart *et al.*, 1968). Cet algorithme permet de développer de façon partielle les hypothèses les plus compétitives à un instant donné, ce qui évite de parcourir entièrement l'espace de recherche. Lors du développement d'une hypothèse compétitive comportant une structure grammaticale énoncée par une des règles d'agrammaticalité, cette structure est détectée et l'hypothèse est fortement pénalisée au niveau de son score de vraisemblance, qui n'est alors plus compétitif afin d'éviter que le système ne choisisse cette hypothèse.

Ainsi, ECSta utilise des règles d'agrammaticalité basées sur des structures syntaxiques et créées à partir de l'analyse des erreurs les plus fréquentes d'un système d'étiquetage probabiliste, afin d'atténuer sensiblement les faiblesses de ce système.

2.3.3 Modèle de langage basé sur des automates stochastiques à *n-grams* variables

(Riccardi *et al.*, 1996) propose un modèle de langage probabiliste construit à partir d'automates stochastiques particuliers : les automates stochastiques à *n-grams* variables (VNSA : *Variable N-gram Stochastic Automata* en anglais). Bien que ce modèle n'intègre pas d'information autre que statistique, son étude est intéressante dans la mesure où il utilise exclusivement des automates, généralement réservés aux grammaires.

2.3.3.1 Définition d'un automate stochastique à *n-grams* variables

Un automate stochastique à *n-grams* variables Q (Riccardi *et al.*, 1995) est un automate non déterministe défini par le quintuplet $\{S, V, F, s_0, S_f\}$, où :

- S est un ensemble d'états s,
- V est un ensemble de mots w, contenant également le mot vide ϵ
- F est une fonction qui, pour un état $s \in S$ et un mot $w \in V$ donnés, renvoie l'ensemble des couples (t_i^w, p_i^w) : $F(s, w) = \{(t_i^w, p_i)\}(i \geq 1)$, où p_i^w sont les probabilités de transition entre l'état s et l'état t_i^w.
- s_0 est l'état initial
- S_f est l'ensemble des états finaux, et $S_f \subset S$.

Le processus d'analyse par l'automate d'une séquence de mots $W = w_1 w_2 ... w_k$ commence à l'état initial s_0 (instant 0) avec le mot w_1 et par l'utilisation de F avec s_0 et w_1. Ensuite, pour un état s à l'instant i, avec comme mot courant le mot w_i de W, l'automate procède en deux étapes :

1. il applique la fonction de transition F à l'état s et au mot w_i, avec $w_i \neq \epsilon$, se déplace sur chaque état renvoyé par F (si $F(s, w_i) \neq \epsilon$) et lit le prochain mot de W, qui devient le mot courant,
2. il applique la fonction de transition F à l'état s et au mot vide ϵ, et se déplace sur chaque état renvoyé (le mot courant restant inchangé).

Par définition, un état qui n'est atteint que par une transition associée au mot vide est appelé état vide. On remarque qu'un VNSA est effectivement un automate non déterministe puisque, à partir d'un état s et d'un mot w, la fonction de transition F peut proposer plusieurs états suivants.

Chaque état s d'un VNSA est associé à un m-tuple $v_1, v_2, ..., v_m$ où $0 \leq m < n$ et $v_i \in V$: n est appelé l'ordre de l'automate ; $v_1, v_2, ..., v_m$ est l'historique de l'état s et m la taille de cet historique. Dans le cas où $m = 0$, le mot vide ϵ est associé à l'état s.

L'historique $v_1, v_2, ..., v_m$ d'un état n'est utilisé que lors de la création de l'automate et de son apprentissage stochastique. Lors de l'utilisation du VNSA pour l'analyse d'une séquence de mots, cet historique est inutile, puisqu'il est implicitement connu grâce à la position de l'état courant dans l'architecture de l'automate.

2.3.3.2 Fonctionnement d'un automate stochastique à *n-grams* variables

Supposons être à un instant quelconque i pendant l'analyse d'une séquence de mots $W = w_1...w_k$. Au début de l'analyse, le mot w_1 a été présenté à l'entrée de l'automate. A l'instant i, le mot courant en entrée de l'automate est le mot w_j, et l'état courant est l'état s associé à l'historique $v_1, ..., v_m$. La fonction de transition F ne peut proposer alors que deux types de transitions :

1. Une transition de type 1 consomme le mot w_j, et mène soit à l'état non vide t_1^w associé à l'historique $v_1, v_2, ..., v_m, w_j$ (la taille de l'historique est augmentée), soit à l'état non vide t_2^w associé à l'historique $v_2, ..., v_m, w_j$ (la taille de l'historique reste inchangée). Dans les deux cas, le mot courant devient w_{i+1}. La taille maximale de l'historique associé à l'état du VNSA définit l'ordre du VNSA : la taille d'un historique utilisé par un automate d'ordre n ne peut excéder n.
2. Une transition de type 2 ne consomme aucun mot en entrée de l'automate et mène à un état vide t^ϵ associé à l'historique $v_2, ..., v_m$ (la taille de l'historique est diminuée). La probabilité p^ϵ associée à cet automate permet de gérer le mécanisme de repli (*backoff*) généralement utilisé dans un modèle de langage probabiliste. Une transition de type 2 est toujours disponible : ceci permet de pallier la non-existence d'une transition de type 1 due à un manque d'information lors de l'apprentissage du VNSA.

2.3.3.3 Exemple

La figure 2.3 représente une partie d'un VNSA du troisième ordre. Seuls les mots associés aux transitions sont affichés : les probabilités ne le sont pas par souci de

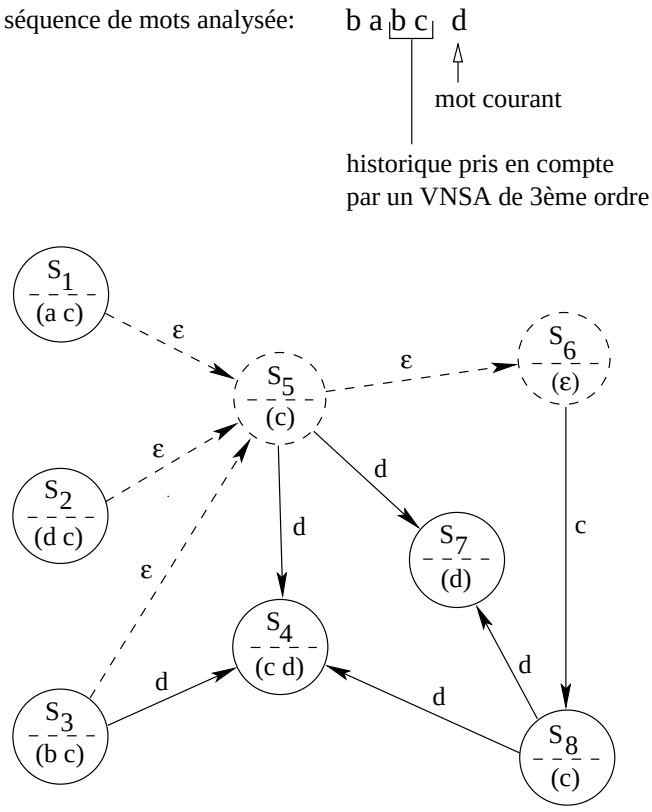


FIG. 2.3 – *Représentation d'une partie d'un automate stochastique à n-grams variables du troisième ordre*

clarté. L'historique associé à un état est écrit entre parenthèses (puisque l'historique n'est pas utile lors de l'utilisation du VNSA). Le vocabulaire de cet automate est $V = \{a, b, c, d, \epsilon\}$. Les états vides sont représentés par des cercles dessinés en pointillés. De la même manière, les transitions associées au mot vide ϵ sont dessinés par des flèches en pointillés.

Dans cet exemple, la séquence de mots "*b a b c d*" a été présentée au VNSA, et l'analyse courante se situe sur le dernier mot : la sous-séquence de mots "*b a b c*" a donc déjà été traitée. Puisqu'il s'agit d'un VNSA du troisième ordre, l'historique utilisé par ce VNSA ne peut pas s'étendre au-delà de "*b c*".

Comme le mot "*c*" vient d'être traité, l'historique ne peut être que "*b c*" ou "*c*" : l'automate doit être dans l'état S_3 ou l'état S_8 (pour se trouver dans l'état S_5, il faut d'abord transiter par l'état S_3).

A partir de l'état S_3, l'automate peut transiter vers l'état S_4 ou l'état S_5. L'état S_5 est un état vide vers lequel les états S_1, S_2 et S_3 peuvent transiter afin de diminuer la taille de l'historique (pour chaque état, la probabilité de transition est distincte !) et permettre ainsi de gérer le repli. Une fois dans l'état S_5, l'automate peut soit éliminer encore un mot de l'historique en transitant vers l'état vide S_6, soit conserver la même taille d'historique en transitant vers l'état non vide S_7, soit enrichir

l'historique en transitant vers l'état S_4.

A partir de l'état S_8, l'automate peut soit conserver la même taille d'historique en transitant vers l'état non vide S_7, soit augmenter la taille de l'historique en transitant vers S_4.

2.3.3.4 Non-déterminisme et probabilité donnée par un VNSA à une séquence de mots

Comme nous avons pu le voir, un VNSA est un automate non-déterministe : il existe donc plusieurs séquences d'états possibles au sein d'un VNSA associées à une même séquence de mots proposée en entrée. Chaque séquence d'états est associée à une probabilité (obtenue en multipliant les probabilités portées par les transitions menant à la séquence d'états). Or, l'objectif de la modélisation reste l'attribution d'une probabilité, unique, à une séquence de mots.

La probabilité P d'une séquence de mots W donnée par un VNSA Q se définit ainsi (Riccardi *et al.*, 1996) :

$$P(W,Q) = \sum_{\xi \in \Xi_W} P(W,\xi) \tag{2.6}$$

où Ξ_W est l'ensemble de toutes les séquences d'états ξ associées à la séquence de mots W dans l'automate Q.

Généralement, l'utilisation de la somme des probabilités des différentes séquences d'états est approchée par l'utilisation de la probabilité la plus grande. En d'autres terme, l'approximation suivante est acceptée, et très largement utilisée (Riccardi *et al.*, 1996) :

$$P(W,Q) \approx \underset{\xi \in \Xi_W}{max}\ P(W,\xi) \tag{2.7}$$

Chapitre 3

Modèles de langage et moteurs de reconnaissance de la parole

Sommaire

3.1 Reconnaissance de la parole : notions de base 39
3.1.1 Combinaison des modèles acoustiques et des modèles de langage 40
3.1.2 Espace de recherche et graphe de mots 41
3.2 Exemples de modèles de langage utilisés en seconde passe 42
3.2.1 Modèles *n-grams* 43
3.2.2 Modèles de langage à base de classes syntaxiques 44
3.2.3 Modèles de langage structurés 44
3.3 Adaptation d'un modèle de langage 45
3.3.1 Motivations 45
3.3.2 Acquisition des données d'adaptation 46
3.3.3 Techniques d'adaptation 47
3.3.4 Discussion 51
3.4 Évaluation d'un système de reconnaissance de la parole 52
3.4.1 Taux d'erreurs sur les mots 52
3.4.2 Intervalle de confiance 53
3.5 Présentation générale d'un système de dialogue 53
3.5.1 Architecture modulaire 53
3.5.2 Spécificités de la reconnaissance de la parole dans une application de dialogue 55

3.1 Reconnaissance de la parole : notions de base

Le but d'un système de reconnaissance de la parole est de pouvoir transcrire la phrase prononcée par un locuteur.

Pour cela, à partir de la séquence d'observations acoustiques $X = x_1 x_2 ... x_m$, un système de reconnaissance de la parole recherche la séquence de mots $\hat{W} = w_1 w_2 ... w_k$ qui maximise la probabilité *a posteriori* $P(W|X)$. Ceci s'écrit :

$$\hat{W} = \underset{W}{argmax}\, P(W|X) = \underset{W}{argmax}\, \frac{P(W)P(X|W)}{P(X)} \tag{3.1}$$

Comme la séquence d'observations acoustiques X est fixée, $P(X)$ est une valeur constante qui est inutile dans (3.1). On a donc :

$$\hat{W} = \underset{W}{argmax}\, P(W)P(X|W) \tag{3.2}$$

Deux types modèles sont donc utilisés pour la recherche de la phrase la plus probable : un modèle acoustique, qui fournit la valeur de $P(X|W)$, et un modèle de langage, qui fournit la valeur de $P(W)$.

L'unité utilisée pour la modélisation acoustique n'est pas nécessairement le mot. Généralement, l'unité choisie se trouve à un niveau inférieur : phonème, demi-syllabe, syllabe, ou une combinaison de ces unités de bas niveau.

Il existe également différents types de modèles de langage : les premiers chapitres de ce mémoire leur ont été consacrés. La modélisation acoustique ne sera pas abordée ici.

3.1.1 Combinaison des modèles acoustiques et des modèles de langage

fudge factor Bien que la formule (3.2) suggère que la probabilité du modèle acoustique et la probabilité du modèle de langage peuvent être combinées à travers une simple multiplication, il est nécessaire en pratique d'effectuer une pondération. Sans cela, la participation d'un des modèles est négligeable à cause de la différence d'ordre de grandeur des variations des deux distributions : lorsque la différence d'ordre de grandeur est trop importante, un des modèles prend l'ascendant sur l'autre en terme de puissance de discrimination. Ainsi, seul un des modèles est alors effectivement utilisé pour la prise de décision finale. Ceci s'ajoute à la différence de nature des deux probabilités : le modèle de langage fournit de probabilité de valeur discrète, alors que le modèle acoustique manipule des densités de probabilités.

La solution la plus couramment utilisée pour atténuer ce problème consiste à ajouter un poids, noté lw (pour *linguistic weight*) et souvent appelé *fudge factor*, au modèle de langage. On a alors :

$$\hat{W} = \underset{W}{argmax}\, P(W)^{lw} P(X|W) \tag{3.3}$$

Le poids lw est déterminé empiriquement à partir d'expériences effectuées sur un corpus de développement : la valeur choisie est celle qui optimise les performances du système de reconnaissance. Généralement, $lw > 1$.

Pénalité linguistique La contribution du modèle de langage peut aussi être interprétée comme une pénalité sur le nombre de mots. En fonction des valeurs des probabilités du modèle de langage, le système peut privilégier une séquence composée de peu de mots longs ou, au contraire, une séquence constituée de nombreux mots courts. Afin d'ajuster au mieux la tendance du système à insérer ou supprimer des mots, une valeur appelée pénalité linguistique et notée lp est insérée dans la formule (3.3), qui devient :

$$\hat{W} = \underset{W}{argmax}\, P(W)^{lw} lp^{N(W)} P(X|W) \tag{3.4}$$

où $N(W)$ est le nombre de mots de la séquence W.

Tout comme le fudge factor lw, la pénalité linguistique lp est déterminée empiriquement : la valeur choisie doit optimiser les performances du système de reconnaissance pour des expériences effectuées sur un corpus de développement.

Utilisation des logarithmes Les multiplications successives de probabilités, c'est-à-dire de valeurs comprises entre 0 et 1, mènent à manipuler des valeurs de plus en plus proches de 0. La limite de capacité de représentation de valeurs proches de 0 d'un ordinateur est rapidement atteinte, à moins de mettre en place des mécanismes coûteux en terme de temps de calcul. En pratique, les systèmes de reconnaissance de la parole ne manipulent pas directement les probabilités : ce sont les logarithmes de ces probabilités qui sont utilisés. Le passage aux logarithmes entraîne l'utilisation d'additions plutôt que de multiplications : ce type d'opérations conforte la propriété intéressante des logarithmes qui changent très lentement d'ordre de grandeur. Ainsi, la formule (3.4) se ré-écrit :

$$\hat{W} = \underset{W}{argmax}\, lw.\log P(W) + \log P(X|W) + lp.N(W) \tag{3.5}$$

3.1.2 Espace de recherche et graphe de mots

A partir de l'observation d'événements acoustiques et de connaissances *a priori* (lexique, modèles acoustiques, ...) , un système de reconnaissance génère un ensemble d'hypothèses de séquences de mots. Cet ensemble est appelé espace de recherche : le système doit en extraire la phrase qui satisfait l'équation (3.5). L'espace de recherche est généralement représenté sous la forme d'un graphe, appelé graphe de recherche, qui intègre les informations utilisées pour la génération des hypothèses : informations temporelles, unités acoustiques (phonèmes, syllabes, demi-syllabes, ...) associées à leurs scores acoustiques (probabilités données par le modèle acoustique), mots induits par les séquences d'unités acoustiques, ...

La recherche de la phrase de probabilité maximale au sein d'un graphe de recherche est analogue au problème de la recherche du chemin de poids minimal dans un graphe. De nombreux algorithmes existent pour résoudre ce problème (Cettolo *et al.*, 1998). Cependant, pour la majorité des systèmes, la taille de l'espace

de recherche est très importante et ralentit considérablement le traitement. Pour obtenir une solution dans un délai acceptable, une recherche en faisceau, appelée *beam search*, permet de restreindre l'espace de recherche en supprimant des hypothèses qui semblent localement peu probables (Ney *et al.*, 1992). Cet élagage ne garantit pas l'obtention de la phrase la plus probable, mais le compromis entre la durée du traitement et la perte de précision est très souvent largement acceptable.

L'utilisation de modèles de langage sophistiqués, par exemple un modèle *n-gram* avec un *n* assez grand, ralentit la recherche de la phrase de probabilité maximale. La solution la plus répandue consiste à utiliser ce type de modèle lors d'une deuxième passe : le graphe de recherche généré lors d'une première passe est élagué grâce à l'application d'un algorithme de *beam search,* et n'est plus composé que de mots. Chaque mot est alors associé à un score acoustique calculé à partir des scores des unités acoustiques qui le composent. Le graphe obtenu pour la deuxième passe est un graphe de mots : il est l'objet de traitements linguistiques lourds qui auraient fortement ralenti le processus de reconnaissance s'ils avaient été appliqués sur l'intégralité de l'espace de recherche dès la première passe.

La figure 3.1 montre l'exemple d'un graphe de mots simple : chaque transition est associée à un couple (mot, score acoustique). Les hypothèses de mots finissant au même instant sont représentées par des transitions terminant sur le même état du graphe : chaque état est lié à un instant t. Les symboles $< s >$ et $< /s >$ sont respectivement les symboles de début et fin de phrase : ils peuvent être associés à un état du signal de parole que le système de reconnaissance interprète comme du silence[1].

3.2 Exemples de modèles de langage utilisés en seconde passe

Généralement, le modèle de langage stochastique utilisé en première passe d'un processus de reconnaissance de la parole est un modèle *bigram*, voire *trigram*. Ces modèles ont la particularité d'être simples d'emploi et relativement peu coûteux en temps de calcul. Ces caractéristiques, combinées à l'influence largement bénéfique de ces modèles sur les résultats d'un processus de reconnaissance, sont à l'origine de leur très forte implantation dans les systèmes de reconnaissance de la parole.

Les modèles de langage plus évolués, faisant appel à des historiques plus importants ou à des sources de connaissances supplémentaires, sont utilisés en seconde passe sur un espace de recherche réduit à un graphe de mots ou à une liste des *n* meilleures phrases. Le graphe de mots ou la liste des *n* meilleures hypothèses sont issus du décodage effectué en première passe. Cette seconde passe, qui consiste à utiliser un ou plusieurs modèles nécessitant plus de ressources qu'un modèle *bigram* afin d'améliorer encore la reconnaissance, est habituellement connue sous le nom de phase de *rescoring*.

[1] Tout comme pour les mots, il peut y avoir plusieurs hypothèses de silence de début ou fin de phrase en concurrence.

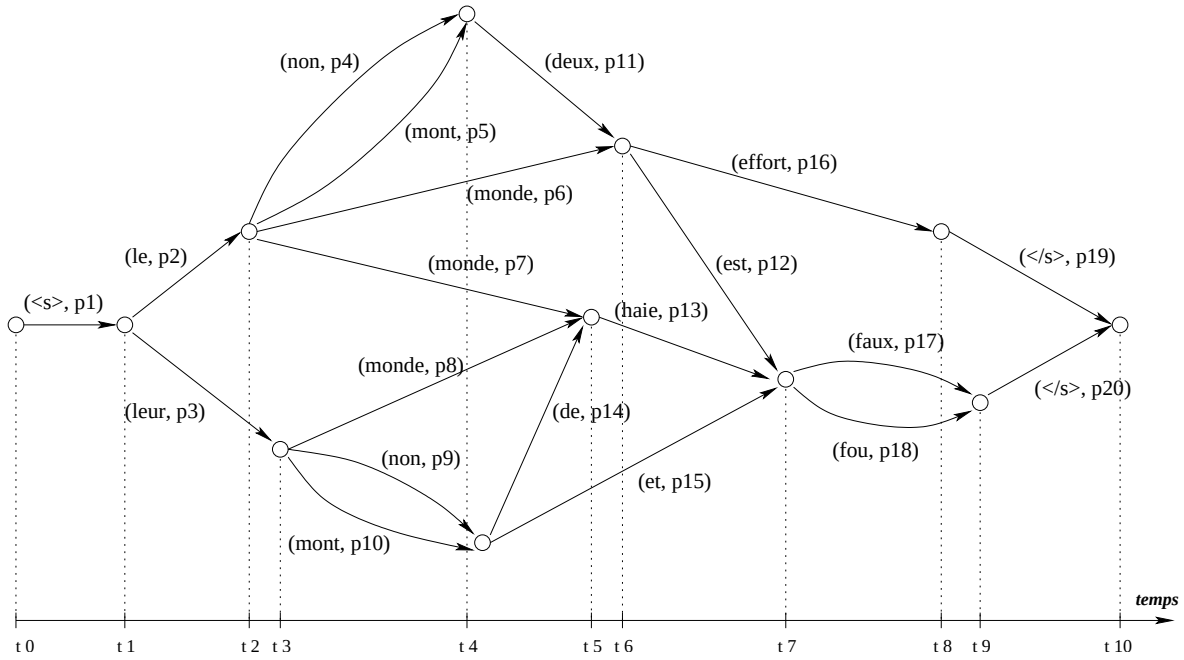


FIG. 3.1 – *Exemple simple de graphe de mots*

Bien entendu, rien n'empêche d'utiliser ces modèles de langages gourmands en ressources dans un système de reconnaissance basé sur une seule passe. Malheureusement, les algorithmes utilisés et la technologie actuelle ne permettent pas d'obtenir des résultats satisfaisants dans des délais raisonnables.

Dans une application conviviale de dialogue entre un homme et une machine, le système de reconnaissance de la parole doit avoir un temps de réponse proche du temps réel. L'utilisation de systèmes multi-passes permet d'utiliser des modèles de langage nécessitant de grosses ressources sans trop ralentir le processus global de reconnaissance.

Quelques exemples de modèles de langage utilisés en phase de rescoring sont présentés ici.

3.2.1 Modèles *n-grams*

Les modèles de langage *n-grams*, avec $n > 3$ sont souvent utilisés en seconde passe. La gestion d'un historique de taille n devient très lourde, en temps de calcul et en ressource mémoire, dès que n dépasse 3. Par exemple, (Woodland *et al.*, 1997) décrit un système de reconnaissance de la parole présenté lors de la campagne d'évaluation "1996 Broadcast News Hub4 Evaluation". Ce système est un système multi-passes qui utilise un modèle *quadrigram* dans sa dernière passe. (Gauvain *et al.*, 2000) présente deux systèmes : un système à une passe, et un système multi-passes. Ces deux systèmes peuvent utiliser un modèle de langage *quadrigram*.

Pour le système à une seule passe, l'utilisation d'un modèle de langage *quadrigram* donne de meilleurs résultats qu'un modèle *trigram* lorsque le temps de calcul nécessaire au processus de reconnaissance dépasse 20 fois le temps réel (le temps de calcul varie en fonction de la puissance de l'élagage permettant de réduire l'espace de recherche).

(Gauvain *et al.*, 2000) montre l'intérêt d'utiliser un modèle *quadrigram* en dernière passe d'un système multi-passe : sur les données de l'évaluation "1998 Broadcast News Hub4 Evaluation", le taux d'erreurs sur les mots atteint $14,2\%$ pour un temps de calcul égal à $8,4$ fois le temps réel en utilisant un modèle *quadrigram* dans un système multi-passes, alors que ce taux d'erreurs monte à $16,1\%$ en utilisant ce même modèle dans un système à simple passe pour un temps de calcul égal à $10,5$ le temps réel.

3.2.2 Modèles de langage à base de classes syntaxiques

Les modèles de langage à base de classes syntaxiques (voir section 1.1.2.2) ou morpho-syntaxiques[2] peuvent également être utilisés en seconde passe. En effet, comme un mot peut appartenir à plusieurs classes, une transition d'un graphe de mots est alors associée au triplet (mot, classe, score acoustique) au lieu du couple (mot, score acoustique). Ainsi, le nombre de transitions est augmenté. Si cette augmentation du nombre de transitions est réalisée en amont du système de reconnaissance, le temps de traitement sera alors accru par rapport à une utilisation en phase de *rescoring*.

Cependant, le débat reste ouvert entre les partisans de systèmes intégrant le maximum d'information dès le début du processus de reconnaissance, et les partisans de l'injection de l'information à différents niveaux. Pour des systèmes de dialogue qui doivent traiter le signal de parole en temps réel, les systèmes multi-passes avec utilisation de connaissances différentes à plusieurs niveaux sont pour l'instant privilégiés, au vu de leur meilleure vitesse de traitement.

3.2.3 Modèles de langage structurés

Les modèles de langage structurés ont été introduits dans (Chelba et Jelinek, 2000,). Ce type de modèle prend en compte des informations syntaxiques pour la détermination de classes d'équivalence des historiques des *n-grams*. Le modèle attribue une probabilité $P(\sigma, \pi)$ à chaque séquence de mots $\sigma = w_1 ... w_N$, où π est une des analyses grammaticales qui peuvent être déduites de σ. π est construite de manière hiérarchique, à partir de symboles terminaux et de symboles non terminaux : les symboles terminaux sont les mots de σ associés à leur classe syntaxique, alors que les symboles non terminaux sont notés par le "*headword*" (que l'on peut traduire par le "mot principal") du syntagme associé à l'analyse grammaticale partielle .

[2] ou d'autres types de classes qui admettent qu'un même mot puisse appartenir à plusieurs d'entre elles

Les travaux effectués dans ce domaine, en particulier dans (Chelba et Jelinek, 2000,) et (Yougner, 1967) , mènent au modèle suivant :

$$P(w_q|\bar{h}_q) = \frac{1}{Z(\bar{h}_q)} \sum_{\{\pi_q\}} P(w_q|\bar{h}_q, \pi_q) P(\bar{h}_q, \pi_q) \tag{3.6}$$

où $\bar{h}_q$ représente l'historique (historique débutant sur le premier mot w_1 de la phrase et se terminant sur w_q), $\{\pi_q\}$ est l'ensemble des analyses grammaticales partielles possibles de w_1 jusqu'à w_q, et $Z(\bar{h}_q)$ permet de normaliser la valeur pour assurer la sommation à 1.

L'analyse grammaticale d'une séquence de mots permet d'extraire des sous-séquences de mots liés par des contraintes syntaxiques : les syntagmes. De chaque syntagme est extrait le mot le plus important (du point de vue syntaxique, et sémantique) : le "*headword*". En pratique, le modèle peut être simplifié en remplaçant π_q par l'ensemble des *headwords* mis en évidence par l'analyse grammaticale partielle ayant abouti à π_q : cet ensemble, noté p_q, peut aussi être réduit aux $(n-1)$ *headwords* précédents, ce qui réduit le modèle de langage structuré à une variante de modèle *n-gram* standard. Le modèle qui en résulte est de la forme :

$$P(w_q|\bar{h}_q, \pi_q) \simeq P(w_q|h_q, p_q)$$

où h_q correspond aux $(k-1)$ mots précédents. Ainsi, le modèle prend en compte un historique p_q construit à partir de connaissances syntaxiques sur une portée de taille supérieure à celle obtenu par un modèle *n-gram* classique, et un historique h_q standard portant sur des événements proches.

Ce type de modèles, qui intègre des contraintes sur de longues distances dont la gestion s'avère relativement complexe, nécessite une utilisation en phase de *rescoring*.

3.3 Adaptation d'un modèle de langage

3.3.1 Motivations

Un modèle de langage spécifique à un domaine est plus efficace qu'un modèle généraliste. Par exemple, un système de dictée vocale appliquée au domaine de la radiologie médicale donnera de meilleures performances avec un modèle de langage adéquat.

De même, il est intéressant de modifier un modèle de langage en fonction des variations du contexte d'une application : cette approche nécessite de détecter ces variations et d'adapter dynamiquement le modèle. Historiquement, la première approche traitant de l'adaptation d'un modèle de langage a été présentée (Kuhn et De Mori, 1990,) lors de l'introduction des modèles à mémoire cache : les probabilités d'un modèle de langage à mémoire cache sont modifiées dynamiquement en fonction des événements observés dans un passé récent.

L'élaboration de modèles de langage spécifiques est coûteux pour le développement de nouvelles applications. La principale cause de leur coût vient du besoin en ressources linguistiques nécessaires à leur apprentissage. L'adaptation d'un modèle de langage déjà existant est une solution. Le modèle de langage obtenu par adaptation est plus proche d'une application spécifique que le modèle de langage initial, tout en étant plus robuste qu'un modèle de langage appris sur le peu de données d'apprentissage disponibles.

La figure 3.2 décrit le schéma général d'un processus d'adaptation de modèle statistique. Deux corpora de transcriptions de phrases sont utilisés :

– le corpus d'adaptation (noté corpus A), de petite taille, spécifique à l'application visée
– un corpus généraliste (noté corpus G), de taille satisfaisante, suffisamment proche de la tâche visée

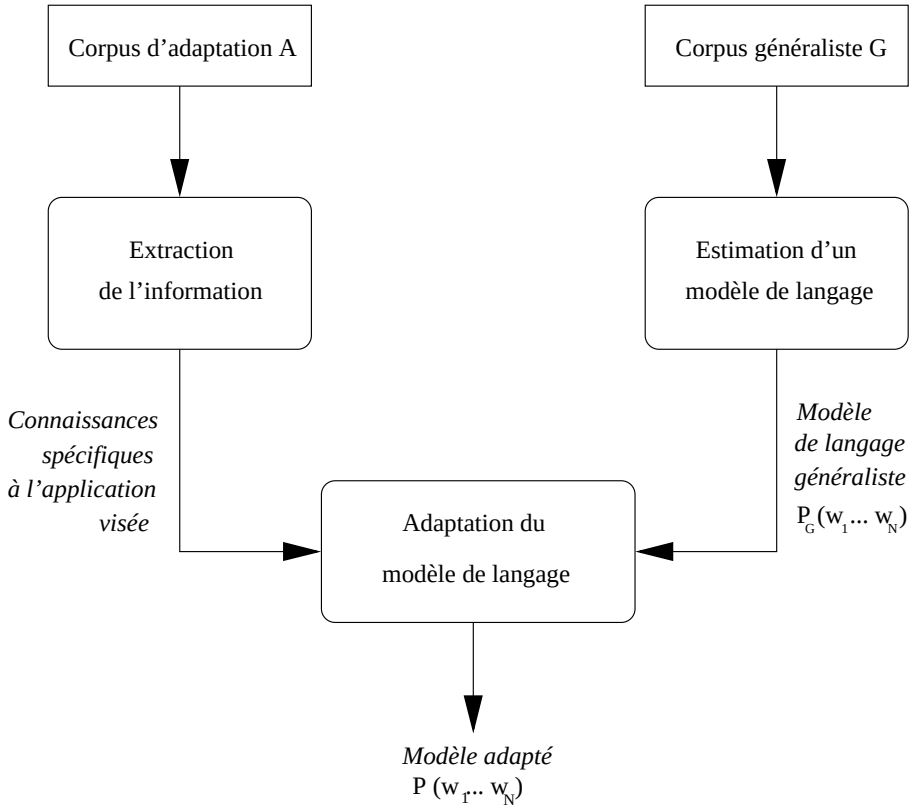


FIG. 3.2 – *Schéma général du processus d'adaptation d'un modèle de langage*

3.3.2 Acquisition des données d'adaptation

Le meilleur moyen d'obtenir un corpus de données spécifiques à une nouvelle application est d'organiser une campagne de collecte en situation réelle suffisamment importante. Malheureusement, cela nécessite un système de reconnaissance

proche de la nouvelle application, en plus de moyens économiques très importants. D'autres techniques ont donc été mise en place.

3.3.2.1 Magicien d'Oz

Le corpus utilisé pour développer une nouvelle application de reconnaissance (corpus d'adaptation) est rarement disponible : dans certaines circonstances, il est nécessaire de le créer en menant une campagne d'acquisition qui consiste à transcrire manuellement des phrases prononcées par des utilisateurs réels, ou de le créer artificiellement. Ainsi, dans le cas d'applications de dialogue homme-machine, la technique du magicien d'Oz est souvent utilisée. Cette méthode permet d'obtenir un matériau d'adaptation pertinent (Riccardi et Gorin, 2000,).

3.3.2.2 Grammaires génératrices

Lorsque les phrases susceptibles d'être prononcées sont entièrement couvertes par une grammaire, celle-ci peut être utilisée pour générer des phrases qui pourraient être énoncées par l'utilisateur. Une faible quantité de données d'apprentissage permet de pondérer les règles de grammaire et de générer un corpus d'adaptation réaliste(Kellner, 1998).

3.3.2.3 Listes de *n*-best

Parfois, aucun corpus d'adaptation n'est disponible. Dans ce cas, il est possible de créer ce corpus à partir du processus de reconnaissance de la parole. La liste des n hypothèses de phrases les plus probables d'après le système de reconnaissance est utilisée ; cette liste de n phrases est communément appelée liste des $n-best$. Dans ce cas, chaque hypothèse participe à la construction du corpus d'adaptation en fonction de sa vraisemblance *a posteriori*. Les séquences de mots bien reconnues apparaissent généralement dans plusieurs hypothèses de la liste des $n-best$, et ont donc plus d'incidence que les séquences de mots erronées (Souvignier *et al.*, 2000).

3.3.2.4 Recherche documentaire

Enfin, il est également possible d'utiliser des techniques de recherche documentaire, par exemple sur le *Web*, pour augmenter encore le corpus d'adaptation. Ce type de techniques est présenté dans (Zhu et Rosenfeld, 2001,) et (Vaufreydaz *et al.*, 2001).

3.3.3 Techniques d'adaptation

Pour une étude détaillée des différentes techniques d'adaptation d'un modèle de langage, il est intéressant de se reporter à (Bellegarda, 2001) ou (De Mori et Federico, 1999,).

Brièvement, voici quelques techniques d'adaptation :

Interpolation de modèles Comme nous l'avons observé précédemment, l'interpolation de modèles est un schéma couramment utilisé pour le lissage des paramètres d'un modèle de langage : il est également très utilisé pour l'adaptation d'un modèle de langage.

Plusieurs techniques d'adaptation utilisent l'interpolation de modèles. La fusion d'un modèle généraliste avec un modèle spécifique par interpolation linéaire est la technique la plus simple.

La mixture de modèles est une généralisation de l'interpolation de modèles de langage : les modèles utilisés sont pré-calculés et spécialisés. Supposons que le corpus généraliste couvre plusieurs thèmes différents (par exemple, si le corpus généraliste est issu d'un ensemble d'articles de journal, plusieurs thèmes sont abordés dans ce corpus : sport, politique, etc.). Le modèle généraliste est alors estimé sur l'ensemble du corpus, alors que chaque modèle spécialisé est estimé sur une sous-partie de ce corpus, chaque sous-partie correspondant à un thème particulier.

Une mixture de $k+1$ modèles de langage se définit ainsi :

$$P(w_i|h_i) = \sum_{k=0}^{K} \lambda_{A,k} P_{B,k}(w_i|h_i) \tag{3.7}$$

où $P_{B,0}(w_i|h_i)$ est la probabilité donnée par le modèle généraliste et les probabilités $P_{B,k}(w_i|h_i)$ telles que $k > 0$ sont les probabilités des modèles spécialisés dans un thème. Les coefficients d'interpolation $\lambda_{A,k}$ sont estimé à partir du corpus d'adaptation. Généralement, c'est le critère de maximum de vraisemblance qui est alors utilisé.

Modèles à mémoire cache dynamiques Les modèles à mémoire cache (Kuhn et De Mori, 1990,) sont basés sur l'hypothèse qu'un mot déjà apparu dans le passé récent de l'énoncé a une probabilité de réapparition plus forte que celle suggérée par un modèle de langage statique. En utilisant une fenêtre de texte de taille N, il est possible d'implémenter une mémoire cache stockant les N derniers mots et mise à jour en permanence. Ainsi, par exemple, les fréquences d'*unigrams* $f_t^N(w)$ dans la mémoire cache peuvent être calculées au fur et à mesure du processus de reconnaissance et utilisées pour calculer dynamiquement la probabilité d'apparition du mot w.

Les modèles à mémoire cache sont généralement utilisés pour l'adaptation en les combinant avec des modèles à base de classes, sous la forme suivante :

$$P(w_i|h_i) = \sum_{\{gi\}} P(w_i|g_i) P(g_i|h_i) \tag{3.8}$$

où $\{g_i\}$ est l'ensemble des classes (généralement des classes syntaxiques) pouvant être associées au mot w_i. La composante *n-gram* de classes $P(g_i|h_i)$ est considérée comme statique, c'est-à-dire indépendante de la tâche, alors que la composante d'affectation de classe $P(w_i|g_i)$ est soumise à une adaptation dynamique par mémoire cache : le calcul de cette composante prend en compte les informations contenues dans cette mémoire. On a alors :

$$P(w_i|c_i) = (1-\lambda)P_A(w_i|g_i) + \lambda P_G(w_i|g_i) \tag{3.9}$$

où le coefficient d'interpolation λ est calculé comme indiqué précédemment.

Adaptation par Maximum *a posteriori* Au lieu de combiner les informations au niveau des modèles, il est possible de les combiner directement au niveau des fréquences de mots au moment de l'estimation du modèle de langage. L'utilisation du critère de maximum *a posteriori* est la technique la plus employée : c'est le cas par exemple dans (Chen et Huang, 1999,) ou (Masataki *et al.*, 1997).

L'estimation du modèle adapté à l'aide du critère de maximum *a posteriori* peut s'écrire :

$$P^{MAP}(w_i|h_i) = \begin{cases} \frac{\varepsilon c_A(h_i w_i) + c_G(h_i w_i)}{\varepsilon c_A(h_i) + c_G(h_i)} & \text{si } \varepsilon c_A(h_i w_i) + c_G(h_i w_i) > 0 \\ 0 & \text{sinon} \end{cases} \tag{3.10}$$

où ε est un facteur constant estimé empiriquement, destiné à optimiser l'influence du corpus d'adaptation (Federico, 1996), et où $c_A(h_i w_i)$ et $c_G(h_i w_i)$ sont les nombres d'occurrences de la séquence de mots $h_i w_i$ dans, respectivement, le corpus A et le corpus G .

Dans des travaux plus récents, ε varie en fonction de l'historique h_i (Chen et Huang, 1999,).

Adaptation par spécification de contraintes Les techniques d'adaptation par spécification de contraintes consistent à utiliser le corpus d'adaptation pour en extraire ses caractéristiques les plus significatives. Celles-ci doivent alors être satisfaites par le modèle de langage adapté : elles sont considérées comme des contraintes à respecter. Un des avantages de cette méthode, par rapport à l'interpolation de modèles de langage, est la possibilité de donner plus ou moins d'importance à l'une ou l'autre des contraintes issues de l'observation du corpus d'adaptation.

Historiquement, les modèles de langage estimés à partir de spécifications de contraintes étaient associés à l'utilisation du critère d'entropie maximale (Jaynes, 1957). Son intérêt porte sur sa faculté à traiter les événements non observés dans le corpus d'apprentissage. Cette méthode est maintenant considérée comme un cas particulier de l'estimation par le critère d'information de discrimination minimale (*minimum discrimination information* : MDI), critère actuellement très utilisé pour l'adaptation des modèles de langage.

L'estimation de modèles de langage à partir du critère d'entropie maximale est présentée dans (Della Pietra *et al.*, 1992). Dans le cas de modèles de type *n-gram*, les contraintes sont définies comme des fonctions caractéristiques de tous les *n-grams* observés dans le corpus d'apprentissage. Pour un modèle *trigram*, par exemple, les contraintes sont définies pour les *unigrams*, *bigrams* et *trigrams* observés : ces observations ne concernent en général que les fréquences d'apparition de ces *n-grams*.

Information de discrimination minimale (MDI) A partir d'un ensemble de contraintes sur la distribution θ et d'une distribution *a priori* θ', la distribution θ estimée par le critère d'information de discrimination minimale est une distribution qui satisfait l'ensemble des contraintes tout en minimisant la fonction d'information discriminatoire $D(\theta, \theta')$, appelée distance de Kullback-Leibler, définie par :

$$D(\theta, \theta') = \sum_{w \in} \theta_w log \frac{\theta_w}{\theta'_w} \tag{3.11}$$

(Cover et Thomas, 1991,) montre que cette distance est toujours positive, et devient nulle uniquement si $\theta = \theta'$.

La distribution θ définie par le critère MDI est la distribution respectant les contraintes la plus proche de la distribution *a priori* θ'. Si θ' est une distribution uniforme, le critère MDI devient alors équivalent au critère de maximum d'entropie (ME).

Triggers Les modèles de langage basés sur des *triggers* (déclencheurs) exploitent les dépendances qui existent entre certains mots, dans le cas où l'observation de ces mots est suffisamment corrélée dans le corpus d'apprentissage. Par exemple, dans le cadre d'une utilisation des *triggers* pour la détection de thème, le thème d'un énoncé peut se déduire à partir de certains couples de mots : le couple ("élections", "discours") suffit pour conclure sur la présence d'un thème politique.

Les *triggers* à base de mots (en général des couples de mots) sont identifiés en recherchant les co-occurrences significatives de mots sur une fenêtre de taille N. Des statistiques sont collectées pour chaque couple de mot (x, y). A partir de ces informations statistiques, et en utilisant des techniques basées sur l'information mutuelle, sur le χ^2, ou autre, il est alors possible de choisir les couples de mots qui formeront des *triggers*.

Les statistiques concernant les *triggers* à base de mots peuvent être calculées de manière dynamique et combinées avec un modèle de langage statique pour un historique h_t donné. L'intégration avec un modèle statique se fait alors par les schémas classiques de l'interpolation linéaire ou de repli.

Il est à noter qu'il est également possible, à condition d'exprimer ces statistiques sous la forme de contraintes de distribution, d'utiliser les techniques fondées sur le critère ME ou le critère MDI. C'est le cas dans (Lau *et al.*, 1993), (Rosenfeld, 1996).

Modèles de langage structurés Une autre approche consiste à considérer que les textes généralistes et les textes spécialisés dans une tâche spécifique sont soumis aux mêmes contraintes syntaxiques. L'idée qui vient alors est d'utiliser le corpus généraliste pour extraire les structures syntaxiques. Les informations issues du corpus d'adaptation sont alors combinées à celles portées par ces structures. Il est ainsi possible d'estimer un modèle robuste au niveau de la connaissance des structures syntaxiques et proche de la tâche visée.

Cette approche peut être implémentée à l'aide, par exemple, des modèles de langage structurés (Chelba et Jelinek, 2000,) présentés en section 3.2.3. (Chelba, 2001) montre que l'apprentissage d'un modèle structuré $P_G(w_q|h_q p_q)$ sur un corpus généraliste et la ré-estimation de ses paramètres sur le corpus d'adaptation donnent de meilleurs résultats que l'apprentissage d'un modèle structuré $P_A(w_q|h_q p_q)$ appris directement sur le corpus d'adaptation, pour lequel les données sont moins nombreuses.

3.3.4 Discussion

3.3.4.1 Campagnes d'évaluation et développement industriel

Les techniques d'adaptation des modèles de langage représentent un axe de recherche très intéressant pour le développement de nouvelles applications. Le manque de données est un problème chronique pour les développeurs d'applications commerciales.

Les campagnes d'évaluation de systèmes de reconnaissance ne reflètent pas les problèmes réellement rencontrés par les industriels : ces campagnes fournissent une quantité énorme de données d'apprentissage. Ainsi, les grands laboratoires qui peaufinent leurs systèmes de reconnaissance dans le but de participer à ces campagnes d'évaluation atteignent maintenant des performances impressionnantes lors de ces évaluations.

Malheureusement, les performances de ces systèmes lors des campagnes d'évaluation ne reflètent pas les performances auxquelles peuvent prétendre les systèmes de reconnaissance développés pour de nouvelles applications. Cela est d'autant plus vrai pour les nouvelles applications de dialogue, en plein essor, qui généralement abordent des domaines sémantiques jamais abordés auparavant et qui disposent de peu d'informations réutilisables.

Il est donc souhaitable d'encourager les travaux basés sur l'adaptation des modèles, que ce soient des modèles acoustiques ou des modèles de langages. En particulier, il serait intéressant d'organiser des campagnes d'évaluation de techniques d'adaptation.

3.3.4.2 Propositions

La contribution de cette thèse à la modélisation du langage touche à deux aspects de l'adaptation des modèles de langage :

1. Le manque de données d'apprentissage.
2. La spécialisation d'un modèle de langage en fonction de l'état du dialogue.

L'intégration d'automates stochastiques à états finis dans un modèle *n-gram* revient à intégrer au sein de ce type de modèles probabilistes des grammaires régulières locales. Ces grammaires locales permettent de modéliser des événements non rencontrés dans le corpus d'apprentissage, comme c'est le cas avec l'utilisation de classes de mots. A la grande différence des classes de mots, les grammaires locales permettent également d'intégrer des contraintes locales fortes, et d'élargir la longueur des contraintes d'un modèle *n-gram* .

La sélection dynamique d'un modèle de langage spécialisé est également une approche intéressante du point de vue de l'adaptation de modèle pour une application de dialogue. Il est déjà intéressant d'adapter un modèle de langage à une application spécifique : nous proposons de spécialiser les modèles non pas au niveau de l'application, mais également au niveau du type de phrase prononcée par l'utilisateur.

3.4 Évaluation d'un système de reconnaissance de la parole

Pour comparer les performances de deux systèmes de reconnaissance de la parole, il est nécessaire de disposer d'un outil adéquat. La mesure la plus répandue est le taux d'erreurs sur les mots.

3.4.1 Taux d'erreurs sur les mots

Cette mesure se calcule à partir du nombre d'erreurs survenues sur les mots et du nombre de mots de la transcription de la phrase prononcée. Trois types d'erreurs sont possibles :

1. les substitutions : il y a substitution lorsqu'un mot est reconnu à la place d'un autre. Notons Sub le nombre de substitutions observées.
2. les insertions : ce sont les mots qui ont été reconnus sans qu'il existe de mot correspondant, même différent, dans la phrase de référence. Notons Ins le nombre d'insertions observées.
3. les suppressions : il y a suppression lorsqu'un mot de la phrase de référence ne peut être mis en relation avec un mot, même différent, de la phrase reconnue. Notons Sup le nombre de suppressions observées.

Notons OK le nombre de mots observés correctement reconnus.

Le taux d'erreurs sur les mots, noté $w.e.r$ (*word error rate*, en anglais), se calcule de la manière suivante :

$$w.e.r = \frac{Sub + Ins + Sup}{OK + Sub + Sup}$$

Remarque : $OK + Sub + Sup$ est en fait égal au nombre de mots de la transcription de la phrase prononcée, appelée phrase de référence.

3.4.2 Intervalle de confiance

Afin de connaître la fiabilité des résultats des expériences en fonction du nombre d'échantillons utilisés pour les tests, il convient de calculer un intervalle de confiance pour chacun des résultats. Cet intervalle de confiance est calculé en considérant que l'apparition d'une erreur de reconnaissance sur un mot ou sa non-apparition est associée à une variable aléatoire binomiale, dont la distribution dépend des couples (mot reconnu, mot prononcé).

Le taux d'erreurs sur les mots, noté wer_f, est la proportion d'erreurs observées sur le corpus de test. k est le nombre de mots qui constituent le corpus de test. L'intervalle de confiance permet d'estimer, à partir de l'observation de wer_f sur les k échantillons disponibles, l'intervalle de valeurs dans laquelle se situe la proportion wer_p d'erreurs sur les mots pour une population infinie de mots répondant aux critères de l'application.

Un intervalle de confiance est également défini par un niveau $1 - \alpha$ qui permet de déterminer la fiabilité de cet intervalle. Généralement, l'intervalle de confiance est un intervalle à risques symétriques $\alpha/2$, c'est-à-dire que la valeur wer_f observée sur les k échantillons se trouve au centre de cet intervalle.

Pour un nombre d'échantillons k suffisamment grand (habituellement, si $k > 100$), l'expression suivante définit l'intervalle de confiance de niveau $1 - \alpha$ de wer_p (Saporta, 1990) :

$$wer_f - u_{\alpha/2}\sqrt{\frac{wer_f(1 - wer_f)}{k}} < wer_p < wer_f + u_{\alpha/2}\sqrt{\frac{wer_f(1 - wer_f)}{k}} \qquad (3.12)$$

où la valeur de $u_{\alpha/2}$ dépend de α et est disponible dans la table dite de *Student*.

L'intervalle de confiance le plus couramment utilisé est l'intervalle de niveau $1 - 0.95 = 0.05$, appelé aussi intervalle de confiance à 95%. Dans ce cas, $u_{0.425} = 1,96$.

3.5 Présentation générale d'un système de dialogue

3.5.1 Architecture modulaire

Un système de dialogue oral homme-machine doit être capable d'effectuer correctement plusieurs tâches pour que l'utilisateur humain puisse communiquer naturellement avec lui. Ces diverses tâches sont traitées à plusieurs niveaux par des modules spécialisés, et peuvent être regroupées en quatre grandes catégories :

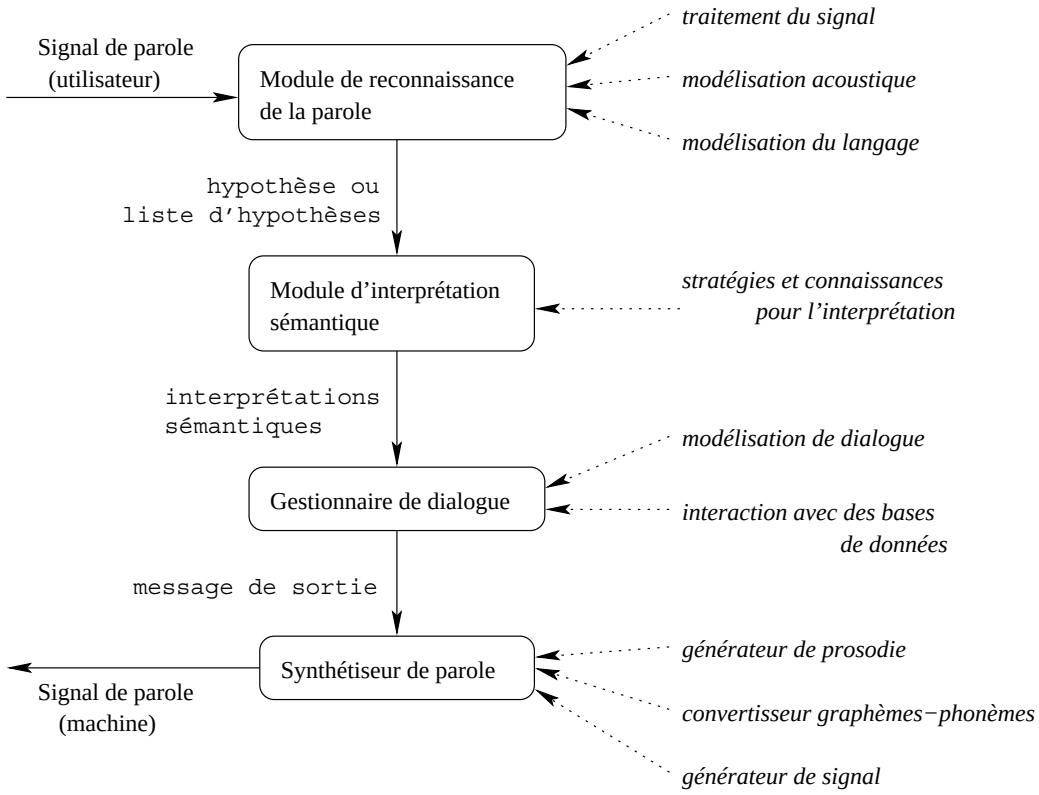


FIG. 3.3 – *Architecture général d'un système de dialogue*

1. La reconnaissance de la parole : elle consiste à déterminer quelle phrase a été prononcée par l'utilisateur humain en fonction du signal de parole capté par un microphone. Cette fonctionnalité fait appel à des outils de traitement du signal, de modélisation acoustique et de modélisation linguistique.
2. L'interprétation sémantique : elle permet de donner du sens à la phrase reconnue par le module de reconnaissance de la parole en procédant, par exemple, à une analyse grammaticale, ou en détectant dans la phrase reconnue des mots clés qui permettent d'activer des structures sémantiques. Plusieurs techniques d'interprétation sont décrites dans (Kuhn et De Mori, 1998,).
3. La gestion du dialogue : établie à partir de l'interprétation sémantique et à partir de la connaissance de l'historique du dialogue, elle permet de guider correctement la conversation. Par exemple, si l'utilisateur pose une question, ou émet une requête, le gestionnaire de dialogue interrogera la base de données afin de satisfaire aux exigences de l'utilisateur. Si une incompréhension apparaît, due à l'utilisation par l'utilisateur de mots qui ne font pas partie du vocabulaire du système, le gestionnaire de dialogue doit être capable de réagir afin d'amener l'utilisateur à reformuler ses propos différemment. (Sadek et De Mori, 1998,) présente diverses approches de gestion du dialogue.
4. La synthèse de la parole : elle consiste à générer les sons permettant d'exprimer la phrase déterminée par le gestionnaire de dialogue. Le synthétiseur manipule des outils de génération de prosodie, de conversion de graphèmes en phonèmes, etc. Des informations plus précises se trouvent dans (Sorin et De Mori, 1998,).

La figure 3.3 représente un système de dialogue basé sur les quatre phases présentées ci-dessus. Dans l'architecture présentée par cette figure, les différents modules sont indépendants, et n'échangent qu'un minimum d'information.

3.5.2 Spécificités de la reconnaissance de la parole dans une application de dialogue

La reconnaissance de la parole est contrainte par différents phénomènes selon les applications pour lesquelles elle est utilisée. Le traitement de la parole continue spontanée a des caractéristiques qui ne se rencontrent pas dans une application de dictée vocale ou dans une application de commandes vocales. Quant à la modélisation du langage, sa nature varie même en fonction du système de dialogue.

3.5.2.1 Particularités du traitement de la parole spontanée

En parole spontanée, l'environnement sonore n'est pas toujours maîtrisé. Des phénomènes extra-linguistiques interfèrent dans la communication homme-machine : bruits extérieurs, raclements de gorge, variations émotionnelles de l'utilisateur (énervement, rire, ...), etc.

L'élocution de l'utilisateur est souvent marquée par des hésitations, des reprises, ou encore des réparations. Ces différentes caractéristiques du discours spontané

génèrent une plus grande variabilité des phrases prononcées : la même phrase peut être prononcée correctement, ou bien avec quelques hésitations sur certains mots, ou sur d'autres mots, ou encore elle peut être prononcée avec des erreurs que l'utilisateur s'empresse de corriger. A cette variabilité, s'ajoute la faible quantité de corpus généralement disponible pour développer une nouvelle application de dialogue. Ceci rend l'estimation de modèles de langage robustes plus difficile que dans le cadre d'une application de dictée vocale qui peut s'appuyer sur de grosses quantités de documents écrits.

3.5.2.2 Modèles de langage et types de dialogue

Deux grands types de dialogues homme-machine sont possibles :

1. Le dialogue à l'initiative du système : ce type de dialogue laisse peu de liberté à l'utilisateur. Le système guide l'utilisateur tout au long du dialogue et attend de l'utilisateur des réponses très précises. Ce type de dialogues se satisfait d'un modèle de langage contraint, à base de grammaires régulières par exemple.
2. Le dialogue à l'initiative de l'utilisateur : le dialogue est ouvert, toute liberté est laissée à l'utilisateur. Bien entendu, le dialogue est limité par le lexique utilisé et par les performances du système de reconnaissance. Un modèle de langage permettant de couvrir une grande partie du langage est nécessaire. Les modèles *n-grams* sont les modèles généralement utilisés pour ce type de dialogue

Le système de dialogue de notre étude (le démonstrateur AGS) est un système qui laisse l'initiative à l'utilisateur. Malgré tout, comme le thème du dialogue est limité, la stabilité des phrases prononcées pour cette application offre la possibilité d'estimer des modèles de langage *n-grams* exploitables sur un petit corpus d'apprentissage.

Deuxième partie

Contributions

Chapitre 4

Intégration d'automates stochastiques à états finis dans un modèle *n-gram*

Sommaire

4.1 Introduction 60
4.1.1 Motivations 60
4.1.2 Contexte de l'étude 61
4.1.3 Automates à états finis et classes de séquences de mots 61
4.2 Architecture du modèle 63
4.2.1 Composantes 63
4.2.2 Combinaison des composantes 63
4.3 Probabilité d'une phrase selon le modèle intégré 64
4.3.1 Intégration d'un automate stochastique dans un modèle *n-gram* 64
4.3.2 Généralisation 65
4.3.3 Exemple 68
4.4 Estimation des paramètres du modèle 69
4.4.1 Modèles *internes* 69
4.4.2 Modèle *externe* 74
4.5 Utilisation du modèle hybride dans un système de reconnaissance de la parole 76
4.5.1 Utilisation du modèle hybride 76
4.5.2 Algorithme de détection d'une séquence de mots spécifique . . 76
4.5.3 Séquence de mots détectée : insertion d'une transition 77
4.6 Expérimentations 78
4.6.1 Description des données expérimentales 78
4.6.2 Evaluation des modèles de langage 82

4.1 Introduction

Les travaux combinant les approches formelles et statistiques du traitement automatique du langage naturel se développent dans le but de profiter de la complémentarité de ces deux approches. C'est le cas par exemple dans (Salomaa, 1969), (El-Bèze et Spriet, 1995,) ou (Chelba et Jelinek, 2000,). Notre proposition de modèle de langage hybride s'inscrit dans cette démarche.

Nous présentons un modèle de langage de type *n-gram* enrichi de grammaires régulières locales représentées sous forme d'automates stochastiques à états finis.

4.1.1 Motivations

Des travaux publiés sur l'utilisation de contraintes grammaticales dans un système d'étiquetage syntaxique automatique (El-Bèze et Spriet, 1995,) ont démontré l'intérêt de l'enrichissement d'un modèle de langage stochastique par des règles grammaticales. Ces règles peuvent être associées à des phénomènes particuliers : c'est le cas par exemple des structures syntaxiques spécifiques à l'énonciation d'une date ou encore, dans le cadre d'une application de dialogue, pour l'énonciation de certains types de requêtes.

Il semble intéressant de pouvoir utiliser des informations connues *a priori* pour modéliser un phénomène particulier : ces informations sont facilement représentées sous la forme d'une grammaire locale. En intégrant ces grammaires locales dans un modèle de type *n-gram*, la souplesse et la couverture du modèle stochastique sont associées aux connaissances *a priori* des grammaires et à leur capacité à décrire des contraintes locales.

Enfin, trois caractéristiques intéressantes sont liées à cette approche :

1. Les mêmes grammaires locales peuvent être utilisées dans des applications différentes. Ainsi, si une grammaire locale a été produite pour le traitement d'une application spécifique, elle pourra être réutilisée ultérieurement dans une application différente.
2. L'utilisation des grammaires locales permet, tout comme l'utilisation de classes de mots dans les modèles *n-grams* à base de classes, une certaine généralisation : des événements non vus dans le corpus d'apprentissage peuvent être modélisés. Par exemple, dans le cas d'une grammaire locale décrivant l'énonciation d'une date, il n'est pas nécessaire d'avoir rencontré toutes les dates possibles (1er janvier 2002, 2 février 2003, ...) dans le corpus d'apprentissage pour aboutir à une modélisation robuste de ce phénomène[1].
3. La taille des séquences de mots décrites par les grammaires locales peut dépasser la taille d'un *n-gram*. Par exemple, pour un modèle *trigram*, la taille de l'unité de modélisation ne dépassera pas trois mots, alors que dans le cas

[1]Néanmoins, il ne faut pas perdre de vue la contrainte que peut exercer une date particulière sur son contexte d'énonciation (et réciproquement). Par exemple, la date du 11 septembre 2001 est fortement attachée aux actes terroristes qui ont touché les États-Unis ce jour-là.

d'une grammaire régulière, cette longueur dépend des règles utilisées et peut dépasser cette taille. L'intégration de grammaires locales dans un modèle de type *n-gram* permet donc de modéliser des événements composés de plus de n mots.

4.1.2 Contexte de l'étude

Les travaux présentés dans (Riccardi *et al.*, 1996) ont montré qu'il est possible de créer et d'utiliser dans un système de reconnaissance de la parole un modèle de langage basé uniquement sur des automates stochastiques. Le modèle obtenu est en fait proche d'un modèle *n-gram*, puisque la seule différence notable provient de son architecture. Dans le modèle présenté ici, dont une première étude a été présentée dans (Nasr *et al.*, 1999), les automates stochastiques ne sont utilisés que localement : ils sont intégrés dans la structure d'un modèle de type *n-gram* sous la forme de classes de séquences de mots. Dans (Wang *et al.*, 2000) l'utilisation des automates stochastiques a été étendue à l'utilisation de grammaires probabilistes hors-contexte. Dans le cadre d'une application de dialogue comme celle qui fait l'objet de notre étude, l'utilisation de grammaires régulières s'avère plus appropriée : la faible variabilité des phrases et des structures syntaxiques ne nécessite pas l'utilisation de grammaires probabiliste hors-contexte, surtout quand peu de données d'apprentissage sont disponibles. Néanmoins, les paramètres stochastiques des grammaires probabilistes hors-contexte peuvent être estimés sur un corpus d'apprentissage différent du corpus spécifique à l'application visée. Ceci est également vrai pour les grammaires régulières locales utilisées sous forme d'automates à états finis dans le modèle présenté ici.

4.1.3 Automates à états finis et classes de séquences de mots

Un automate à états finis est associé à une classe de séquences de mots : chaque chemin de l'automate représente en fait une séquence de mots, comme l'illustre la figure 4.1.

La représentation d'une classe de séquences de mots sous forme d'automate est plus pratique qu'une représentation énumérative :

1. la représentation sous forme d'automate est plus compacte (voir l'exemple du mot "je" dans la figure 4.1, où ce mot apparaît 8 fois dans la représentation énumérative et une seule fois dans l'automate),
2. elle facilite la détection dans une phrase d'une séquence de mots spécifique. En effet, l'automate associé à une classe peut être utilisé comme automate accepteur d'une séquence de mots. Cette détection est utile pour le calcul de la probabilité d'une phrase avec le modèle de langage présenté ici.

Les automates utilisés sont des automates stochastiques à états finis : à chaque état final de l'automate est associée une probabilité. Cette probabilité correspond à

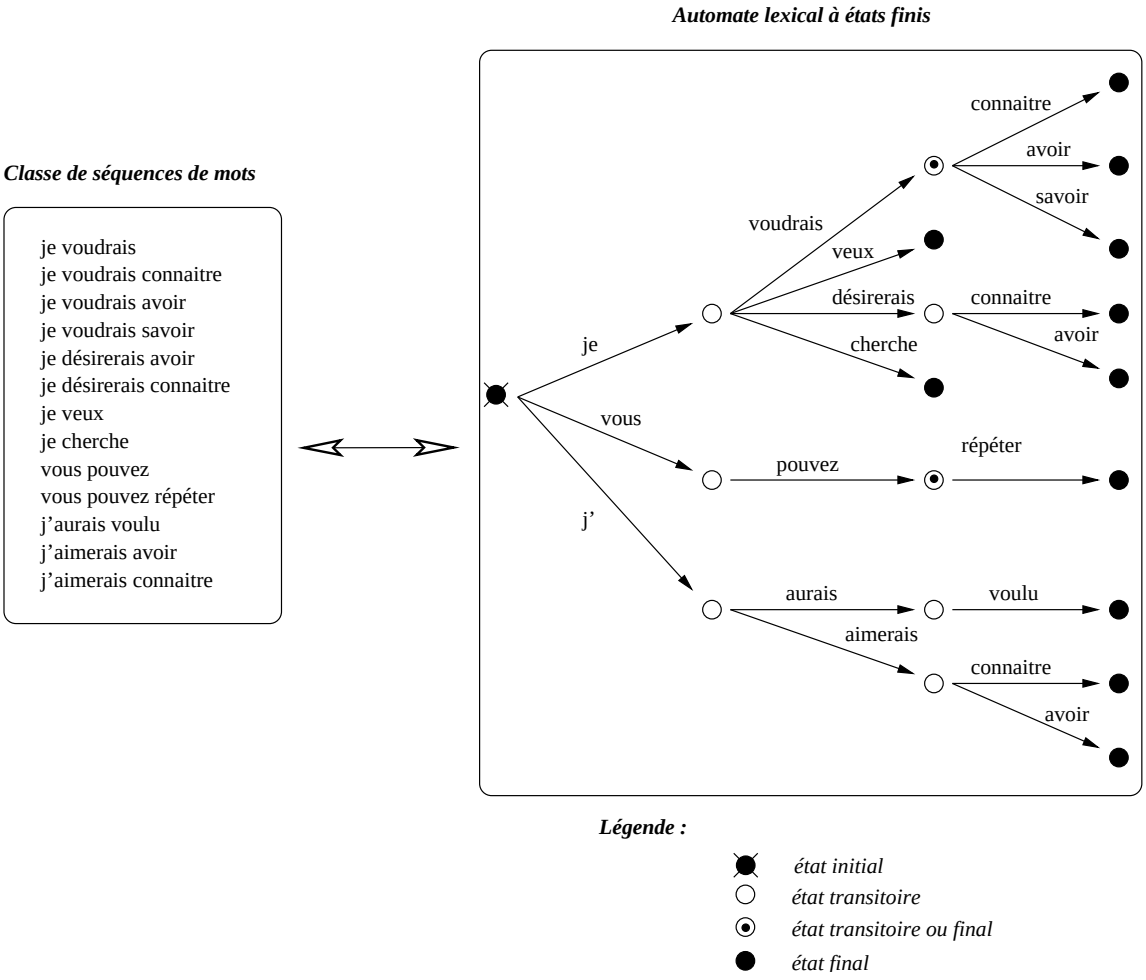


FIG. 4.1 – *Classe de séquences de mots et exemple d'automate lexical à états finis associé.*

la probabilité d'apparition de la séquence reconnue dans la classe de séquences de mots que représente l'automate. Un seul chemin se termine sur un état final.

Il y a d'autres manières de représenter une classe de séquences de mots sous forme d'automate lexical à états finis. Nous préférons utilisé des automates sous forme d'arbres des préfixes (qui factorisent les séquences de mots à gauche) car cette forme, en plus de compacter l'information, permet de connaître précisément le chemin parcouru dans l'automate pour atteindre un de ses états.

4.2 Architecture du modèle

4.2.1 Composantes

Pour intégrer des automates stochastiques à états finis dans un modèle de type *n-gram*, le modèle de langage doit être constitué de deux composantes :

1. Une composante dite probabilité *externe*, qui correspond à la probabilité d'apparition de l'automate après un historique donné, l'historique étant composé de mots ou d'automates. Cette probabilité est estimée comme une probabilité issue d'un modèle *n-gram* classique, l'automate étant considéré comme un mot. L'ensemble de ces probabilités, notées P_{ng}, où n vient de la taille des *n-grams modélisés*, est assemblé sous l'appellation de modèle externe.
2. Une composante dite probabilité *interne*, qui permet d'intégrer au modèle externe des modèles locaux, appelés modèles internes. La probabilité interne correspond à la probabilité d'apparition d'une séquence de mots pour un automate donné. La probabilité d'apparition d'une séquence de mots $w_i...w_j$ dans un automate A_k est notée $P_a(w_i...w_j|A_k)$.

4.2.2 Combinaison des composantes

La figure 4.2 montre un exemple d'architecture du modèle de langage, représenté sous la forme d'un modèle de Markov caché. Dans cette figure, un modèle local est intégré à l'aide de l'automate A_1 (mis en valeur dans la figure par un rectangle aux angles arrondis formé de tirets et de points). Au sein de cet automate, les changements d'états (représentés par des flèches aux traits pleins) ne sont pas soumis explicitement à des probabilités. En fait, la probabilité du chemin dans l'automate est portée par la transition entre l'état final de ce chemin vers l'état qui relie l'automate aux autres entités (mots et automates) du modèle externe : ces transitions sont représentées dans la figure par des flèches constituées de points. Ces transitions ne sont pas associées à des mots : l'arrivée dans un état final implique automatiquement un passage sur la transition finale associée.

Le modèle externe permet d'établir des transitions liant les mots w_1 et w_2 et l'automate A_1entre eux . Ces transitions (représentées par des pointillés) sont porteuses des probabilités P_{ng} du modèle externe. Le modèle interne lié à l'automate A_1 n'intervient que pour donner une probabilité interne à une séquence de mots décrite par l'automate A_1.

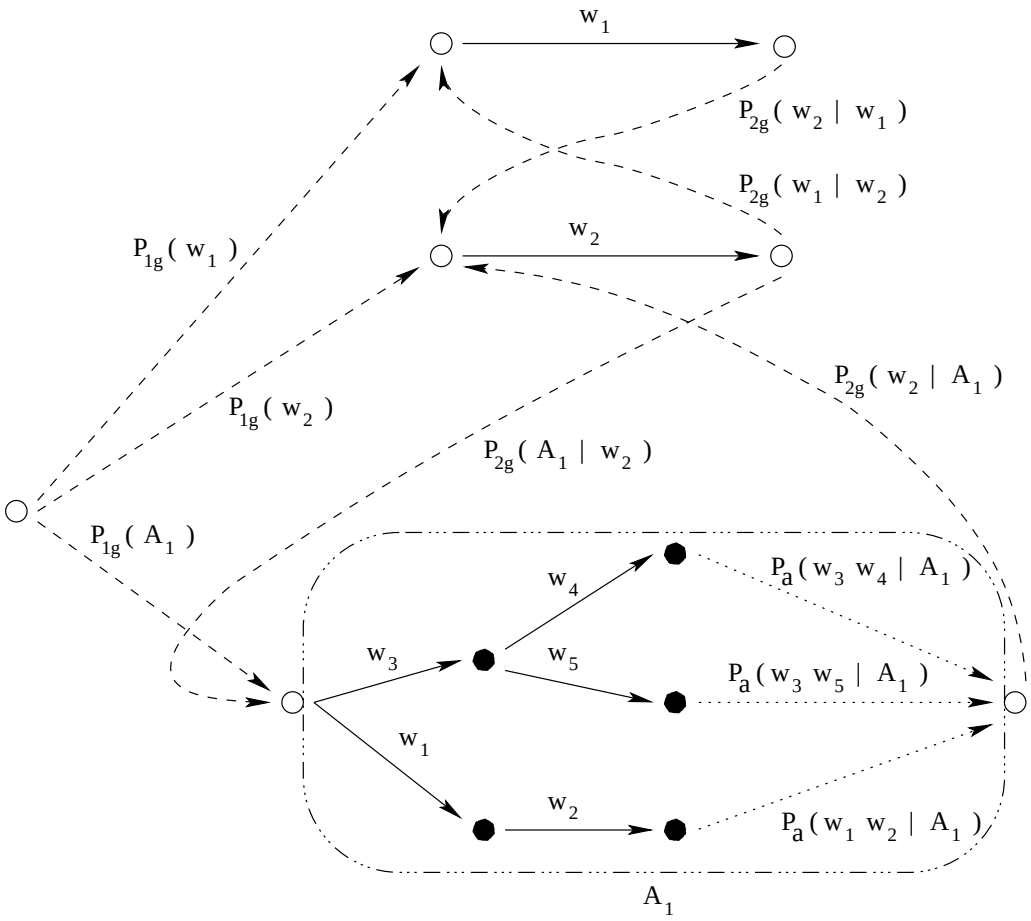


FIG. 4.2 – *Architecture du modèle de langage*

4.3 Probabilité d'une phrase selon le modèle intégré

Pour des raisons de clarté, il est préférable d'étudier un cas particulier pour bien comprendre la structure du modèle. En premier lieu, nous nous concentrerons sur la manière dont est intégré un unique automate stochastique à états finis dans un modèle *n-gram*. L'étude de la généralisation en sera simplifiée.

4.3.1 Intégration d'un automate stochastique dans un modèle *n-gram*

Soit la phrase $W_1^N = w_1 w_2 ... w_N$. Cette phrase peut s'écrire de la manière suivante :

$$W_1^N = W_1^i W_{i+1}^j W_{j+1}^N \tag{4.1}$$

Supposons que la séquence de mots W_{i+1}^j est reconnue par un et un seul automate A_k. Supposons également qu'aucune autre séquence de mots de W_1^N n'est reconnue par un automate. La phrase W_1^N peut se ré-écrire :

$$W_1^N = W_1^i A_k(W_{i+1}^j) W_{j+1}^N \tag{4.2}$$

où le terme $A_k(W_{i+1}^j)$ signale l'acceptation par l'automate A_k de la séquence de mots W_{i+1}^j.

La probabilité de la phrase W_1^N , avec les hypothèses précédentes, est donnée par la formule suivante :

$$\begin{aligned} P(W_1^N) = & \quad \alpha P(W_1^i) P(A_k(W_{i+1}^j)|W_1^i) P(W_{j+1}^N|A_k W_1^i) \\ & + \quad (1-\alpha) P(W_1^i) P(W_{i+1}^j|W_1^i) P(W_{j+1}^N|W_1^i W_{i+1}^j) \end{aligned} \tag{4.3}$$

Explications :

- La formule est composée d'une somme afin de prendre en compte les deux segmentations possibles de la phrase W_1^N (une segmentation sans automate et une segmentation prenant en compte l'automate A_k). Les coefficients α et $1-\alpha$, compris entre 0 et 1, sont nécessaires pour assurer la normalisation de la formule.
- Les termes $P(W_1^i)$, $P(W_{i+1}^j|W_1^i)$, et $P(W_{j+1}^N|W_1^i W_{i+1}^j)$ sont calculés de la même manière que pour un modèle *n-gram* classique.
- Le terme $P(A_k(W_{i+1}^j)|W_1^i)$ se décompose ainsi :

$$P(A_k(W_{i+1}^j)|W_1^i) = P(W_{i+1}^j|A_k W_1^i) P(A_k|W_1^i) \tag{4.4}$$

  où $P(A_k|W_1^i)$ est la probabilité d'apparition de l'automate A_k selon l'historique W_1^i. Cette probabilité est calculée de la même manière que pour un modèle *n-gram* classique en remplaçant[2] chaque occurrence d'une séquence de mots dans le corpus d'apprentissage par le nom de l'automate qui la reconnaît.
- Le terme $P(W_{i+1}^j|A_k W_1^i)$ correspond à la probabilité donnée par l'automate A_k à la séquence W_{i+1}^jen présence de l'historique W_1^i. Comme le terme $P(A_k|W_1^i)$ intègre déjà une contrainte de l'historique W_1^i sur l'automate A_k, nous proposons de faire l'approximation suivante : $P(W_{i+1}^j|A_k W_1^i) \cong P(W_{i+1}^j|A_k)$, où $P(W_{i+1}^j|A_k)$ est la probabilité donnée par l'automate A_k à la séquence de mots W_{i+1}^j, c'est-à-dire la probabilité d'apparition de la séquence W_{i+1}^jdans la classe de séquence de mots C_k que représente l'automate A_k.

La probabilité de la phrase W_1^N, peut alors s'écrire :

$$\begin{aligned} P(W_1^N) = & \quad \alpha P(W_1^i) P(W_{i+1}^j|A_k) P(A_k|W_1^i) P(W_{j+1}^N|W_1^i A_k) \\ & + \quad (1-\alpha) P(W_1^i) P(W_{i+1}^j|W_1^i) P(W_{j+1}^N|W_1^i W_{i+1}^j) \end{aligned} \tag{4.5}$$

4.3.2 Généralisation

4.3.2.1 Probabilité d'un segment

Il est possible de considérer qu'une séquence de mots appartient à une classe de séquences de mots composée uniquement d'elle seule.

[2] une explication détaillée est présentée en section 4.4

Appelons séquence de mots *unique* une séquence de mots qui n'appartient qu'à la classe qu'elle compose seule. Un segment est une séquence de mots *unique* ou une séquence de mots appartenant à une classe composée de plusieurs séquences.

La probabilité d'apparition en fonction d'un historique h du segment s appartenant à la classe c s'écrit :

$$P(s|h) = P_a(s|c)P_g(c|h) \tag{4.6}$$

telle que :

- $P_a(s|c)$ est la probabilité *interne*, qui est égale à 1 si le segment s est une séquence de mots *unique*,
- $P_g(c|h)$ est la probabilité *externe*, qui est égale à $P_g(s|h)$ si s est une séquence de mots *unique*. P_g est une probabilité analogue à une probabilité issue d'un modèle *n-gram* classique.

La formule (4.6) devient :

$$P(s|h) = \begin{array}{ll} P_g(s|h) & \text{si } s \text{ est une séquence de mots } unique \\ P_a(s|c)P_g(c|h) & \text{sinon} \end{array} \tag{4.7}$$

4.3.2.2 Segmentations multiples

Dans l'étude du cas particulier de l'intégration d'un seul automate dans un modèle de langage *n-gram* (en sous-section 4.3.1), nous avons fait l'hypothèse qu'il n'existait que deux segmentations de la phrase W_1^N : la première segmentation ne comprenait aucun automate, la seconde ne comprenait qu'un et un seul automate, l'automate A_k. Cette hypothèse était utile pour se focaliser sur l'intégration d'un automate dans le modèle *n-gram*. Cependant, une même phrase peut contenir plusieurs séquences de mots reconnues par plusieurs automates : dès lors, diverses segmentations sont à considérer (voir figure 4.3).

Notons Ξ_W l'ensemble des segmentations ς_z possibles pour la phrase W. Notons $P(W, \varsigma_z)$ la probabilité[3] de la phrase W sous la forme segmentée ς_z.

Alors :

$$P(W) = \sum_{\varsigma_z \in \Xi_W} \alpha_z P(W, \varsigma_z) \tag{4.8}$$

avec :

$$\sum_{\varsigma_z \in \Xi_W} \alpha_z = 1$$

Deux hypothèses sont possibles :

[3] Précédemment, nous ne considérions que deux segmentations pour la phrase W_1^N. Les probabilités $P(W_1, \varsigma_0)$ et $P(W_1^N, \varsigma_1)$ correspondaient respectivement à la segmentation sans automate et la segmentation avec l'automate A_k. Nous avions alors : $P(W_1, \varsigma_0) = P(W_1^i)P(W_{i+1}^j|W_1^i)P(W_{j+1}^N|W_1^i W_{i+1}^j)$ et $P(W_1^N, \varsigma_1) = P(W_1^i)P(W_{i+1}^j|A_k)P(A_k|W_1^i)P(W_{j+1}^N|W_1^i A_k)$

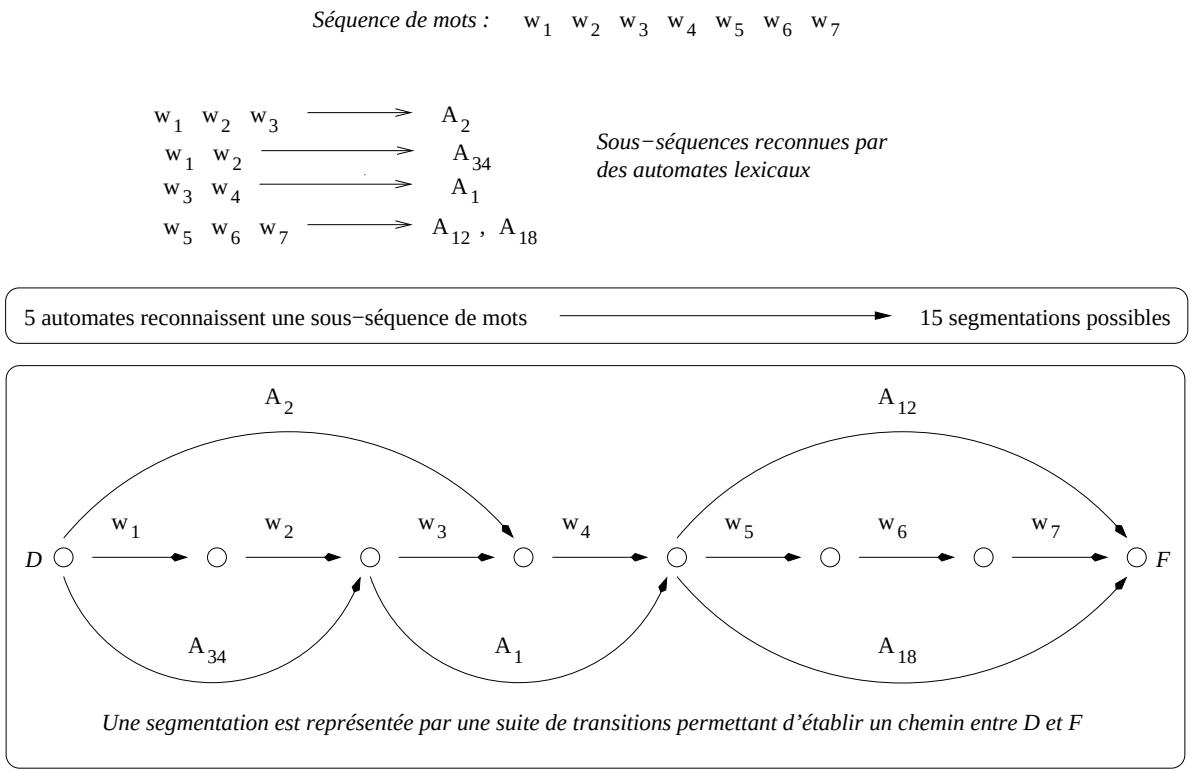


FIG. 4.3 – *Exemple de segmentations possibles d'une séquence de mots*

1. Tous les coefficients α_z sont égaux. Dès lors, si N_ς est le nombre de segmentations possibles de la phrase W, on a $N_\varsigma \alpha_z = 1$ et la formule 4.8 devient :

$$P(W) = \alpha_z \sum_{\varsigma_z \in \Xi_W} P(W, \varsigma_z) = \frac{1}{N_\varsigma} \sum_{\varsigma_z \in \Xi_W} P(W, \varsigma_z) \tag{4.9}$$

2. La valeur la plus grande des probabilités $P(W, \varsigma_i)$ est suffisante pour établir une approximation de la somme des valeurs de ces probabilités[4].

La probabilité de la phrase W se calcule alors de la manière suivante :

$$P(W) = P(W, \underset{\varsigma_z \in \Xi_W}{argmax}\, P(W, \varsigma_z)) \tag{4.10}$$

ou encore :

$$\boxed{P(W) = \underset{\varsigma_z \in \Xi_W}{max}\, P(W, \varsigma_z)} \tag{4.11}$$

[4]Cette approximation est communément acceptée en modélisation stochastique du langage. C'est par exemple le cas dans (Riccardi *et al.*, 1996). En fait, il est vrai que la différence entre la somme des probabilités et la valeur de la probabilité la plus grande est relativement importante. Cependant, comme les modèles de langage sont généralement utilisés pour comparer des hypothèses de phrases, c'est surtout au niveau du pouvoir de discrimination de ces hypothèses que la somme des probabilités est considérée comme proche du critère de probabilité maximale.

Remarque : le terme $\frac{1}{N_\varsigma}$ de la formule (4.9) étant constant, il n'est pas utile pour le calcul de la segmentation de probabilité maximale. Il ne se trouve donc plus dans les formules (4.10) et (4.11).

4.3.3 Exemple

Soit la phrase S suivante : "Je voudrais connaître les prévisions météorologiques pour le Vaucluse". Supposons que les séquences de mots "je voudrais connaître", "les prévisions météorologiques" et "pour le Vaucluse" soit reconnues respectivement par les automates A_1, A_2 et A_3.

Les différentes segmentations ς_z possibles de S sont :

- ς_0 : Je voudrais connaître les prévisions météorologiques pour le Vaucluse
- ς_1 : $[A_1]$ les prévisions météorologiques pour le Vaucluse
- ς_2 : Je voudrais connaître $[A_2]$ pour le Vaucluse
- ς_3 : Je voudrais connaître les prévisions météorologiques $[A_3]$
- ς_4 : $[A_1]$ $[A_2]$ pour le Vaucluse
- ς_5 : $[A_1]$ les prévisions météorologiques $[A_3]$
- ς_6 : Je voudrais connaître $[A_2]$ $[A_3]$
- ς_7 : $[A_1]$ $[A_2]$ $[A_3]$

Supposons utiliser un modèle *bigram* dans lequel les automates stochastiques ont été intégrés. On note P_{2g} les probabilités issues du modèle *bigram* (probabilités que nous avons précédemment appelées probabilités *externes*) et P_a les probabilités d'apparition d'une séquence de mots dans un automate (probabilités *internes*).

Notons $< s >$ et $< /s >$ les symboles de début et de fin de phrase. Voici le détail du calcul de la probabilité de certaines segmentations ς_z de S, caractéristiques des différents cas possibles :

$$\begin{aligned} P(S,\varsigma_0) = \; & P_{2g}(je|<s>) \times P_{2g}(voudrais|je) \times P_{2g}(connaitre|voudrais) \\ & \times ... \times P_{2g}(Vaucluse|le) \times P_{2g}(</s>|Vaucluse) \end{aligned}$$

$$\begin{aligned} P(S,\varsigma_1) = \; & [P_a(je\_voudrais\_connaitre|A_1)P_{2g}(A_1|<s>)] \times P_{2g}(les|A_1) \\ & \times P_{2g}(previsions|les) \times ... \times P_{2g}(</s>|Vaucluse) \end{aligned}$$

...

$$\begin{aligned} P(S,\varsigma_4) = \; & [P_a(je\_voudrais\_connaitre|A_1)P_{2g}(A_1|<s>)] \\ & \times [P_a(les\_previsions\_meteorologiques|A_2)P_{2g}(A_2|A_1)] \\ & \times P_{2g}(pour|A_2) \times ... \times P_{2g}(</s>|Vaucluse) \end{aligned}$$

...

$$\begin{aligned} P(S,\varsigma_7) = \; & [P_a(je\_voudrais\_connaitre|A_1)P_{2g}(A_1|<s>)] \\ & \times [P_a(les\_previsions\_meteorologiques|A_2)P_{2g}(A_2|A_1)] \\ & \times [P_a(pour\_le\_Vaucluse|A_3)P_{2g}(A_3|A_2)] \\ & \times P_{2g}(</s>|A_3) \end{aligned}$$

Notons Ξ_S l'ensemble des segmentations ς_z possibles de la phrase S.

On a : $\Xi_S = \{\varsigma_0, \varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5, \varsigma_6, \varsigma_7\}$.

La valeur de la probabilité donnée par le modèle hybride est donc $P(S) = max_{\varsigma_z \in \Xi_S} P(S, \varsigma_z)$, c'est-à-dire la plus grande des valeurs de probabilités présentées ci-dessus.

4.4 Estimation des paramètres du modèle

4.4.1 Modèles *internes*

Les modèles *internes* représentés par des automates stochastiques sont estimés à partir d'un corpus d'apprentissage et d'un ensemble de règles de grammaire. La figure 4.4 montre le schéma général de l'apprentissage des modèles *internes*.

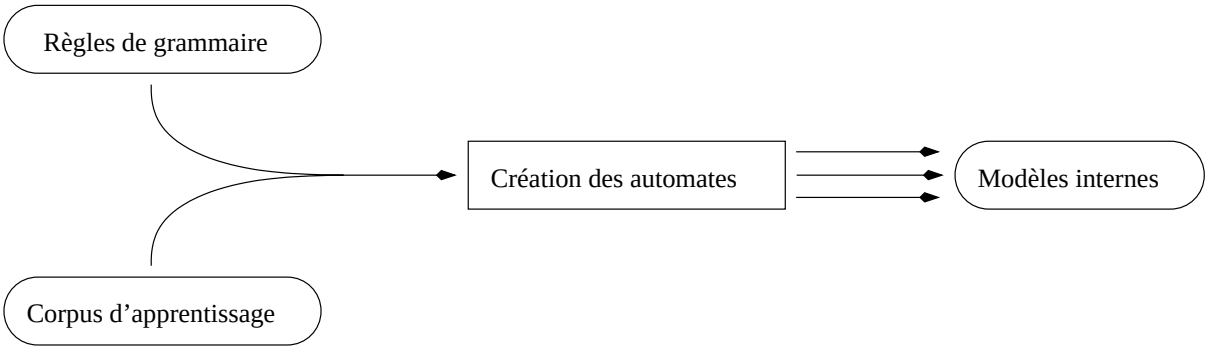


FIG. 4.4 – *Apprentissage des modèles internes (ou locaux)*

4.4.1.1 Création des automates stochastiques

Pour atteindre les objectifs de généralisation, le modèle de langage doit être constitué de classes de séquences de mots homogènes. Intuitivement, nous voulons que le remplacement d'une séquence de mots dans une phrase par une autre séquence de mots de la même classe ne rende pas la phrase incorrecte du point de vue grammatical. Il est possible cependant que le remplacement provoque une altération du sens.

Par exemple, si l'on prend la classe présentée dans la figure 4.1 et la phrase : "je voudrais connaître le numéro de téléphone de M. X", il est possible de remplacer "je voudrais connaître" par "pouvez-vous répéter", ce qui donne : "pouvez-vous répéter le numéro de téléphone de M.X". La phrase obtenue est correcte d'un point de vue syntaxique, mais n'a pas la même signification que la phrase initiale.

La méthode de création des automates stochastiques dépend de plusieurs facteurs, comme l'application de reconnaissance de la parole visée par l'utilisation du modèle de langage hybride ou le corpus d'apprentissage disponible. Dans le cadre de notre

étude, nous disposons de données concernant une application de dialogue. Les principales caractéristiques de ces données sont[5] :

1. un petit lexique (880 mots)
2. un corpus d'apprentissage de taille modeste (9 842 phrases)
3. une application de serveur vocal destiné à une tâche bien déterminée (recherche d'emploi ou renseignements météorologiques)
4. une faible variabilité des phrases prononcées

Cet environnement a fortement conditionné l'approche que nous avons définie pour créer des automates stochastiques et atteindre les objectifs de généralisation et d'intégration de connaissances *a priori* (généralisation et intégration nécessaires pour pallier le manque de données d'apprentissage).

L'approche proposée ici est composée de quatre étapes :

1. l'extraction de séquences de mots spécifiques,
2. la classification de ces séquences de mots,
3. la fusion de classe,
4. la compilation des classes obtenues sous forme d'automates stochastiques à états finis.

4.4.1.2 Extraction des séquences de mots

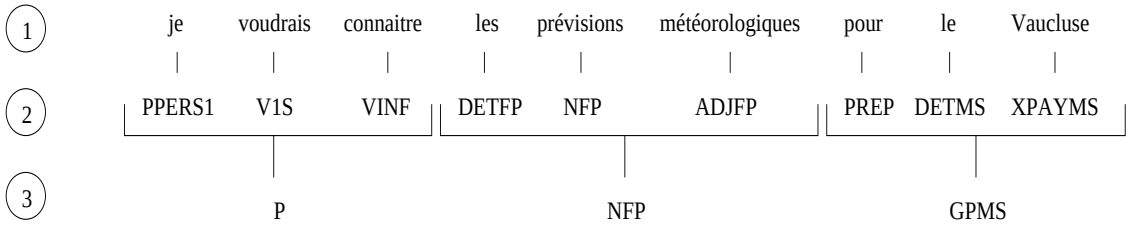
Cette première étape consiste à extraire du corpus d'apprentissage des séquences de mots. Ces séquences de mots sont des séquences de mots liés par des contraintes syntaxiques, et représentent des entités grammaticales (groupe nominal, groupe verbal) appelées syntagmes.

Pour extraire ces séquences de mots du corpus d'apprentissage, il est nécessaire de produire manuellement un ensemble de règles de grammaire définies sur des classes syntaxiques[6]. De plus, le corpus d'apprentissage doit être étiqueté avec ces classes. Ceci nécessite donc trois phases :

1. Étiquetage du corpus d'apprentissage : pour cela, nous utilisons le système ECSta (El-Bèze et Spriet, 1995,), système d'étiquetage syntaxique automatique développé par le LIA.
2. Production des règles de grammaire : cette phase est effectuée manuellement, à l'aide de connaissances linguistiques *a priori*.
3. Analyse grammaticale partielle du corpus d'apprentissage : cette analyse consiste à détecter les séquences de mots décrites par les règles de grammaire. Ces séquences de mots constitueront les éléments des classes de séquences de mots.

[5] Plus de détails en section 4.6.1
[6] Cf. Annexe A

La figure 4.5 illustre ce processus : en 2, la phrase de l'étape 1 est étiquetée avec des étiquettes syntaxiques. En 3, ces étiquettes sont analysées afin de détecter les syntagmes. La grammaire utilisée pour l'analyse partielle se trouve en annexe B. En 4, chaque syntagme extrait est associé à son contexte gauche et son contexte droit, qui sont les séquences de mots apparaissant à gauche et à droite du syntagme extrait et qui représentent chacun une entité grammaticale définie par la grammaire utilisée pour l'analyse.



4

Syntagmes	Contexte gauche	Contexte droit
{Je voudrais connaitre, P}		{les prévisions météorologiques, NFP} {pour le Vaucluse, GPMS}
{les prévisions météorologiques, NFP}	{Je voudrais connaitre, P}	{pour le Vaucluse, GPMS}
{pour le Vaucluse, GPMS}	{Je voudrais connaitre, P} {les prévisions météorologiques, NFP}	

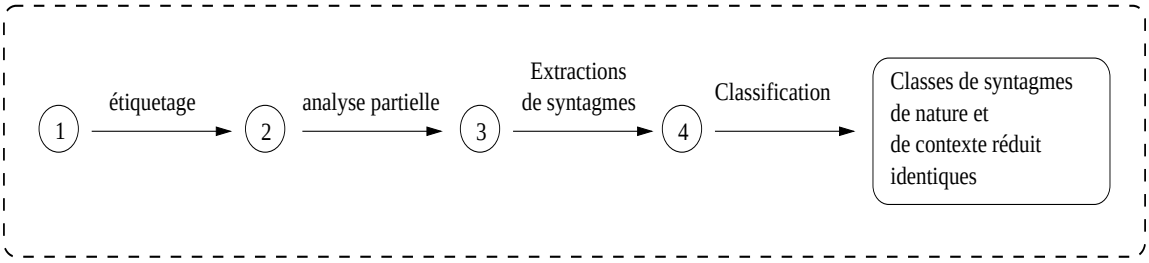


FIG. 4.5 – *Extraction des syntagmes*

4.4.1.3 Classification des séquences de mots spécifiques

À partir des séquences de mots extraites à l'étape précédente, une première classification est effectuée. Cette classification consiste à regrouper les séquences de mots de même nature grammaticale (groupes nominaux par exemple) et qui apparaissent dans des contextes identiques. Le contexte d'une séquence de mots se compose des L séquences de mots se trouvant à sa gauche et des R séquences se trouvant sur sa droite (étape 4 de la figure 4.5).

Un compteur, appelé poids de la classe, est associé à chacune des classes de séquences de mots. Pour chaque occurrence de séquence de mots appartenant à une classe et rencontrée dans le corpus d'apprentissage, le poids de cette classe est incrémenté.

Une fois toutes les classes construites, les classes dont le poids est inférieur à un seuil, appelé seuil de poids minimal et fixé empiriquement, sont éliminées.

4.4.1.4 Fusion de classes de séquences de mots

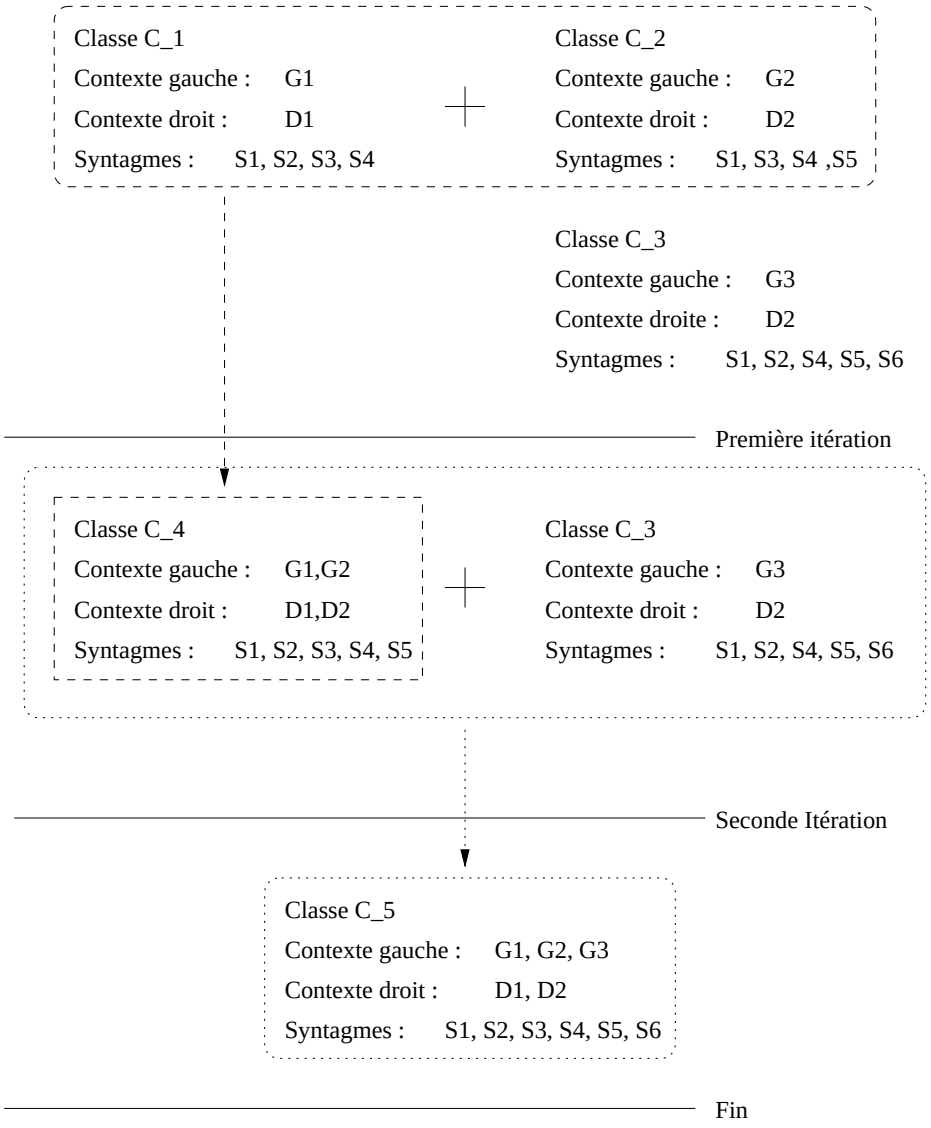


FIG. 4.6 – *Fusion de classes*

Les classes obtenues à l'étape précédente sont composées de syntagmes de même nature et rencontrés dans le même contexte. Le nombre d'occurrences d'un syn-

tagme dans une classe est stocké (ce nombre est calculé sur le corpus d'apprentissage).

Pour l'instant, aucune généralisation n'est effectuée. Celle-ci est réalisée en fusionnant les classes : deux classes proches sont fusionnées au sein d'une seule et même classe, constituée de l'union des syntagmes composant les classes d'origine. Le nombre d'occurrences d'un syntagme de la classe créée est la somme des nombres d'occurrences de ce syntagme dans chacune des classes fusionnées.

La nouvelle classe est associée aux contextes des classes initiales.

Le processus est réitéré jusqu'à ce qu'il soit impossible de trouver deux classes suffisamment proches pour les fusionner.

La distance d utilisée pour mesurer la proximité de deux classes C_i et C_j est définie ainsi :

$$d(C_i, C_j) = \sum_{S \in V} \left| N_{C_i}(S) - N_{C_j}(S) \right| \tag{4.12}$$

avec :

1. V l'ensemble des syntagmes détectés lors de l'extraction des séquences de mots,
2. S un syntagme appartenant à V,
3. $N_{C_k}(S)$ est le nombre d'occurrences du syntagme S dans la classe C_k.

Deux classes C_i et C_j sont considérées comme suffisamment proches pour être fusionnées si :

$$d(C_i, C_j) < \delta \tag{4.13}$$

où δ est le seuil de fusion.

Remarque : la mesure de distance utilisée ici est très nettement perfectible. Par exemple, en normalisant les distributions de chaque classe en fonction du nombre d'apparition d'élements de cette classe dans le corpus d'apprentissage. Nous aurions alors :

$$d(C_i, C_j) = \sum_{S \in V} \left| \frac{N_{C_i}(S)}{\sum_{S \in V} N_{C_i}(S)} - \frac{N_{C_j}(S)}{\sum_{S \in V} N_{C_j}(S)} \right| \tag{4.14}$$

Un autre point perfectible dans cette mesure de distance serait l'intégration plus ou moins importante d'informations portant sur les contextes d'apparition des classes afin d'augmenter la cohérence des classes.

Exemple La figure 4.6 illustre un exemple de fusion de trois classes. Cette fusion est effectuée en deux itérations. Tout d'abord, la classe C_1 est fusionnée avec la classe C_2 : leurs syntagmes sont réunis, ainsi que leurs contextes, dans la nouvelle classe C_4. Les classes C_1 et C_2 sont éliminées. La nouvelle classe C_4 est alors fusionnée avec la classe C_3 lors d'une nouvelle itération : ces deux classes sont éliminées pour créer la classe C_5. Seules les classes suffisamment proches fusionnent.

4.4.1.5 Mise sous forme d'automates stochastiques à états finis

Chaque classe contient un ensemble de séquences de mots. Chaque séquence est associée à son nombre d'occurrences dans cette classe.

La mise sous forme d'automates consiste à associer chaque syntagme d'une classe à sa probabilité d'apparition dans cette classe, puis à changer la représentation de cette classe (voir la figure 4.1, pour laquelle les probabilités de chaque chemin ont été omises dans un souci de clarté). La probabilité P d'apparition d'un syntagme S_i au sein d'une classe de séquences de mots C est donnée par la formule suivante :

$$\boxed{P(S_i|C) = \frac{N_C(S_i)}{\sum_{S_j \in C} N_C(S_j)}} \tag{4.15}$$

où $N_C(S)$ est le nombre d'occurrences du syntagme S dans la classe C.

La figure 4.7 résume le processus d'apprentissage des modèles *internes*, représentés sous la forme d'automates stochastiques à états finis.

4.4.2 Modèle *externe*

Le modèle *externe*, qui modélise le comportement des différentes entités du modèle hybride (mots et automates), n'est autre qu'un modèle de langage *n-gram* classique. Cependant, celui-ci ne peut être estimé qu'après modification du corpus d'apprentissage. Les résultats de l'analyse grammaticale partielle des phrases du corpus d'apprentissage sont également utilisés.

Cette modification consiste à remplacer les syntagmes appartenant à une classe par le nom de l'automate qui représente cette classe. Dans la méthode de création des automates présentée ici, il faut noter deux caractéristiques importantes :

1. Un syntagme peut appartenir plusieurs classes
2. Un syntagme, pour un contexte donné, et une analyse grammaticale partielle fixée, ne peut appartenir à plus d'une classe.

Ainsi, le remplacement éventuel d'un syntagme par le nom d'un automate est déterminé sans aucune difficulté par le contexte de ce syntagme et l'analyse grammaticale partielle du corpus d'apprentissage : soit aucune classe contenant ce syntagme n'est associée à ce contexte et le syntagme n'est pas remplacé, soit il en existe une

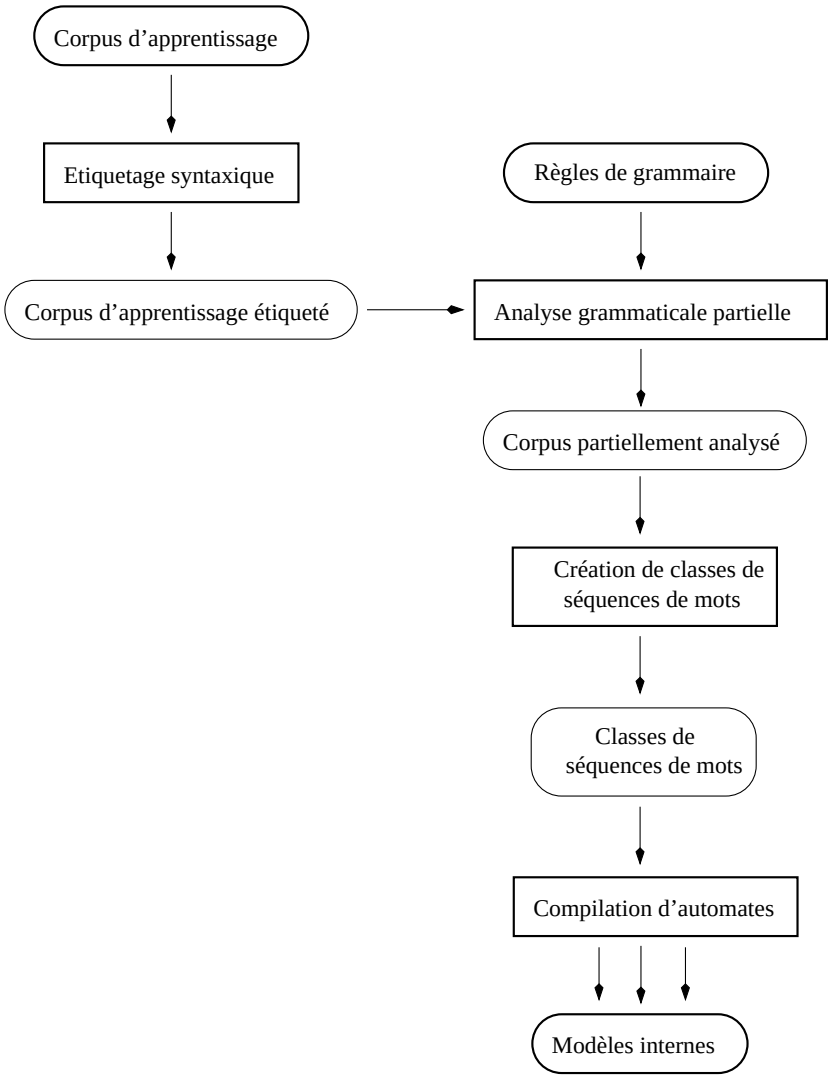


FIG. 4.7 – *Processus détaillé d'apprentissage des modèles internes*

et le syntagme est remplacé. Il ne peut y avoir d'ambiguïté car l'analyseur grammatical ne propose qu'une seule segmentation en syntagme pour une phrase donnée.

Le nouveau corpus d'apprentissage est alors utilisé pour estimer le modèle *externe*, le lexique associé au modèle étant composé de mots et de noms d'automates.

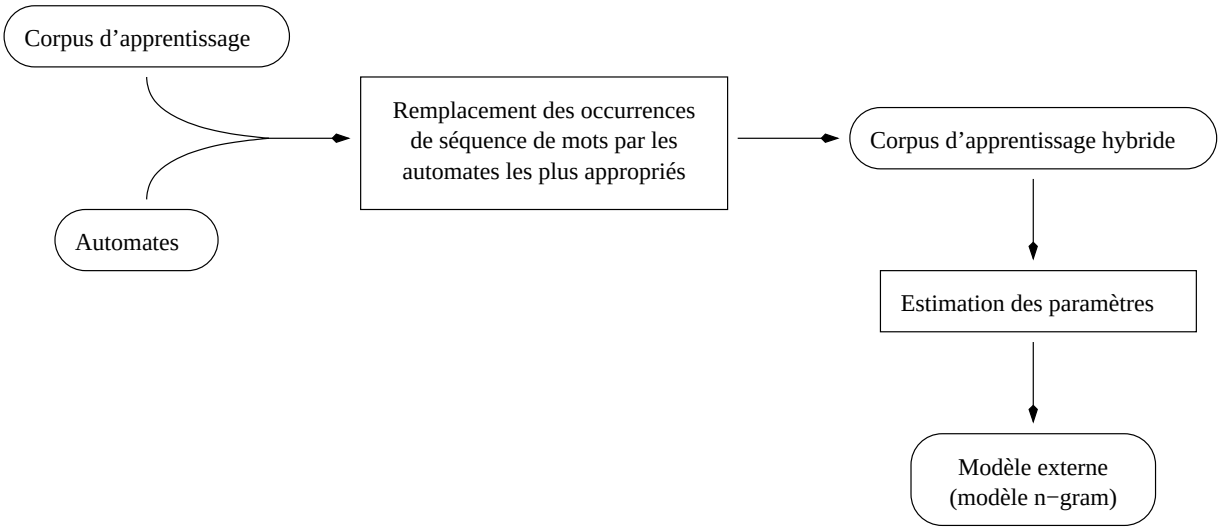


FIG. 4.8 – *Apprentissage du modèle externe*

4.5 Utilisation du modèle hybride dans un système de reconnaissance de la parole

4.5.1 Utilisation du modèle hybride

Nous proposons d'utiliser en deuxième passe d'un système de reconnaissance le modèle hybride présenté dans ce mémoire : nous présentons une stratégie de recherche basée sur un graphe de mots issu d'une première passe[7].

Le graphe de mots est constitué de mots appartenant au lexique utilisé lors du décodage acoustique. Comme les classes de séquences de mots ne sont pas gérées dans ce décodage (le niveau le plus haut lors de la première passe étant le mot, et non la séquence de mots), il faut détecter dans le graphe les séquences de mots appartenant à une classe.

La représentation sous forme d'automates à états finis de ces classes facilite la détection : pour chaque noeud du graphe, ces automates sont utilisés comme automates accepteurs qui valident ou refusent les séquences de mots qui commencent sur ce noeud.

4.5.2 Algorithme de détection d'une séquence de mots spécifique

Un état d'un automate à états finis est dit *terminal* s'il est le dernier état d'un parcours de l'automate. Ce parcours, qui débute à l'état initial, décrit une séquence de mots appartenant à la classe de séquences que représente cet automate.

[7] Voir section 3.2.

Un état est dit *semi-terminal* quand le chemin qui mène à cet état représente une des séquences de mots reconnues par l'automate. A la différence d'un état *terminal*, un état *semi-terminal* n'est pas le dernier état d'un chemin : ce chemin peut continuer pour représenter une ou plusieurs séquences de mots plus grandes qui appartiennent également à l'automate. Par exemple, la séquence de mots "le serveur météo" peut être décrite par un chemin passant par l'état *semi-terminal* auquel aboutit la transition associée au mot "météo". Le chemin qui décrit la séquence de mots "le serveur météo agricole" passera par ce même état *semi-terminal* pour aboutir à l'état suivant la transition associée au mot "agricole".

Un état est dit *transitoire* s'il n'est ni *terminal*, ni *semi-terminal*.

Pour un noeud N_m du graphe de mots G et un automate A, l'algorithme de détection d'une séquence de mots acceptée par A et commençant en N_m est le suivant :

1. Initialisation : l'état s_l est initialisé comme étant l'état initial s_0 de l'automate A, donc $s_l = s_0$ à l'initialisation. Le noeud N_k est initialisé comme étant le noeud N_m. Comme il peut exister X transitions entre deux noeuds N_k et N_m avec $X \geq 1$, on notera chacune de ces transitions $t_{x,k\to m}(G)$, avec $1 \leq x \leq X$. On notera par contre $t_{f\to g}(A)$ la transition, unique par construction, qui lie l'état s_f de l'automate A à l'état s_g.
2. Pour chaque transition $t_{x,k\to k+i}(G)$ partant du noeud N_k vers les noeuds N_{k+i} dans le graphe G, on teste l'égalité entre le mot $w_{x,k\to k+i}(G)$ associée à cette transition et les mots $w_{l\to l+j}(A)$ associés aux transitions $t_{l\to l+j}(A)$ de l'automate A qui partent de l'état s_l vers les états s_{l+j}. Pour chaque égalité, le processus continue en **3**, pour les inégalités, le processus concernant ce parcours de l'automate s'achève sur un échec.
3. Si s_{i+j} est un état *terminal*, la séquence de mots reconnue par l'automate (qui correspond au chemin qui a permis d'arriver à l'état final s_{i+j} en partant de l'état initial s_0) est acceptée et le processus s'achève pour cette branche de l'automate. Si s_{i+j} est un état *semi-terminal*, la séquence de mots reconnue jusqu'alors par l'automate est acceptée, et le processus continue en **4**. Si s_{i+j} est un état *transitoire*, le processus continue en **4**.
4. L'état s_{i+j} est noté s_i, le noeud N_{k+i} est noté N_k. Le processus continue en **2**.

Cet algorithme est appliqué à chaque noeud du graphe G, pour chaque automate associé au modèle de langage. Pour accélérer le processus, les différents automates sont unifiés pour ne former qu'un seul automate, plus grand, mais qui évite de parcourir plusieurs fois les mêmes chemins partiels du graphe. Les états terminaux de l'automate unifié contiennent l'information liant la séquence de mots reconnue aux classes auxquelles elle appartient.

4.5.3 Séquence de mots détectée : insertion d'une transition

Pour chaque séquence de mots S appartenant à une classe C et détectée dans le graphe, le noeud de départ N_d dans le graphe, le noeud d'arrivée N_f et le chemin parcouru $c_{d\to f}$ sont connus. Pour chaque classe associée à la séquence de mots

reconnue, une transition est ajoutée au graphe : elle part de N_d pour finir sur N_f. et porte le nom de la classe et le score acoustique du chemin $c_{d \to f}$ (égal au produit des probabilités des scores acoustiques des mots qu'il représente). De plus, la probabilité d'apparition $P(S|C)$ de la séquence S dans la classe C est associée à la nouvelle transition : cette probabilité est donnée par l'état terminal (ou semi-terminal) liée à cette séquence de mots dans l'automate $A(C)$.

L'insertion d'une transition dans le graphe n'est pas l'ajout d'une hypothèse : il s'agit seulement d'une représentation différente qui permet d'utiliser correctement le modèle de langage hybride. Ainsi, une séquence de mots du graphe appartenant à plusieurs classes sera représentée par plusieurs transitions en concurrence, mais il s'agira toujours de la même hypothèse de séquence de mots, avec le même score acoustique.

Le graphe, une fois enrichi de nouvelles transitions, peut être traité avec les algorithmes habituels (Cettolo *et al.*, 1998) afin de trouver la phrase de probabilité maximale.

4.6 Expérimentations

Des expériences ont été menées pour comparer le modèle hybride présenté ici à un modèle *n-gram* classique. Ces expériences ont pu être effectuées grâce à France Télécom Recherche et Développement qui a fourni les données. Celles-ci concernent une application de dialogue homme-machine par téléphone : il s'agit du démonstrateur Audiotel Guide des Services (AGS), décrit dans (Sadek *et al.*, 1996). Ces données, constituées de données d'apprentissage et de test, ont permis d'estimer les modèles de langage, et d'évaluer leur perplexité sur différents corpora ou leur influence dans un système de reconnaissance de la parole.

Le démonstrateur AGS est utilisé afin de fournir à un utilisateur humain des numéros de téléphone de serveurs vocaux spécialisés dans les prévisions météorologiques ou la recherche d'emploi. Le dialogue qui s'établit par téléphone entre le démonstrateur et l'utilisateur humain a pour but de guider l'utilisateur vers le serveur le plus pertinent vis-à-vis de sa demande de renseignements.

Pour cette application, le lexique est modeste : il comprend 880 mots.

4.6.1 Description des données expérimentales

4.6.1.1 Données d'apprentissage

Les données d'apprentissage se présentent sous la forme d'un corpus de transcriptions de phrases prononcées par des utilisateurs du démonstrateur AGS. Il ne s'agit pas d'un grand corpus, puisqu'il est composé de $9\,842$ phrases, pour $49\,591$ mots, dont 821 différents. Ces phrases ont été récupérées à partir d'une collecte de données effectuées à l'aide de locuteurs naïfs et de locuteur experts. Les locuteurs naïfs sont des personnes ne travaillant pas pour France Télécom R&D et n'ayant pas de

connaissances en reconnaissance de la parole. Les locuteurs experts travaillent pour France Télécom R&D. Les 821 mots du corpus d'apprentissage font partie des 880 mots du lexique du démonstrateur AGS. Plus de détails sur l'acquisition des corpora de test et d'apprentissage sont donnés dans (Damnati, 2000).

Les phrases du corpus d'apprentissage sont des questions, des requêtes, des réponses, ou des commandes ("annulation", par exemple). Elles concernent toutes l'application AGS. Une étude plus précise de ces phrases permet de noter qu'une grande partie d'entre elles (59%) sont des phrases courtes (1 à 4 mots). La figure 4.9 montre la répartition des phrases en fonction de leur nombre de mots.

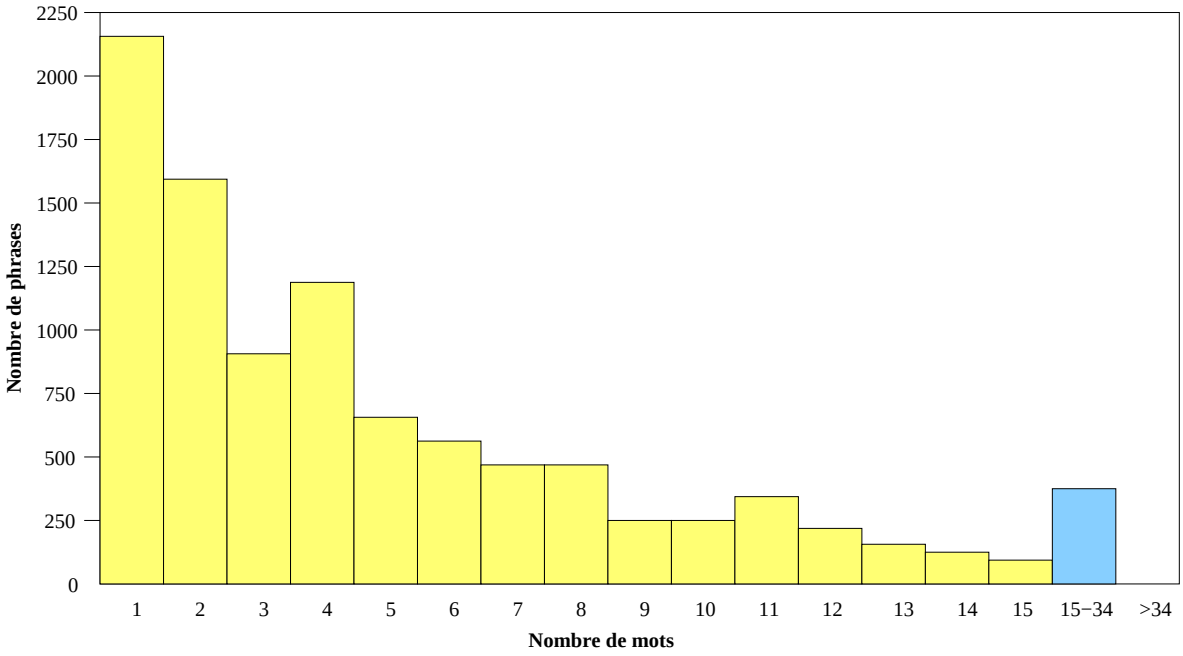


FIG. 4.9 – *Répartition des phrases du corpus d'apprentissage en fonction du nombre de mots qui les composent*

4.6.1.2 Données de test

Les données de test sont des graphes de mots issus du processus de reconnaissance de la parole du démonstrateur AGS. Chacun de ces graphes de mots est associé à une phrase, appelée phrase de référence, qui correspond à la phrase effectivement prononcée par le locuteur. Les scores acoustiques associés aux mots dans un graphe sont calculés lors de la génération du graphe par le module de reconnaissance de la parole du démonstrateur AGS.

Deux graphes sont disponibles pour chaque phrase de référence : ils diffèrent par la taille de l'espace de recherche qu'ils représentent. Cette taille dépend de la valeur du paramètre d'élagage. Pour les expériences, nous avons donc regroupé les graphes en deux ensembles :

1. l'ensemble nommé Graphes I, constitué de graphes fortement élagués,
2. un ensemble nommé Graphes II, constitué de graphes de mots représentant un espace de recherche plus grand (élagage plus faible).

Les phrases de référence sont au nombre de $1\,422$, composés de $7\,014$ mots, dont 504 mots différents. La nature et la longueur de ces phrases sont semblables aux phrases du corpus d'apprentissage : la figure 4.10 illustre la répartition des phrases de référence en fonction de leur nombre de mots.

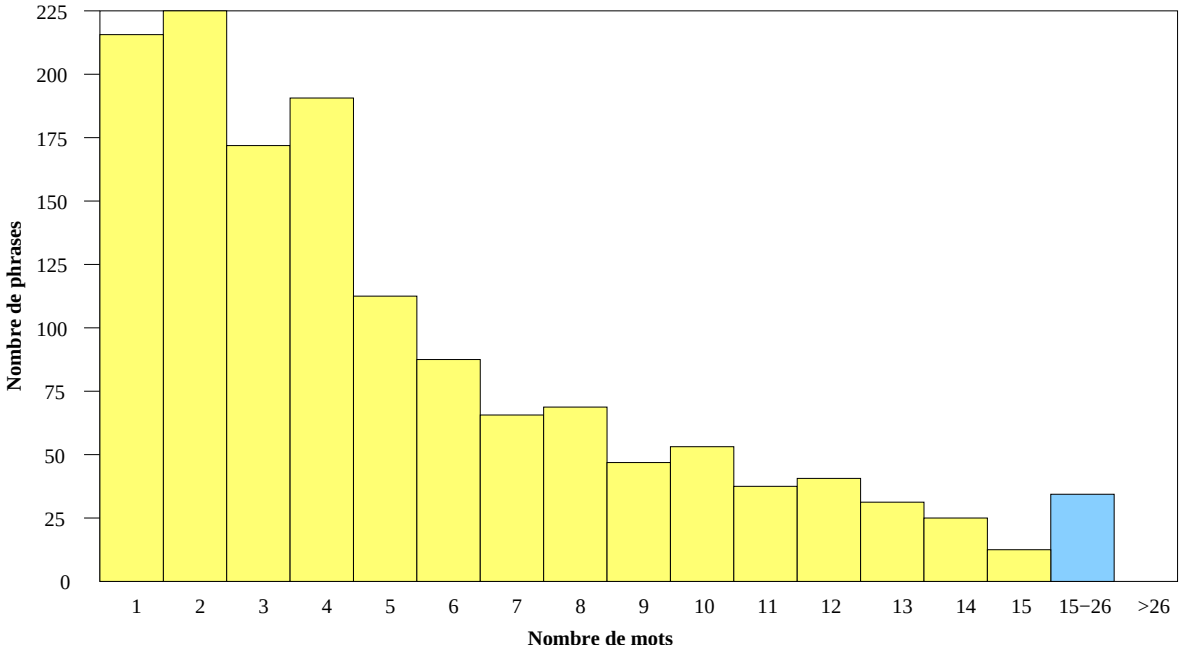


FIG. 4.10 – *Répartition des phrases de référence du corpus de test en fonction du nombre de mots qui les composent*

Il est intéressant de noter que sur les 504 mots différents des phrases de référence du corpus de test, 109 mots n'apparaissent pas dans le corpus d'apprentissage. Certains de ces mots n'appartiennent pas au lexique : ce sont des mots dits hors-vocabulaire. Le tableau 4.1 montrent que ces 109 mots, composés de 45 mots non vus et 64 mots hors-vocabulaire, affectent 187 phrases du corpus de test, soit $13,15\%$ des phrases de référence (aucune phrase du corpus de test ne contient à la fois un mot hors-vocabulaire et un mot non vu). Pour gérer les mots hors-vocabulaire, une entrée lexicale notée <UNK> représentant les mots inconnus est ajoutée au lexique. Au niveau de la modélisation du langage, les événements non vus sont gérés par les techniques de lissage[8].

En dehors du problème des mots non vus qui affectent les performances des modèles de langage et qui a donc une incidence sur les performances globales d'un système de reconnaissance, d'autres facteurs peuvent intervenir. Le décodage acoustique, qui génère les graphes de mots, peut connaître quelques difficultés. Dans le

[8]voir la section 1.3, consacrée au lissage

Nombre de phrases du corpus de test	1422
Nombre de phrases avec mot(s) hors-vocabulaire	120 (8,44 %)
Nombre de phrases avec mot(s) non vu(s)	67 (4,71 %)

TAB. 4.1 – *Phrases de test contenant au moins un mot hors-vocabulaire ou un mot non vu lors de l'apprentissage*

cas du démonstrateur AGS, les conditions d'acquisition de la parole sont difficiles : utilisation du téléphone, environnements sonores différents et bruités, locuteurs différents, ... Ces conditions, associées à un lexique fermé de 880 mots, et à un élagage plus ou moins fort de l'espace de recherche, compliquent la production de graphes de mots contenant des hypothèses acoustiquement fiables. Ainsi, comme le montre le tableau 4.2, pour environ 30% des graphes fortement élagués (ensemble de graphes appelés Graphes I) et environ $24,5\%$ des graphes ayant subi un élagage plus faible, la phrase de référence n'est pas présente dans les graphes. Dans ce cas, il est impossible de retrouver la phrase prononcée par le locuteur à partir du graphe de mots : les hypothèses issues du processus de reconnaissance seront forcément erronées.

	Graphes ne contenant pas la référence	Graphes contenant la référence
Graphes I	426 (29,96%)	996 (70,04%)
Graphes II	350 (24,61%)	1072 (75,39%)

TAB. 4.2 – *Répartition des graphes du corpus de test en fonction de la présence ou de l'absence de la phrase de référence dans ces graphes*

Les phrases du corpus de test peuvent être regroupées en fonction du locuteur qui les a prononcées. Il existe six locuteurs identifiés (l_1, l_2, l_3, l_4, l_5 et l_6), et un panel de locuteurs anonymes. Ce panel est nommé p_0. Le tableau 4.3 montre le nombre de phrases prononcées par chaque locuteur, ainsi que le nombre de sessions de dialogue correspondantes. Une session de dialogue correspond à un appel du locuteur et à l'intégralité du dialogue associé à cet appel.

locuteur	nombre de sessions	nombre de phrases
l_1	74	574
l_2	13	166
l_3	9	91
l_4	12	122
l_5	14	136
l_6	15	209
p_0	25	124

TAB. 4.3 – *Répartition des mots, des phrases et des sessions du corpus de test en fonction du locuteur*

4.6.2 Evaluation des modèles de langage

Avant de comparer les performances d'un système de reconnaissance de la parole utilisant le modèle hybride présenté ici à un système basé sur un modèle *n-gram* classique, les valeurs de perplexité de ces modèles seront d'abord confrontées afin de mesurer la qualité intrinsèque de chacun de ces modèles.

Les modèles étudiés sont des modèles de langage de *bigrams* et *trigrams* puisqu'il s'agit des modèles les plus utilisés. Le modèle hybride construit à partir d'un modèle *bigram* classique est noté *bigram+automates*. De la même manière, le modèle hybride construit à partir d'un modèle *trigram* est noté *trigram+automates*.

4.6.2.1 Perplexité

La figure 4.11 montre la valeur des perplexités des différents modèles : un modèle de langage *bigram* classique, un modèle hybride de type *bigram*, un modèle *trigram* classique, et un modèle hybride de type *trigram*. Ces modèles de langage ont été estimés sur le corpus d'apprentissage décrit précédemment. Les perplexités illustrées par la figure 4.11 ont été calculées sur le corpus d'apprentissage et sur le corpus de test.

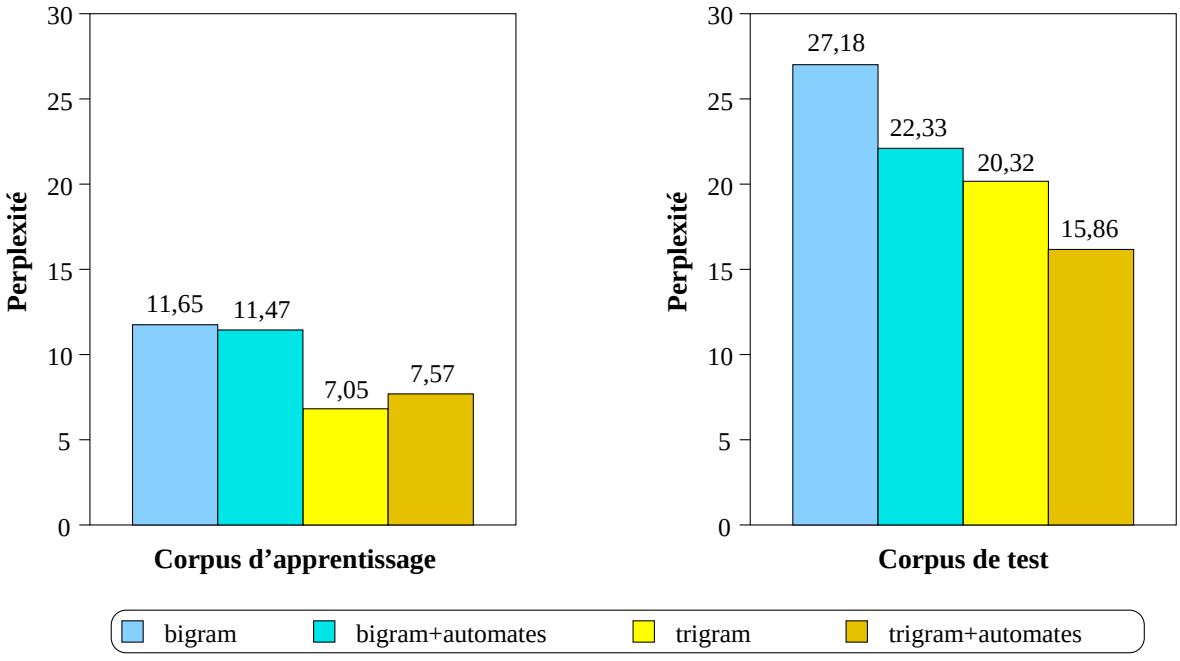


FIG. 4.11 – *Comparaison des perplexités des différents modèles n-grams et modèles hybrides sur les corpus d'apprentissage et de test*

Les résultats concernant les valeurs des perplexités sur le corpus d'apprentissage permettent de distinguer deux comportements généraux : le comportement des modèles à base de *bigrams*, et celui des modèles à base de *trigrams*. La valeur de la

perplexité des modèles *trigrams* est de 35 à 40% inférieure à la valeur de la perplexité des modèles *bigrams*, ce qui confirme la supériorité en terme de modélisation des modèles *trigrams* sur les modèles *bigrams*.

En comparant les modèles hybrides aux modèles classiques du même ordre, toujours sur le corpus d'apprentissage, la valeur de perplexité des modèles avec automates est soit semblable, soit très légèrement supérieure. Ce phénomène s'explique par la généralisation effectuée lors de l'apprentissage de ces modèles, et plus précisément lors du processus de fusion des classes de séquences de mots : les modèles hybrides sont moins proches du corpus d'apprentissage que les modèles classiques.

Cette capacité de généralisation, combinée à une modélisation intégrant des contraintes de taille supérieure à deux ou trois mots, permet aux modèles hybrides d'obtenir de très bons résultats en terme de perplexité sur le corpus de test. Le modèle *bigram* avec automates obtient une perplexité inférieure de 17,8% à celle du modèle *bigram* classique, alors que le modèle hybride *trigram* améliore de 21,9% le résultat du modèle *trigram*. Il est également remarquable que la valeur de perplexité du modèle *trigram* classique est inférieure de seulement 9% à celle du modèle *bigram* avec automates.

En terme de perplexité, les modèles à base d'automates sont donc plus performants que les modèles *n-grams* classiques. Une amélioration en terme de perplexité ne se traduit pas toujours en une amélioration des performances de reconnaissance de la parole, même si il existe souvent une corrélation. Cependant, vu le gain substantiel en termes de perplexité obtenu par les modèles hybrides sur les modèles *n-grams* classiques, il est envisageable d'attendre de l'intégration de ces modèles dans un système de reconnaissance de la parole une amélioration de ses performances par rapport à l'utilisation de modèles *n-grams* classiques.

Le fait que les modèles *trigrams* soient plus performants sur le corpus de test est un indicateur de la faible variabilité des phrases de l'application. En effet, le corpus d'apprentissage est de taille très modeste mais permet l'estimation d'un modèle *trigram* suffisamment robuste pour obtenir une perplexité sur le corpus de test inférieure à celle d'un modèle *bigram* estimé sur le même corpus d'apprentissage.

4.6.2.2 Reconnaissance de la parole

Résultats de reconnaissance Les expériences de reconnaissance de la parole ont été effectuées à partir des graphes de mots générés par le module de reconnaissance du démonstrateur AGS. Les valeurs du *fudge factor* et de la pénalité linguistique, optimisés pour les modèles *bigram* et *trigram* classiques sont constants : les mêmes valeurs sont utilisées pour tous les modèles. Le seul facteur variant dans nos expériences est le modèle de langage utilisé afin d'obtenir la phrase de probabilité maximum pour chacun des graphes. La valeur du *fudge factor* retenu est 7 : cette valeur donne les meilleurs résultats en terme de taux d'erreurs sur les mots pour les modèles *bigram* et *trigram* classiques considérés comme bases de référence pour nos expériences. La figure 4.12 montre le taux d'erreurs sur les mots obtenu en fonction de la valeur du *fudge factor* retenu en utilisant le modèle

bigram classique sur les graphes de type I, avec une pénalité linguistique nulle. Avec le modèle *trigram* classique, la courbe conserve le même comportement avec un taux d'erreurs minimum lorsque le *fudge factor* est égal à 7.

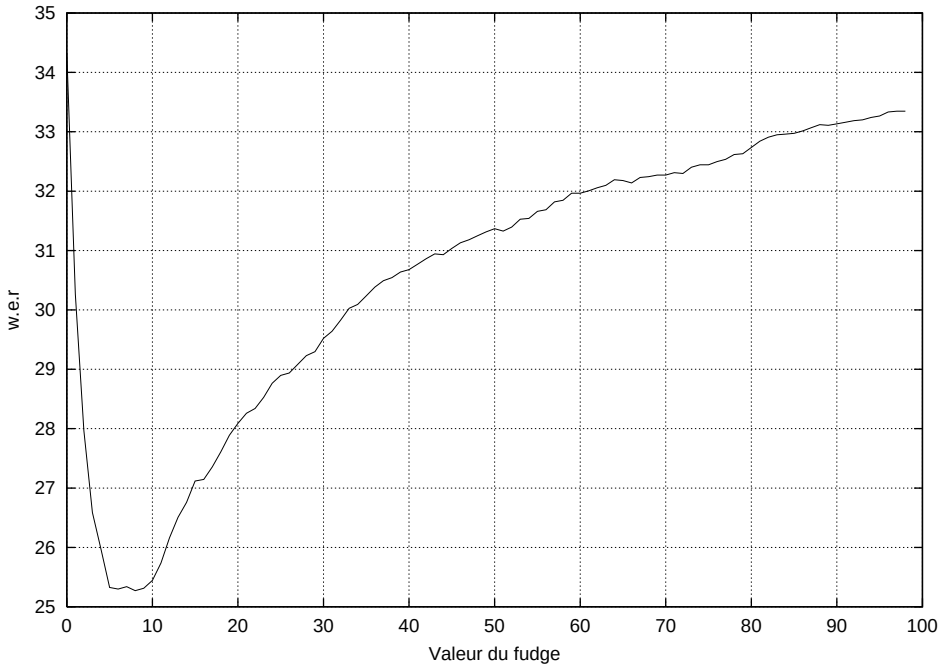


FIG. 4.12 – *Évolution du taux d'erreur sur les mots en fonction de la valeur du fudge factor (Modèle bigram, Graphes I)*

Les expériences ont été menées sur les deux ensembles de graphes : l'ensemble nommé Graphes I pour les graphes fortement élagués, et l'ensemble Graphes II pour les graphes ayant subi un élagage moins important. Il est à noter que les mêmes paramètres d'élagage ont été utilisés pour tous les graphes d'un même ensemble.

La figure 4.13 montre les résultats, en terme de taux d'erreurs sur les mots, obtenus par le système de reconnaissance selon le modèle de langage utilisé pour chacun des ensembles de graphes.

Une première constatation concerne la différence de résultats entre les graphes de l'ensemble I et les graphes de l'ensemble II : pour chaque modèle, le taux d'erreurs sur les mots est plus faible sur les graphes II que sur les graphes I. Comme les graphes II sont moins élagués que les graphes I, il y a moins de risque que des hypothèses viables soient absentes de ces graphes, ce que montre le tableau 4.2. Bien entendu, cette amélioration des résultats de reconnaissance de la parole a un coût en terme de temps de calcul, et il peut être intéressant de sacrifier la précision de la reconnaissance au profit de la vitesse de réponse du système de dialogue.

Dans le cas d'une application de dialogue oral homme-machine, le traitement en temps réel est une nécessité. Cependant, au-delà du nombre de calculs à effectuer, il faut prendre en compte la technologie disponible : la vitesse de traitement

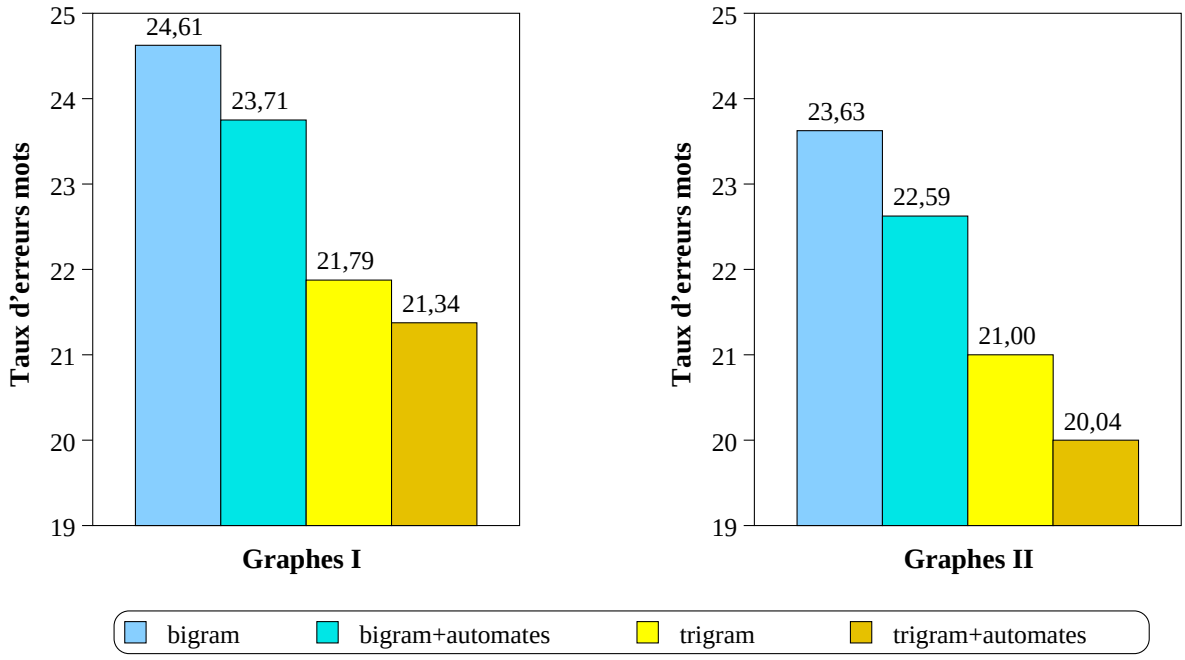


FIG. 4.13 – *Comparaison des taux d'erreurs sur les mots des différents modèles n-grams et modèles hybrides sur les deux ensembles de graphes*

des ordinateurs augmentant en permanence grâce aux progrès de l'industrie électronique, un calcul qui nécessite dix secondes aujourd'hui ne demandera qu'une fraction de seconde dans un futur à moyen terme. Il faut toutefois être vigilant sur la complexité des algorithmes utilisés, une complexité exponentielle n'étant pas satisfaisante quelle que soit la vitesse des ordinateurs, surtout lorsque la taille des données traitées est susceptible de s'accroître.

Les résultats illustrés par la figure 4.13 montrent que les modèles hybrides ont tendance à être plus performants, en terme de taux d'erreurs sur les mots, que les modèles *n-grams* de même ordre. La plus faible réduction relative du taux d'erreurs sur les mots apparaît entre les modèles de type *trigram* sur les graphes I : le modèle *trigram* avec automates permet d'obtenir une baisse relative du taux d'erreur de $2,06\%$ par rapport au taux d'erreurs sur les mots obtenus sur le modèle *trigram* classique. Par contre, entre ces deux modèles, la diminution relative du taux d'erreurs atteint $4,57\%$ sur les graphes II. Ceci peut s'expliquer par la plus grande quantité d'hypothèses contenues dans les graphes II : certaines hypothèses qui semblaient peu probables au niveau local au moment de l'élagage des graphes I, qui utilise un modèle *bigram* pour choisir les chemins partiels à supprimer, peuvent en fait être préférées à juste titre par le modèle hybride dans les graphes II où elles n'ont pas été éliminées.

Cette diminution relative du taux d'erreurs est également effective pour les modèles *bigrams* où elle atteint $3,66\%$ avec les graphes I et $4,4\%$ avec les graphes II.

Pour les graphes I, les diminutions des taux d'erreurs induites par l'utilisation

d'un modèle hybride ne sont pas significatives : en effet, pour le modèle hybride de type *bigram*, l'intervalle de confiance à 95% de son taux d'erreurs observé lors de l'expérience intègre le taux d'erreurs observé pour le modèle *bigram* classique, comme le montre le tableau 4.4.

Taux d'erreurs (%)	Intervalle de confiance à 95%	Commentaires
24,61	[23,60 ; 25,62]	*bigram*, Graphes
23,71	[22,71 ; 24,70]	*bigram+auto*, Graphes I
21,79	[20,82 ; 22, 76]	*trigram*, Graphes I
21,34	[20,38 ; 22, 30]	*trigram+auto*, Graphes I
23,63	[22,64 ; 24,62]	*bigram*, Graphes II
22,59	[21,61 ; 23,57]	*bigram+auto*, Graphes II
21,00	[20,05 ; 21, 95]	*trigram*, Graphes II
20,04	[19,10 ; 20,98]	*trigram+auto*, Graphes II

TAB. 4.4 – *Intervalles de confiance à 95% des différents taux d'erreurs observés sur le corpus de test*

Par contre, pour les graphes II, la diminution du taux d'erreurs engendrée par l'intégration des automates est statistiquement fiable à 95%. Il faut toutefois pondérer ces résultats qui peuvent être biaisés par les arrondis effectués à la seconde décimale[9]. Cependant, il est indéniable qu'il existe une tendance forte des modèles hybrides à améliorer les résultats de reconnaissance des modèles *n-grams* classiques de même ordre.

Résultats de reconnaissance selon la qualité des graphes de mots Comme le montre le tableau 4.2, presque 30% des graphes de l'ensemble I et presque 25% des graphes de l'ensemble II ne proposent pas de solution permettant de reconnaître la phrase qui a été prononcée par l'utilisateur.

Les tableaux 4.5 et 4.6 montrent les résultats, en terme de taux d'erreurs sur les mots, des différents modèles de langages selon la présence ou non de l'hypothèse correcte au sein des graphes. Il apparaît très clairement que lorsque la phrase à reconnaître est présente dans le graphe, les résultats de reconnaissance sont très bons. Alors que sur l'ensemble des graphes le taux d'erreurs sur les mots oscille entre 20 et 25%, il évolue, pour les graphes contenant l'hypothèse correcte, entre 6 et 12%, bien que ces graphes constituent au moins les deux tiers de l'ensemble des graphes. Logiquement, les résultats de reconnaissance à partir des graphes ne contenant pas l'hypothèse correcte sont très mauvais, et dans ces conditions le taux d'erreurs est compris entre 45 et 48%.

Les taux d'erreurs sur les graphes II contenant la phrase de référence sont plus élevés que les taux d'erreurs obtenus en utilisant les graphes I ayant la même caractéristique. Il ne faut pas en conclure que la reconnaissance sur les graphes les plus

[9] Utiliser plus de deux décimales pour obtenir une meilleure précision n'est pas une bonne solution étant donné le nombre d'échantillons utilisés pour les expériences.

Graphes I	*bigram*	*bigram+auto*	*trigram*	*trigram+auto*
avec référence	9,66	8,61	6,97	6,7
sans référence	46,86	46,17	43,85	43,13

TAB. 4.5 – *Taux d'erreurs sur les mots en fonction de la présence ou non de la phrase de référence dans les graphes pour l'ensemble des graphes I*

Graphes II	*bigram*	*bigram+auto*	*trigram*	*trigram+auto*
avec référence	11,90	10,4	8,62	7,8
sans référence	47,65	47,57	46,36	45,10

TAB. 4.6 – *Taux d'erreurs sur les mots en fonction de la présence ou non de la phrase de référence dans les graphes pour l'ensemble des graphes II*

élagués donne de meilleurs résultats que sur des graphes moins élagués lorsque l'hypothèse correcte est présente : ces deux ensembles de graphes (les graphes I contenant la phrase de référence et les graphes II contenant la phrase de référence) ne peuvent pas être comparés. En effet, les graphes I contenant l'hypothèse correcte sont moins nombreux que leurs homologues de l'ensemble II. En réalité, un graphe II qui contient la phrase de référence alors que le graphe I associé au même signal de parole ne la contient pas, est certainement un graphe contenant de l'information peu fiable. Il correspond certainement à une phrase prononcée très peu observée dans le corpus d'apprentissage, ou bien à une phrase émise dans un contexte acoustique difficile : on peut penser que si le processus d'élagage a supprimé la bonne hypothèse, c'est que celle-ci avait une probabilité acoustique ou linguistique relativement faible. Dès lors, même si cette hypothèse est présente dans le graphe II, elle sera difficilement choisie comme hypothèse finale de reconnaissance.

Pour affiner cette étude, nous pouvons diviser le corpus de test en trois parties :

1. une partie notée $(GI+, GII+)$ dans laquelle les graphes I et les graphes II contiennent la phrase de référence (994 cas sur 1422).
2. une partie notée $(GI-, GII+)$ dans laquelle les graphes I ne contiennent pas la phrase de référence alors que les graphes II la contiennent (80 cas sur 1422). Dans notre corpus de test, le contraire n'est jamais observé : lorsqu'une phrase de référence est absente d'un graphe II, elle n'existe pas non plus dans le graphe I correspondant.
3. une partie notée $(GI-, GII-)$ dans laquelles ni les graphes I ni les graphes II ne contiennent la référence (348 cas sur 1422).

Les tableaux 4.7 et 4.8 montrent les résultats en terme de taux d'erreurs sur les mots obtenus avec les modèles de langage classiques et avec automates sur les graphes I et les graphes II regroupés selon le découpage présenté ci-dessus.

Dans tous les cas, les modèles de langage *n-grams* à automates donnent de meilleurs résultats que les modèles de langage *n-grams* classiques, sauf avec les modèles

	2g GI	2g+auto GI	2g GII	2g+auto GII
$(GI+, GII+)$	9,51	8,56	9,59	8,74
$(GI-, GII+)$	38,64	37,59	**31,12**	**25,17** ***(-19,1%)***
$(GI-, GII-)$	48,9	46,78	47,52	47,93

TAB. 4.7 – *Comparaison du taux d'erreurs obtenu à l'aide de modèles de langage bigrams classiques ou avec automates en fonction de l'existence ou de l'absence de la phrase de référence dans les graphes I et II*

	3g GI	3g+auto GI	3g GII	3g+auto GII
$(GI+, GII+)$	6,85	6,38	6,91	6,47
$(GI-, GII+)$	32,17	30,59	**22,9**	**17,65** ***(-22,93%)***
$(GI-, GII-)$	46,67	46,34	46,26	45,21

TAB. 4.8 – *Comparaison du taux d'erreurs obtenu à l'aide de modèles de langage trigrams classiques ou avec automates en fonction de l'existence ou de l'absence de la phrase de référence dans les graphes I et II*

bigrams lorsque la phrase de référence n'existe dans aucun type de graphe : la différence de taux d'erreurs est alors très faible.

Il faut en particulier remarquer la réduction très importante du taux d'erreurs obtenu avec les modèles à automates sur les graphes II lorsque ces graphes contiennent la phrase de référence alors que les graphes I correspondant ne la contiennent pas. Même si cela ne concerne que 80 phrases sur les 1422 du corpus de test, le calcul de l'intervalle de confiance à 95% montre qu'il s'agit tout de même d'une réduction statistiquement significative (par exemple, pour le taux d'erreur $(25,17\%)$ du modèle *bigram* à automates sur les graphes II dans le cas $(GI-, GII+)$ l'intervalle de confiance est $[21,62\ ;\ 28,73]$).

Le fait qu'une phrase de référence n'existe pas dans un graphe I semble indiquer que si elle existe dans le graphe II correspondant un ou plusieurs mot qui la composent sont pénalisés par leur score acoustique (ceci peut être dû par exemple à la présence de bruit lors de l'enregistrement). Il semble donc que les modèles à automates arrivent plus facilement que les modèles *n-grams* classiques à favoriser des hypothèses viables qui n'ont pas un score acoustique très favorable. Il serait intéressant de posséder plus de données de ce type afin d'établir une étude plus pointue de ce phénomène.

4.6.2.3 Bilan

Dans tous les cas de figure, les modèles hybrides permettent d'obtenir un meilleur taux d'erreurs que leurs homologues *n-grams* classiques. Les améliorations les plus intéressantes concernent les diminutions des taux d'erreurs obtenus sur les graphes contenant l'hypothèse correcte : ici, la diminution relative du taux d'erreurs les modèles classiques et hybrides oscille entre $3,87\%$ et $12,6\%$, voire $22,93\%$

dans certaines circonstance très particulières. Certaines des diminutions du taux d'erreurs obtenues lors de ces dernières expériences sont statistiquement significatives, en particulier les résultats concernant les graphes II. La diminution du taux d'erreurs issue des expériences à base de modèles *trigrams* classiques et hybrides sur les graphes I n'est pas statistiquement significative, mais suit tout de même la tendance générale : les résultats des expériences effectuées sur le corpus AGS montrent que le modèle *n-gram* intégrant des automates stochastiques a un effet positif sur les performances du système de reconnaissance de la parole.

Chapitre 5

Sélection dynamique de modèles de langage

Sommaire

5.1 Motivations 91
5.1.1 Travaux existants 91
5.1.2 Particularités de l'étude et propositions 92
5.2 Construction de modèles spécifiques 93
5.2.1 Étiquetage des phrases à l'aide de connaissances *a priori* . . . 94
5.2.2 Utilisation d'un arbre de classification sémantique 95
5.3 Sélection des modèles 102
5.3.1 Modèles construits par étiquetage des phrases 102
5.3.2 Modèles construits à partir d'un arbre de classification sémantique 103
5.4 Expérimentation 104
5.4.1 Apprentissage des modèles de langages spécifiques 105
5.4.2 Perplexité 106
5.4.3 Reconnaissance de la parole 107
5.5 Conclusions 112

5.1 Motivations

5.1.1 Travaux existants

L'utilisation d'un modèle de langage spécifique à une application donne de meilleures résultats que l'utilisation d'un modèle de langage généraliste.

De nombreux travaux tendent à adapter les probabilités d'un modèle de langage en fonction du contexte d'élocution. (Bigi et De Mori, 2000,) propose une méthode basée sur un système de mémoire cache pour détecter le thème du discours et adapter les coefficients d'interpolation linéaire d'une mixture de modèles thématiques. Cette

approche, efficace pour des applications de dictée vocale plus généraliste, est peu appropriée au type d'applications de dialogue de notre étude : le thème du dialogue est déjà très contraint et les phrases prononcées sont courtes.

Notre approche ne consiste pas à adapter les probabilités d'un modèle de langage, comme c'est le cas pour les modèles à mémoire cache (Kuhn et De Mori, 1990,) ou les modèles à base de *triggers* (Lau *et al.*, 1993), mais à créer des modèles de langage spécialisés pour certains types de phrases, et de choisir au cours du dialogue le modèle le plus approprié.

5.1.2 Particularités de l'étude et propositions

Nos travaux visent à améliorer les performances d'un module de reconnaissance de la parole intégré dans un système de dialogue. Pour parvenir à cette amélioration, nous agissons sur le(s) modèle(s) de langage utilisé(s) par le module de reconnaissance. Dans le contexte de cette étude (le démonstrateur AGS de France Télécom), le module de reconnaissance est indépendant du module qui gère le dialogue : ni l'état du dialogue, ni son historique ne sont connus de l'unité de reconnaissance.

L'application de dialogue de notre étude vise un domaine bien particulier, qui détermine un champ sémantique stable. En l'occurrence, il s'agit de la recherche par l'utilisateur de serveurs spécialisés dans les prévisions météorologiques ou la recherche d'emploi. Cette contrainte forte sur les thèmes potentiels du dialogue induit une faible variabilité des structures syntaxiques et lexicales des phrases. En s'appuyant sur cette faible variabilité, il est possible de créer des ensembles de phrases structurellement proches. Pour chaque ensemble de phrases, un modèle de langage spécialisé peut être créé.

Ces modèles de langage spécialisés sont utilisés au niveau du module de reconnaissance en deux passes. La première passe utilise un modèle de langage généraliste, génère un graphe de mots, et propose une première hypothèse de reconnaissance. L'analyse de cette hypothèse permet de choisir le modèle de langage qui semble le plus approprié pour la seconde passe. Celle-ci consiste en la recherche d'une nouvelle hypothèse à partir du graphe de mots généré en première passe. L'hypothèse retenue est l'hypothèse obtenue lors de la seconde passe.

La figure 5.1 illustre le schéma d'utilisation de ce type de modèles spécialisés. La nécessité d'un processus en deux passes vient du manque d'information *a priori* détenue par le module de reconnaissance sur la phrase prononcée par l'utilisateur. La première passe, effectuée à l'aide d'informations générales, émet une première hypothèse qui apporte des informations. Ces informations doivent être manipulées avec précaution car elles ne sont pas forcément très fiables : l'hypothèse peut contenir des erreurs. La seconde passe utilise les informations retenues pour émettre une nouvelle hypothèse à l'aide d'un modèle de langage spécialisé : celui-ci est censé être mieux adapté à la situation que le modèle de langage généraliste.

Deux approches sont proposées pour construire les modèles spécialisés. Une première approche présentée dans (Béchet *et al.*, 2001a) utilise des connaissances *a priori* pour regrouper de manière automatique les phrases en un nombre prédéfini

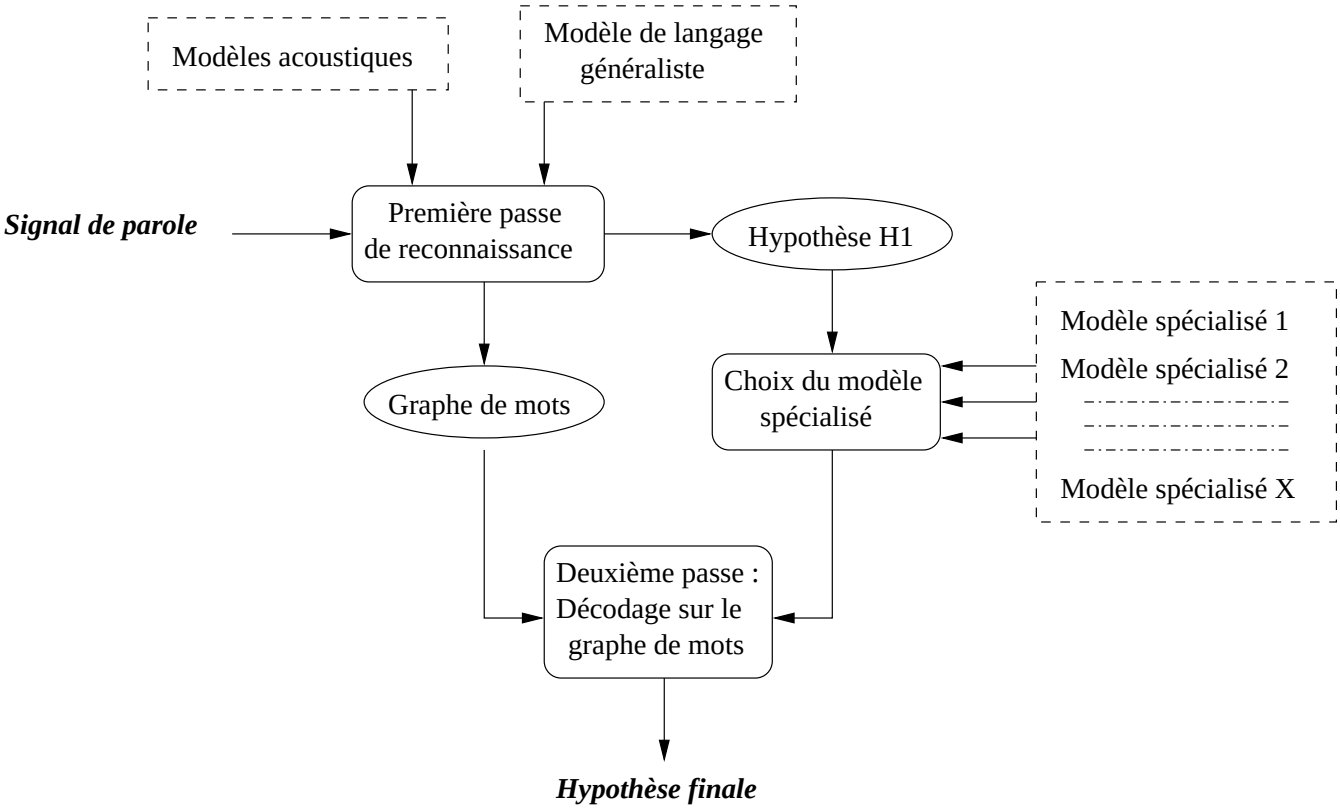


FIG. 5.1 – *Architecture générale de l'utilisation de modèles de langage spécialisés*

de classes. La seconde approche construit de manière automatique et sans connaissance *a priori* un nombre non déterminé de classes de phrases. Cette approche, présentée dans (Estève *et al.*, 2000), utilise la notion d'arbre de classification sémantique.

5.2 Construction de modèles spécifiques

Le principe général de construction de modèles de langage spécialisés est le même pour les deux approches : il s'agit de scinder le corpus d'apprentissage en sous-corpora de phrases de même nature, et d'utiliser chacun de ces sous-corpora afin d'estimer un modèle de langage spécifique. La figure 5.2 illustre ce principe d'apprentissage des modèles spécialisés.

La première méthode de création des sous-corpora consiste à regrouper les phrases en quatre grandes catégories prédéfinies à l'aide de connaissances *a priori*. Les quatre sous-corpora obtenus sont utilisés pour l'estimation de quatre modèles spécialisés. La seconde méthode, basée sur l'utilisation d'un arbre de classification sémantique, scinde le corpus d'apprentissage en un nombre non prédéfini de sous-corpora, sans connaissances *a priori* hormis la liste des mots du vocabulaire de l'application.

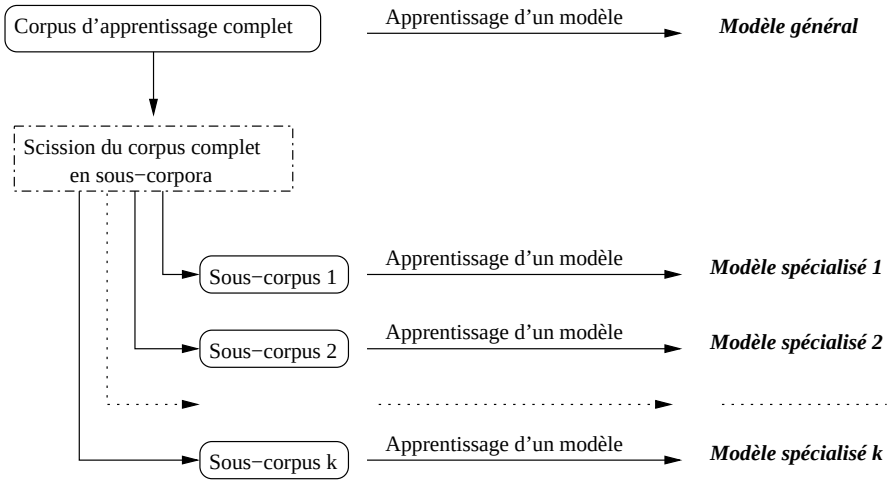


FIG. 5.2 – *Schéma général de construction des modèles de langage spécialisés*

5.2.1 Étiquetage des phrases à l'aide de connaissances *a priori*

Pour le type de dialogue visé, les phrases peuvent être regroupées en quatre grandes catégories :

1. les REQUÊTES : ce sont des phrases comme "je voudrais le numéro de téléphone de ...". La première phrase prononcée par un utilisateur au début d'une session de dialogue est généralement une requête.
2. les QUESTIONS,
3. les RÉPONSES,
4. toutes les AUTRES phrases : ce peut être des commandes ("suivant", "annulation"), ou encore des phrases hors-sujet.

Après analyse manuelle des structures lexicales et syntaxiques des phrases du corpus d'apprentissage, il est possible de créer quelques règles simples pour effectuer une classification automatique des phrases entre ces quatre catégories.

Ces règles portent sur la présence dans les phrases de certains mots, de certaines classes syntaxiques ou encore de certaines structures syntaxiques : ce peut être la présence d'un pronom interrogatif ou une inversion sujet-verbe qui permet de cataloguer une phrase comme QUESTION, ou encore la présence du verbe *vouloir* dans une phrase qui entraîne son étiquetage comme REQUÊTE si cette phrase n'est pas une QUESTION.

Ces règles se présentent sous la forme d'un arbre de décision construit manuellement. Avant de procéder à la classification à l'aide de l'arbre de décision, il est toutefois nécessaire d'effectuer quelques traitements sur les phrases du corpus initial, comme un étiquetage syntaxique des mots, ou une analyse grammaticale partielle.

La figure 5.3 illustre un exemple d'arbre de décision de ce type. Dans cet exemple, chaque phrase est étiquetée et une analyse grammaticale partielle est effectuée. Ensuite, la phrase est présentée à l'arbre de décision dont la première question concerne la présence éventuelle d'une structure de requête. Cette structure est révélée au préalable par l'analyse grammaticale partielle. Si une telle structure existe dans la phrase, celle-ci est classée comme étant une REQUÊTE, sinon une nouvelle question est posée. Si la phrase répond négativement à toutes les questions, elle est considérée comme faisant partie de la catégorie AUTRE. Il est important de noter que l'ordre des questions est fondamental. La hiérarchisation des questions rend naturelle l'utilisation d'un arbre de décision qui, structurellement, est parfaitement adapté à ce type de classification.

En utilisant, de manière automatique, l'arbre de décision construit manuellement, quatre sous-corpora sont obtenus, spécialisés dans les phrases de type question, requête, réponse ou autre[1].

Les sous-corpora obtenus sont alors utilisés pour estimer les quatre modèles de langage spécialisés correspondants, selon le principe général décrit par la figure 5.2.

5.2.2 Utilisation d'un arbre de classification sémantique

Les premiers travaux portant sur l'application des arbres de décision au traitement du langage naturel ont été proposés dans (Bahl *et al.*, 1990). Dans ces travaux, un arbre de décision est utilisé pour établir une modélisation probabiliste du langage : la probabilité $P(w)$ d'apparition du mot w est calculée à partir des réponses (issues de l'analyse de la phrase) aux questions de l'arbre. Ces questions portent sur la position des mots précédents. La notion d'arbre de classification sémantique, introduite dans (Kuhn et De Mori, 1995,), ressemble au type d'arbre évoqué dans (Bahl *et al.*, 1990), mais s'appuie sur des expressions régulières. Avant d'évoquer plus en détails les arbres de classification sémantique, un rappel sur les arbres de décisions et de régressions, décrits dans (Breiman *et al.*, 1984), est présenté.

5.2.2.1 Arbres de classification et de régression

Généralités Un arbre T est constitué d'un ensemble de noeuds $TN = \{t_0, t_1, ..., t_I\}$. Chaque noeud t_i a un unique noeud parent noté $PAR(t_i)$ (excepté le noeud racine t_0 qui n'a aucun parent) et un ensemble $K(t_i)$ de noeuds fils qui est un sous-ensemble de TN. Le niveau de profondeur du noeud t_i est calculé récursivement comme suit :

[1] Pour mesurer le degré de précision de cette classification, il faudrait étiqueter manuellement les phrases du corpus d'apprentissage et comparer leur étiquette avec le résultat de la classification obtenue à l'aide des questions de l'arbre de décision. L'étiquetage manuel d'un corpus de 10.000 phrases est un travail laborieux que nous n'avons pas réalisé : nous considérons que si certaines erreurs de classification se produisent (et nous en avons constatées très peu), la même erreur se produira lors de l'utilisation de l'arbre sur le corpus de test. Si certaines erreurs interviennent (par exemple une requête considérée comme une question), les phrases concernées ont généralement une construction syntaxique proche des phrases appartenant à la classe qui leur a été attribuée : c'est ce qui importe le plus pour la modélisation du langage.

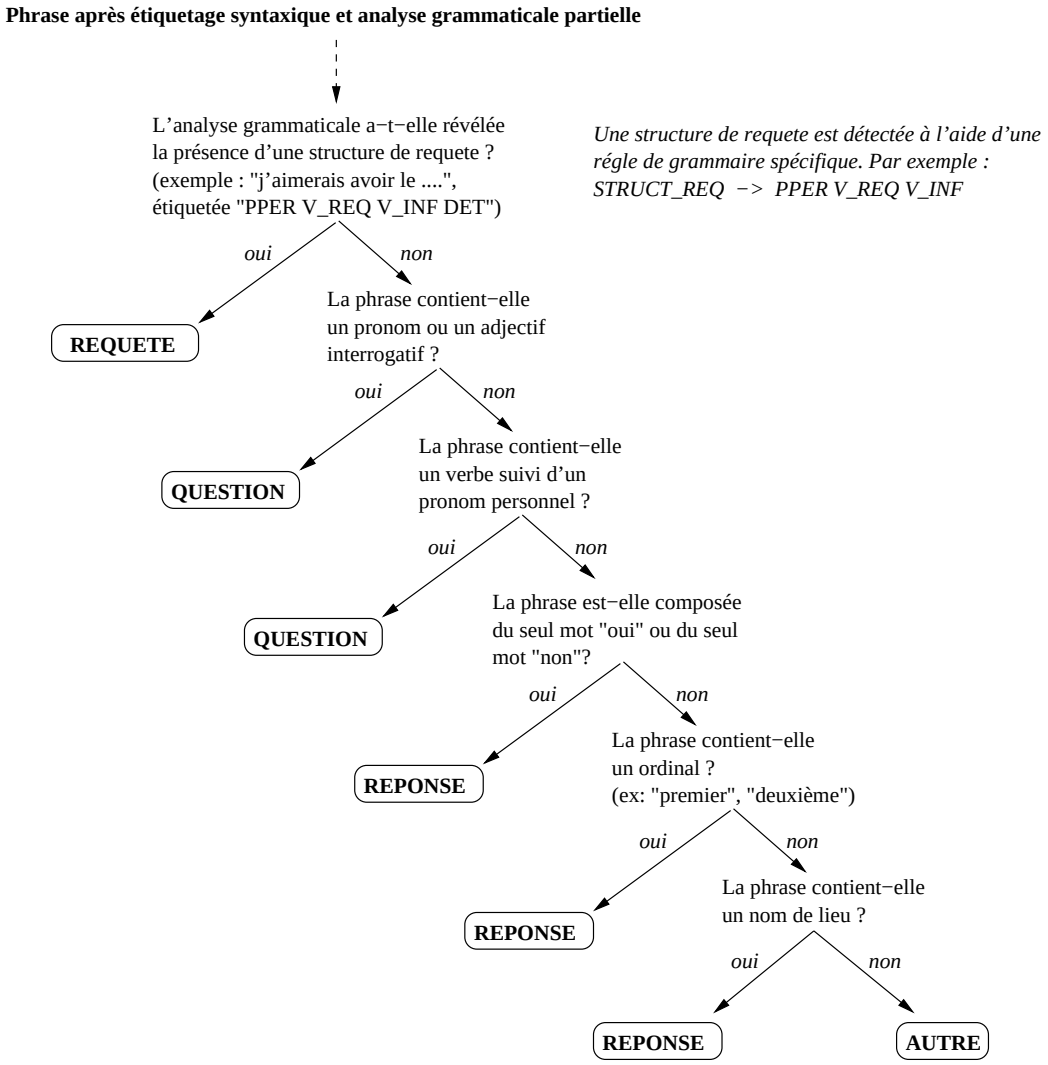


FIG. 5.3 – *Exemple d'arbre de décision construit manuellement à partir de connaissances a priori et de l'observation du corpus d'apprentissage afin de classer les phrases en quatre catégories prédéfinies*

$$DEPTH(t_i) = \begin{cases} 0 & \text{si } i = 0 \\ DEPTH(PAR(t_i)) + 1 & \text{sinon} \end{cases}$$

L'ensemble L des noeuds feuilles est l'ensemble des noeuds de TN qui n'ont pas de fils.

Chaque noeud t_i non feuille est associé à une question $Q(i)$. Cette question partitionne le noeud t_i en ses noeuds fils. Une fonction de probabilité $P(v|t)$ est également associée aux noeuds, y compris aux noeuds feuilles. La probabilité $P(v|t_i)$ est la probabilité d'apparition de l'événement v pour le noeud t_i. L'événement v est un des événements pour lesquels les questions Q sont posées et sa présence dans le noeud t_i est conditionnée par les réponses aux questions associées aux ancêtres de t_i. Ces questions permettent de classer les événements en fonction de leurs réponses.

L'impureté H d'un noeud t_i calculée à partir de l'entropie, s'écrit :

$$H(t_i) = -\sum_{v \in V} f(v|t_i) log_2 f(v|t_i) \tag{5.1}$$

où V est l'ensemble des événements pour lesquels sont posées les questions Q et $f(v|t_i) = \frac{c(v,t_i)}{c(v)}$, avec $c(v,t_i)$ le nombre d'événements v observés dans le corpus d'apprentissage qui ont répondu correctement aux questions menant au noeud t_i et $c(v)$ le nombre d'événements v observés dans le corpus d'apprentissage.

Plus la valeur de l'entropie d'un noeud est faible, plus la classification engendrée par les questions qui ont permis d'atteindre ce noeud est pertinente : plus l'entropie est réduite, plus les questions apportent de l'information.

Une autre mesure d'impureté souvent utilisée est l'index de Gini, ou index de diversité :

$$G(t_i) = 1 - \sum_{v \in V} P(v|t_i)^2 \tag{5.2}$$

Comme pour l'entropie, la réduction de la valeur de l'index de Gini peut être considérée comme le critère déterminant dans le choix des questions.

Construction d'un arbre de décision Un corpus d'apprentissage est associé au noeud racine. Une question est choisie parmi l'ensemble des questions disponibles. Cette question permet de scinder le corpus associé au noeud racine t_0 en plusieurs sous-ensembles : tous les échantillons de ce corpus qui répondent de la même manière sont regroupés dans le même sous-ensemble. Pour une question amenant une réponse de type *oui*/*non*, deux sous-ensembles sont créés en même temps que les deux noeuds fils t_1 et t_2 auxquels ils sont associés. Dans ce cas, le noeud racine t_0 associé au corpus d'apprentissage est scindé en deux noeuds fils t_1 et t_2 : t_1 est associé au sous-corpus constitué des échantillons qui ont répondu *oui* à la

question posée alors que t_2 est associé au sous-corpus constitué des échantillons qui ont répondu *non*.

Une fois la question posée, la réduction de l'impureté ΔH peut être calculée selon la formule suivante :

$$\Delta H = H(t_0) - H(t_1) - H(t_2)$$

où $H(t_i)$ est la valeur pour le noeud t_i de la mesure d'impureté choisie, généralement l'entropie ou l'index de Gini.

Ce processus de scission est répété récursivement sur les noeuds fils tant que la réduction de l'impureté s'avère satisfaisante.

Deux types d'algorithmes de construction sont envisageables : dans le premier, toutes les questions sont posées pour chaque noeud, et l'algorithme choisit la question qui mène à une réduction maximale de l'impureté (c'est-à-dire une majoration de ΔH). Dans ce cas, il s'agit d'un algorithme sous-optimal : la recherche de la réduction maximale de l'impureté d'un noeud ne remet pas en question le choix des ancêtres de ce noeud. La réduction de l'impureté est maximale localement.

Le second type d'algorithme, moins intéressant, choisit la première question qui permet une réduction de l'impureté. Nous n'avons pas retenu cette approche qui semble multiplier les questions retenues avant d'atteindre un critère d'arrêt : cette approche construit un arbre dont la taille est beaucoup plus importante que pour la méthode précédente, sans que nous puissions envisager une meilleure classification.

Les paramètres les plus importants lors de la construction d'un arbre de décision sont le choix des questions et la taille de l'arbre. Le choix des questions est déterminant puisque les questions influencent directement la classification. Le contrôle de la taille de l'arbre s'avère également important : cette taille joue un rôle primordial dans la capacité de généralisation de l'arbre de décision. Le risque de sur-apprentissage doit être évité. Dans ce but, des techniques d'élagage ont été proposées : dans (Gelfand *et al.*, 1991), l'algorithme de construction, qui intègre un processus d'élagage, consiste en un cycle d'itérations expansion/élagage initié sur deux sous-ensembles disjoints et de taille égale du corpus d'apprentissage.

5.2.2.2 Arbre de classification sémantique

Les arbres de classification sémantique sont un type d'arbres de décision particulier. Ils ont été introduits dans (Kuhn et De Mori, 1995,) afin de construire un système de compréhension du langage naturel sans utiliser de règles définies manuellement. Comme pour tout arbre de décision, quatre éléments sont nécessaires à leur construction :

1. Un corpus d'apprentissage : il s'agit d'un ensemble de phrases qui seront regroupées en classes au fur et à mesure de la construction de l'arbre.

2. Un ensemble de questions : les arbres de classification sémantique utilisent des expressions régulières comme questions de type *oui*/*non*. Les questions sont posées sur les phrases du corpus d'apprentissage. Une phrase répond *oui* à une question si elle satisfait l'expression régulière représentant cette question. Elle répond *non* dans le cas contraire.
3. Une règle permettant de choisir la meilleure question pour un noeud donné. Une mesure d'impureté est nécessaire : l'index de Gini G est utilisé dans (Kuhn et De Mori, 1995,), mais d'autres mesures sont applicables. La question retenue est celle qui permet d'obtenir la meilleure réduction d'impureté pour passer d'un noeud t à ses deux noeuds fils. Si les deux fils de T sont notés OUI et NON, et si la proportion des échantillons du noeud t qui ont répondu *oui* (respectivement *non*) à la question est notée P_O (resp. P_N), alors la réduction d'impureté ΔI, calculée à partir de l'index de Gini, s'écrit :

$$\Delta I = G(t) - P_O * G(OUI) - P_N * G(NON)$$

4. Une méthode d'élagage afin d'éviter le sur-apprentissage : l'algorithme présenté dans (Gelfand *et al.*, 1991) peut être utilisé. Il est également possible d'utiliser des seuils, par exemple un nombre minimum de phrases associées à un noeud, ou une valeur minimale de la réduction de l'impureté, pour stopper la construction de l'arbre.

Particularités L'expression régulière utilisée comme question au niveau des noeuds d'un arbre de classification sémantique est appelée structure connue (KS : *Known Structure* en anglais). Cette expression régulière est formulée à partir des symboles suivants :

1. $<$ et $>$, qui signifient respectivement le début et la fin de l'expression régulière,
2. $+$, qui représente un ensemble non vide de mots indéterminés : $+$ représente n'importe quelle séquence de mots,
3. les mots du lexique.

Ainsi, l'expression régulière $<$ je voudrais $+$ numéro $+$ $>$ accepte les phrases qui commencent par "je voudrais" et qui contiennent le mot "numéro". Toutes les phrases qui répondent à ces critères, quels que soient les mots existant entre "voudrais" et "numéro" et quels que soient les mots composant la fin de la phrase, répondront *oui* à la question définie par cette expression régulière. Toutes les autres répondront *non*.

L'originalité de la construction des arbres de classification sémantique provient de l'élaboration des questions : la première question s'écrit $< + >$. Ceci entraîne que toutes les phrases du corpus d'apprentissage sont regroupées au niveau du noeud racine. A chaque noeud dont les échantillons associés ont répondu *oui* à la question du noeud parent, une question est créée en remplaçant un symbole $+$ de la question du noeud parent par une des quatre expressions suivantes :

1. w_i (w_i un mot du lexique ou d'une partie déterminée du lexique)

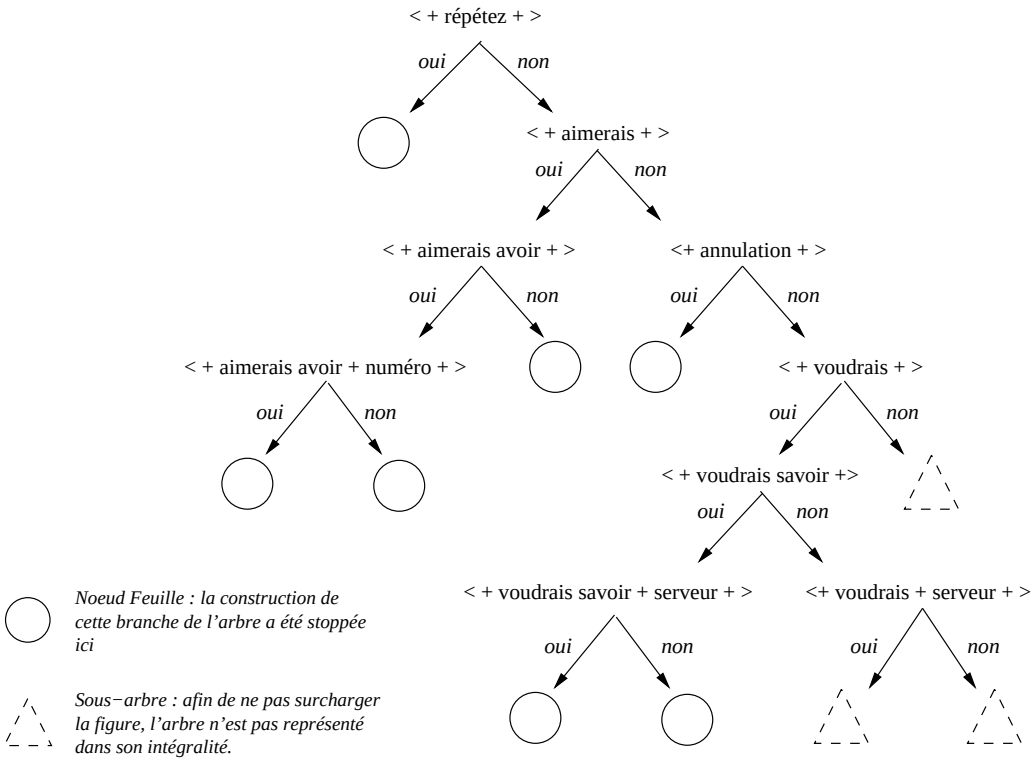


FIG. 5.4 – *Exemple abrégé d'un arbre de classification sémantique*

2. w_i+
3. $+w_i$
4. $+w_i+$

La totalité des échantillons du corpus d'apprentissage est associée au noeud racine : comme nous l'avons vu, la question virtuelle[2] menant au noeud racine s'écrit $< + >$.

Pour le noeud associé aux phrases ayant répondu positivement à la question du noeud parent, il y aura $4 \times n \times m$ expressions régulières à construire et à tester, en supposant qu'il existe n symboles $+$ dans la question du noeud parent et m mots dans le lexique ou la partie prédéfinie du lexique utilisée pour les questions. Seule la meilleure question sera conservée.

Pour les noeuds dont les échantillons associés ont répondu *non*, on considère que la question à modifier est la dernière question à laquelle ses échantillons ont répondu *oui*, et le même processus s'applique.

Afin d'ajouter des connaissances et d'obtenir une généralisation au niveau des questions, nous avons modifié la portée des questions. Comme il est écrit précédemment, les questions sont normalement construites à partir des mots du lexique. Seul le niveau lexical est donc généralement pris en compte. En étiquetant chaque mot du corpus d'apprentissage avec sa classe syntaxique et son lemme, nous avons modifié l'algorithme afin de prendre en compte trois niveaux : le niveau lexical, le niveau syntaxique, et le niveau lemmatique. Cela est particulièrement intéressant pour les noms de villes ou de régions. Ainsi, dans la continuité de l'arbre de classification sémantique présenté dans la figure 5.4, il est alors possible de rencontrer des expressions régulières du type : < + *voudrais* + *serveur* + *XVILLE*>.

Construction de l'arbre de classification sémantique Le corpus d'apprentissage est scindé au fur et à mesure de la génération de l'arbre : lorsque cette construction est terminée, chaque noeud de l'arbre est associé à un sous-ensemble du corpus d'apprentissage. Chaque sous-corpus est utilisé afin d'estimer un modèle de langage *n-gram*. Au final, chaque noeud de l'arbre est associé à un modèle de langage.

Pour la construction de l'arbre de classification sémantique, aucune mesure d'impureté comme la mesure d'entropie ou l'index de Gini n'est utilisée pour le choix de la question. Nous proposons d'utiliser le critère de réduction maximale de la perplexité pour choisir la question. La question choisie est celle qui permet de maximiser la valeur ΔPP calculée comme suit :

$$\Delta PP = PP(t) - PP(OUI/NON)$$

où $PP(t)$ est la perplexité calculée sur les échantillons associés au noeud père, du modèle associé au noeud père, et $PP(OUI/NON)$ est la perplexité des modèles

[2]Toutes les phrases du corpus d'apprentissage répondent par l'affirmative à cette question : elle est donc inutile pour scinder le corpus. Par contre, elle sert de point de départ à l'élaboration des questions suivantes.

associés aux deux noeuds fils. La mesure du gain de perplexité ne nécessite pas d'étiquetage *a priori* des phrases.

Les modèles de langages sont construits de manière hiérarchique : plus on descend dans l'arbre, plus les modèles sont spécialisés. Cette architecture permet d'adapter le niveau de spécialisation du modèle de langage (Béchet *et al.*, 2001b).

5.3 Sélection des modèles

La principale difficulté rencontrée dans l'utilisation des modèles spécialisés lors d'une session de dialogue est le choix du modèle le plus pertinent pour la reconnaissance de chaque nouvelle phrase prononcée par l'utilisateur.

Il n'est pas intéressant d'utiliser une mixture de modèles[3] quand un grand nombre de modèles spécialisés est disponible. Utiliser une mixture de modèles dans ce cas peut ralentir considérablement le processus de reconnaissance[4], mais pose également le problème de l'estimation des coefficients d'interpolation qui pondèrent chacun des nombreux modèles de langage au sein de la mixture. Surtout, chaque modèle de langage spécialisé détient de l'information très spécifique, et a un comportement hasardeux en dehors de son domaine de compétences. Chaque modèle spécialisé dans un domaine différent du contexte de reconnaissance risque d'introduire du bruit dans la mixture de modèle, plutôt que de l'information pertinente.

Nous proposons une approche en deux passes, illustrée par la figure 5.1, où les résultats de la première passe, effectuée à l'aide du modèle général, sont utilisés pour sélectionner un modèle spécialisé pertinent. Les deux méthodes présentées ici suivent ce principe général et utilisent les arbres qui ont permis de générer les modèles spécialisés pour choisir le modèle adéquat.

La première méthode concerne les modèles spécialisés construits par classification des phrases en quatre catégories [REQUÊTE, QUESTION, RÉPONSE et AUTRE]. La seconde méthode est proposée pour les modèles construits à partir d'un arbre de classification sémantique.

5.3.1 Modèles construits par étiquetage des phrases

Appelons H_1 l'hypothèse obtenue lors de la première passe avec le modèle de langage *n-gram* généraliste.

Nous proposons d'appliquer l'arbre de décision qui a permis la classification des phrases du corpus d'apprentissage à l'hypothèse H_1. Une des étiquettes [REQUÊTES, QUESTION, RÉPONSE ou AUTRE] est associée à H_1 en utilisant l'arbre qui a étiqueté les phrases du corpus d'apprentissage. Le modèle spécialisé choisi est alors le modèle associé à l'étiquette donnée à H_1.

[3] voir section 3.3.3

[4] Ce ralentissement existe lorsque les valeurs renvoyées par la mixture de modèles sont calculées à la volée. Si les coefficients d'interpolation restent statiques, il est possible d'éviter (ou de réduire considérablement) ce ralentissement en pré-calculant les valeurs fournies par la mixture de modèles.

L'hypothèse H_1 peut comporter des erreurs qui faussent la sélection du modèle de langage. Comme nous le verrons plus loin, peu d'erreurs d'étiquetage de H_1 sont observées par cette méthode.

5.3.2 Modèles construits à partir d'un arbre de classification sémantique

Comme pour l'approche précédente, l'arbre T qui a généré les modèles spécialisés lors de la phase d'apprentissage est mis à contribution pour le choix du modèle spécialisé en phase de reconnaissance.

Soit H_1 la meilleure hypothèse obtenue lors du décodage d'un graphe en première passe avec le modèle *n-gram* général. Pour choisir un modèle spécialisé, la méthode consiste à parcourir l'arbre T à l'aide de l'hypothèse H_1. Le modèle choisi correspond au modèle associé au noeud de T atteint par H_1.

Il est aussi intéressant d'utiliser les n meilleures hypothèses obtenues en première passe (Béchet *et al.*, 2001b). Ces hypothèses parcourent ensemble l'arbre T. Dès que des hypothèses choisissent des chemins différents, la descente stoppe : cette méthode permet de sélectionner un modèle de langage plus ou moins généraliste. Le modèle choisi est le modèle associé au dernier noeud que toutes les hypothèses H_i avaient atteint ensemble.

Cette méthode est illustrée par la figure 5.5.

Le parcours de l'arbre se fait, au choix, selon deux critères : soit en fonction des expressions régulières associées aux noeuds de l'arbre de classification sémantique, soit en fonction de la mesure de perplexité des modèles spécialisés également associés à ces noeuds.

Utilisation des expressions régulières Si le critère choisi est l'utilisation des expressions régulières des noeuds, l'hypothèse H_1 est comparée à l'expression régulière proposée par un noeud pour continuer le parcours de l'arbre. Si elle satisfait à l'expression régulière, le noeud suivant parcouru par l'hypothèse H_1 est le noeud fils associé à la réponse OUI, sinon, il s'agit de celui associé à une réponse négative.

Utilisation de la mesure de perplexité Si un des noeuds fils du noeud courant est associé à un modèle dont la mesure de perplexité calculée sur l'hypothèse H_1 est inférieure à celle du modèle courant, le parcours de l'arbre est poursuivi avec ce noeud fils. Dès qu'il n'y a plus de diminution de la perplexité, le parcours est stoppé et le modèle choisi est le modèle associé au dernier noeud atteint. Dans (Béchet *et al.*, 2001b) et dans (Estève *et al.*, 2001), cette méthode est suivie à l'aide d'une liste de *n*-best : dès qu'il y a un désaccord sur la suite du parcours de l'arbre entre les n hypothèses, le parcours se termine sur le noeud courant. Cette utilisation des *n*-best permet de conserver un certain niveau de généralisation du modèle de langage choisi.

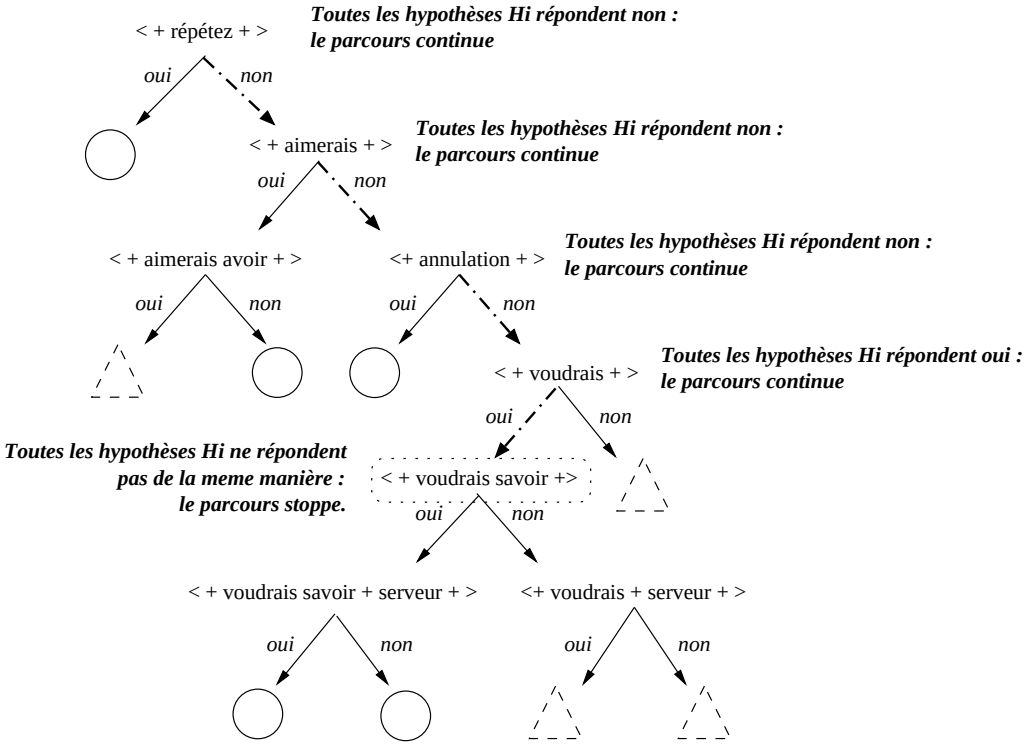


FIG. 5.5 – *Exemple de parcours d'un arbre de classification sémantique par une liste d'hypothèses produites en première passe pour le choix d'un modèle de langage spécialisé*

5.4 Expérimentation

Les expériences ont été effectuées sur les données décrites en section 4.6.

Seuls des modèles *bigrams* sont utilisés : comme le corpus d'apprentissage initial n'est pas grand et que les méthodes de construction des modèles de langage spécialisés suppose de scinder plusieurs fois ce corpus, les sous-corpora obtenus sont de tailles réduites (voir tableau 5.1). Cette faible taille des corpora d'apprentissage implique une faible robustesse statistique : les modèles *trigrams* étant plus sensibles au manque de données que les modèles *bigrams*, seuls ces derniers sont utilisés.

Dans cette section qui récapitule certaines des expériences menées sur les modèles de langage spécifiques, nous revenons d'abord sur la construction de ces modèles. Il faut différencier les modèles issus d'une scission du corpus à l'aide de connaissances *a priori* , que nous notons LM_{Cat} des modèles issus d'une scission purement statistique, notés LM_{Stat}. Pour ces derniers modèles, les deux méthodes de sélection présentées en section 5.3 sont utilisées. La méthode basée sur l'usage d'expressions régulière est notée $Select_{Pattern}$. La méthode basée sur la réduction de la perplexité est notée $Select_{PP}$.

Les valeurs des perplexités des différents modèles sont ensuite présentées.

5.4.1 Apprentissage des modèles de langages spécifiques

5.4.1.1 Modèles construits à partir de connaissances *a priori* (LM_{Cat})

Quatre sub-corpora de phrases [REQUÊTE, QUESTION, RÉPONSE et AUTRE] ont été sollicités pour l'apprentissage de quatre modèles spécifiques. Ces sub-corpora sont issus de la scission du corpus d'apprentissage initial, qui comporte 9842 phrases. Cette scission a été effectuée à l'aide d'un arbre de décision construit manuellement selon la méthode décrite en section 5.2.1 et illustrée par l'exemple de la figure 5.3.

Il y a un modèle de langage spécialisé pour les requêtes, un modèle pour les réponses, un autre pour les questions et enfin un modèle pour toutes les autres phrases.

Le tableau 5.1 montre la répartition des phrases du corpus d'apprentissage initial entre les quatre catégories de phrases.

	REQUÊTE	QUESTION	RÉPONSE	AUTRE
Nombre de phrases	2128	1132	5140	1442
Proportion	21,62 %	11,50 %	52,23 %	14,65 %

TAB. 5.1 – *Répartition des phrases du corpus d'apprentissage en fonction de l'étiquetage effectué par l'arbre de décision construit manuellement*

Le tableau 5.2 montre la répartition des phrases du corpus de test entre ces mêmes catégories.

	REQUÊTE	QUESTION	RÉPONSE	AUTRE
Nombre de phrases	286	195	738	203
Proportion	20,11 %	13,71 %	51,90 %	14,28 %

TAB. 5.2 – *Répartition des phrases de référence du corpus de test en fonction de l'étiquetage effectué par l'arbre de décision utilisé pour scinder le corpus d'apprentissage*

Le corpus de test et le corpus d'apprentissage présentent une distribution similaire des types de phrases qui les composent. La catégorie RÉPONSE est, de loin, celle qui englobe le plus grand nombre de phrases (plus de 50% des phrases dans les deux corpora).

Cette disproportion est logique pour le type de dialogue associé aux données de l'expérimentation. L'application AGS consiste à fournir à l'utilisateur une réponse à une question ou à une requête. Le dialogue qui s'engage alors entre l'utilisateur et la machine suit le schéma stable suivant :

1. la machine accueille l'utilisateur et lui propose de formuler sa requête,
2. l'utilisateur exprime ses besoins (requête ou question)

	Apprentissage	Test
bigram généraliste	*11,65*	*27,18*
LM_{Cat}	8,20(-29,61%)	20,52 (-24,50%)
LM_{Stat}, $Select_{PP}$	7,48 (-35,8%)	19,43 (-28,5%)
LM_{Stat}, $Select_{Pattern}$	4,58 (-60,7%)	17,05 (-37,3%)

TAB. 5.3 – *Comparaison de la perplexité obtenue par les différents modèles bigrams sur les corpora d'apprentissage et de test.*

3. la machine tente de satisfaire au mieux les besoins de l'utilisateur en lui posant des questions afin de confirmer et préciser sa requête : l'utilisateur est alors sollicité pour répondre à ces questions. Cette phase du dialogue est à l'origine du grand nombre de réponses formulées par l'utilisateur.

5.4.1.2 Modèles de langage spécialisés hiérarchiques (LM_{Stat})

La seconde méthode basée sur la construction d'un arbre de classification sémantique, a permis la production de 53 modèles de langages : un modèle de langage par noeud de l'arbre.

Ces modèles de langage sont hiérarchisés. Cette structure hiérarchique est bâtie sur la structure de l'arbre. Plus le niveau de profondeur d'un noeud est élevé , plus le modèle associé à ce noeud est spécialisé : son champ de compétence est réduit à un ensemble plus restreint de phrases.

5.4.2 Perplexité

La comparaison de la valeur de perplexité des modèles spécialisés avec celle du modèle généraliste suppose de choisir pour chaque phrase le modèle spécialisé adéquat. Il est vain de calculer la perplexité d'un modèle spécialisé sur l'ensemble des phrases du corpus de test ou du corpus d'apprentissage puisque ce modèle contient uniquement des informations sur un sous-ensemble spécifique de ces phrases.

Pour calculer la perplexité d'un modèle spécialisé, il faut choisir pour chaque phrase S le modèle le plus approprié. Pour cela, nous appliquons les techniques de sélection proposées en section 5.3, l'hypothèse H_1 obtenue en première passe étant remplacée par la phrase S elle-même.

Le tableau 5.3 montre la réduction relative très importante de la perplexité pour chacun des modèles de langage spécialisés comparée à celle du modèle *bigram* généraliste. Les différentes abréviations de ce tableau signifient :

– LM_{Cat} : modèles de langage *bigrams* construits et sélectionnés après étiquetage des phrases ou hypothèses en quatre grandes catégories prédéfinies.
– LM_{Stat}, $Select_{PP}$: mêmes modèles de langage que précédemment, mais leur sélection s'effectue sur le critère de la minoration de la perplexité lors du parcours de l'arbre.

– LM_{Stat}, $Select_{Pattern}$: modèles *bigrams* construits à l'aide d'un arbre de classification sémantique et choisis en utilisant les expressions régulières de cet arbre.

Les modèles qui obtiennent la valeur de perplexité la plus faible sur le corpus d'apprentissage sont les modèles construits à l'aide de l'arbre de classification et choisis en utilisant les expressions régulières des noeuds de cet arbre. Ces résultats sont conformes à ce que l'on pouvait attendre : c'est la construction de cet arbre qui a permis les scissions du corpus d'apprentissage initial, scissions générées dans l'objectif de minimiser la perplexité des modèles de langage estimés sur les sous-corpora obtenus.

Les valeurs des perplexités des modèles spécialisés restent bien inférieures à celle du modèle bigram généraliste sur le corpus de test. Il est intéressant de constater que ce sont les mêmes modèles que pour le corpus d'apprentissage qui ont la valeur de perplexité la plus faible sur le corpus de test.

5.4.3 Reconnaissance de la parole

Les résultats des expériences de reconnaissance de la parole menées sur modèles LM_{Cat} et les modèles LM_{Stat} sont présentés séparément car la manière dont ils sont employés diffère.

5.4.3.1 Modèles spécifiques à des catégories de phrases prédéfinies (LM_{Cat})

Afin de choisir le modèle de langage spécifique le plus pertinent, l'hypothèse de reconnaissance H_1 obtenue lors d'une première passe est utilisée.

Le tableau 5.4 montre que l'étiquetage de ces hypothèses s'avère fiable à plus de 90% pour une hypothèse H_1 obtenue à l'aide d'un modèle *bigram* généraliste. Un étiquetage de la phrase reconnue en première passe est considéré comme correct s'il est identique à l'étiquetage de la phrase de référence effectivement prononcée par l'utilisateur. Les hypothèses H_1 ont été obtenues à l'aide d'un modèle *bigram dont* les paramètres ont été estimés sur l'intégralité du corpus d'apprentissage.

	Taux d'étiquetage correct
Graphes I	91 %
Graphes II	90,79 %

TAB. 5.4 – *Taux d'étiquetage correct des phrases reconnues en première passe à l'aide d'un modèle généraliste bigram en fonction de l'ensemble de graphes utilisé pour le décodage*

Pour les expériences effectuées sur les graphes de Type I, l'hypothèse H_1 de première passe est obtenue à la suite d'un décodage sur ces graphes à l'aide d'un modèle *bigram* généraliste estimé sur la totalité du corpus d'apprentissage.

Pour les expériences effectuées sur les graphes de Type II, l'hypothèse H_1 de première passe est obtenue à la suite d'un décodage sur ces graphes à l'aide du même modèle *bigram* généraliste.

Les résultats obtenus sont décrits dans le tableau 5.5.

	Graphes I	Graphes II
Modèle général (première passe, hypothèses H_1)	*24,61*	*23,63*
Modèles spécialisés (deuxième passe)	24,16 (-1,8%)	23,49 (-0,6%)

TAB. 5.5 – *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé*

Ces résultats montrent une légère baisse du taux d'erreurs grâce à l'utilisation des modèles spécialisés, mais cette réduction du taux d'erreurs est peu importante, et n'est statistiquement pas significative.

En introduisant un biais dans l'expérience en utilisant l'étiquette attribuée par l'arbre de décision à la phrase de référence au lieu de l'hypothèse H_1, les résultats de l'expérience donnés par le tableau 5.6 montrent que l'utilisation de modèles spécialisés offre certaines potentialités lorsque le choix du modèle spécialisé est correct. Pourtant, l'utilisation de l'hypothèse H_1 pour le choix du modèle de langage est identique dans plus de $90\ \%$ des cas au choix obtenu en utilisant la phrase de référence (voir tableau 5.4) : cette précision n'est donc pas suffisante et doit être améliorée pour obtenir des résultats significatifs.

	Graphes I	Graphes II
Modèle général	*24,61*	*23,63*
Modèles spécialisés (choisi à l'aide de la référence)	23,67 (-3,8%)	22,80 (-3,5%)

TAB. 5.6 – *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi à partir de l'étiquette attribuée à la phrase de référence*

Il est intéressant de comparer les performances de chacun des modèles spécialisés aux performances du modèle généraliste en fonction de l'étiquette donnée à partir de l'hypothèse H_1.

Le tableau 5.7 présente les résultats de ces comparaisons pour des expériences menées respectivement sur les graphes de type I. Les résultats sur les graphes de type II sont similaires.

Les phrases étiquetées comme REQUÊTES sont mieux reconnues à l'aide du modèle général. Par contre, les phrases étiquetées comme RÉPONSES ou AUTRES sont mieux reconnues à l'aide du modèle spécialisé correspondant. Pour les phrases étiquetées comme QUESTIONS, le modèle spécialisé et le modèle général semblent avoir des performances équivalentes.

Ce type d'étude sur le comportement des modèles de langage permet d'élaborer une stratégie fondée sur les différentes performances des modèles selon le type de phrases. Le chapitre 6 propose une stratégie basée sur cette approche.

Il est également intéressant de comparer les résultats obtenus par les modèles spécialisés aux résultats obtenus par le modèle de langage général lorsque le choix

	bigram général (w.e.r)	bigrams spécialisés (w.e.r)
Requêtes	17,60	17,97 % (+2,1%)
Réponses	28,47	27,37 (-3,9%)
Questions	25,54	25,26 (-1,1%)
Autres	35,45	33,95 (-4,2%)

TAB. 5.7 – *Comparaison des taux d'erreurs sur les mots obtenus par le système de reconnaissance avec le modèle bigram général ou avec un modèle bigram spécialisé choisi en fonction de la catégorie attribuée à l'hypothèse de première passe H_1 pour les graphes de l'ensemble I*

	bigram général (w.e.r)	bigrams spécialisés (w.e.r)
Requêtes	19,05	19,09 (+0,2%)
Réponses	27,35	25,83 (-5,6%)
Questions	27,05	25,39 (-6,1%)
Autres	33,33	30,40 (-8,8%)

TAB. 5.8 – *Comparaison des taux d'erreurs sur les mots obtenus par le système de reconnaissance avec le modèle bigram général ou un modèle bigram spécialisé choisi en fonction de la catégorie de la phrase de référence pour les graphes de l'ensemble I*

du modèle est effectué à partir de la phrase de référence. Ces résultats, présentés dans le tableau 5.8 montrent que le modèle spécialisé dans les requêtes a des performances inférieures au modèle général même si le choix du modèle est correct. Par contre, les trois autres modèles ont des performances réellement supérieures au modèle général. Le manque de réduction d'erreurs relevé lors de l'utilisation du modèle spécialisé dans les questions dans le tableau 5.7 semble donc dû à un étiquetage erroné lorsque l'hypothèse H_1 est étiquetée (ou n'est pas mais devrait l'être) comme QUESTION. En effet, lorsque la phrase de référence est utilisée pour choisir le modèle spécialisé, ce modèle est plus performant que le modèle général.

5.4.3.2 Modèles conçus à partir d'un Arbre de Classification Sémantique (LM_{Stat})

Les modèles spécialisés ont été construits à partir d'un arbre de classification sémantique noté T.

Pour choisir le modèle de langage spécialisé qui sera utilisé en seconde passe du processus de reconnaissance, les deux méthodes de sélection présentées en section 5.3 sont expérimentées.

La première méthode, que nous appelons méthode $Select_{Pattern}$, utilise le parcours de T par l'hypothèse H_1 défini en fonction des comparaisons de H_1 avec les expressions régulières associées aux noeuds de T. Aucune liste de n-best n'est utilisée ici : les expériences ont montré peu de modification sensible du choix du modèle.

Ceci s'explique par la ressemblance des n meilleures hypothèses qu'une expression régulière à du mal à différencier.

La seconde méthode, que nous appelons méthode $Select_{PP}$, utilise la réduction consensuelle des valeurs de perplexité calculées sur les n meilleures hypothèse H_i de première passe lors du parcours de l'arbre T. L'utilisation d'une liste de n-best se justifie par le besoin d'éviter que les erreurs éventuelles de l'hypothèse H_1 ne biaisent la sélection des modèles.

Méthode $Select_{Pattern}$ Les résultats des expériences utilisant la méthode de sélection des modèles de langage basée sur la comparaison d'expressions régulières avec l'hypothèse de reconnaissance de la première passe sont présentés dans le tableau 5.9. Les taux d'erreurs obtenus avec cette méthode dégradent les performances de première passe du système de reconnaissance. Cette technique de sélection est trop sensible aux erreurs.

Même en utilisant la phrase de référence pour effectuer la sélection du modèle, la réduction relative du taux d'erreurs obtenue ne dépasse pas $4,5\%$ dans le meilleur des cas.

	Graphes I	Graphes II
Modèle général (première passe, H_1)	*24,61*	*23,63*
Modèles spécialisés choisis à partir de H_1	25,33 (+2,9%)	24,91 (+5,4%)
Modèles spécialisés choisis à partir de Ref	23,5 (-4,5%)	23,16 (-2%)

TAB. 5.9 – *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi par la méthode* $Select_{Pattern}$

Méthode $Select_{PP}$ La figure 5.6 montre que l'utilisation de la liste des *n* meilleurs hypothèses de première passe réduit le taux d'erreurs sur les mots par rapport à l'utilisation de la meilleure hypothèse de première passe H_1 seule : le taux d'erreurs sur les mots passe de $25,5\%$ à $24,4\%$ lorsque la taille de la liste des *n*-best passe de 1 à 4 pour les expériences menées sur les graphes I. Pour les expériences menées sur les graphes II, le taux d'erreurs sur les mots passe de $24,75\%$ à moins de $23,5\%$ lorsque la taille de la liste des *n* meilleures hypothèses passe de 1 à 6.

Ce gain apporté par l'utilisation d'une liste de n-best au lieu de l'utilisation unique de l'hypothèse H_1 s'explique par le niveau de spécialisation des modèles de langage choisis : plus le modèle de langage utilisé est à un niveau profond de l'arbre, plus il est spécialisé. Théoriquement, si le modèle choisi est celui qui correspond réellement à la situation de dialogue, les performances de reconnaissance doivent être améliorées, ce que confirme les résultats du tableau 5.10. Hors, si le modèle de langage choisi n'est pas correct et qu'il est très spécialisé, son utilisation pour le *rescoring* du graphe de mots ne peut que dégrader les performances du système de reconnaissance. En exigeant un consensus entre les n meilleures hypothèses

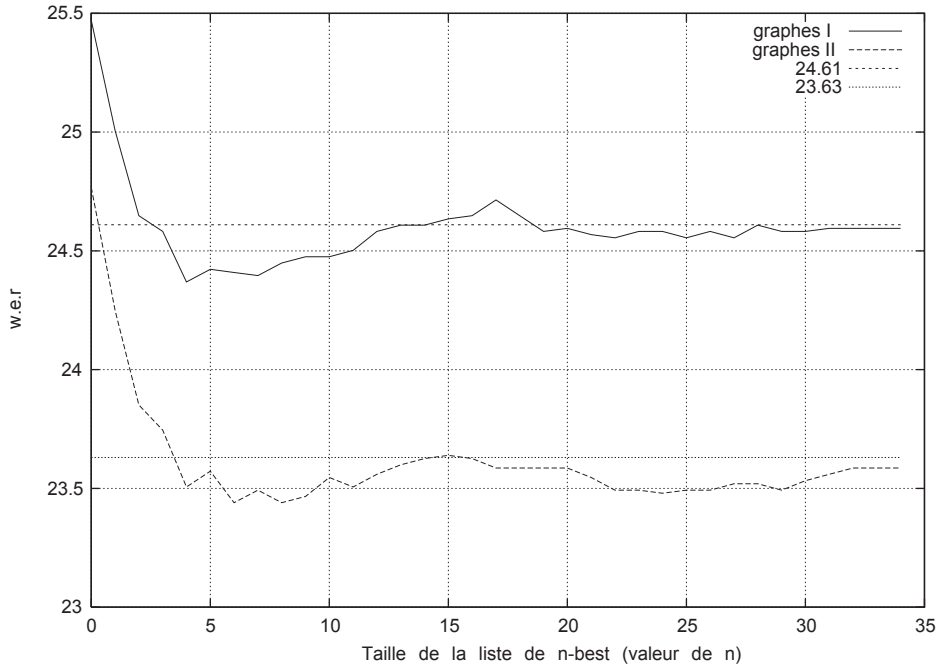


FIG. 5.6 – *Évolution du taux d'erreurs sur les mots en fonction la taille de la liste de n-best utilisée pour la sélection des modèles spécialisés*

de reconnaissance pour le choix du modèle de langage spécialisé, nous limitons la spécialisation de ce modèle : plus taille de la liste des n meilleures hypothèses est élevée, moins le modèle choisi sera spécialisé.

Cependant, les résultats sont décevants lorsqu'ils sont comparés aux taux d'erreurs sur les mots obtenus par le modèle *bigram* généraliste lors de la première passe. Les modèles de langage spécialisés n'améliorent que très peu les performances de reconnaissance lorsque la taille de la liste des n-best atteint 4.

Pourtant, le tableau 5.10 montre le potentiel de ces modèles. Lorsque la phrase de référence est utilisée pour sélectionner les modèles spécialisés, la réduction relative du taux d'erreurs dépasse 13%. Ces résultats montrent que la technique de sélection des modèles de langage basée sur les hypothèses de reconnaissance de la première passe n'est pas suffisamment robuste pour exploiter au mieux les modèles de langage spécialisés.

	Graphes I	Graphes II
Modèle général (première passe, H_1)	*24,61*	*23,63*
Modèles spécialisés choisis à partir de Ref	21,38 (-13,1%)	20,72(-12,3%)

TAB. 5.10 – *Comparaison des taux d'erreurs sur les mots du système de reconnaissance obtenus avec le modèle bigram général ou un modèle bigram spécialisé choisi par la méthode $Select_{PP}$*

5.5 Conclusions

Les expériences montrent que les modèles de langage spécialisés pour un type de phrases spécifique ont un potentiel intéressant : une réduction très importante du taux d'erreurs sur les mots est envisageable. Les modèles de langage LM_{Stat} construits à partir de scissions du corpus d'apprentissage à l'aide d'informations statistiques ont un potentiel supérieur aux modèles LM_{Cat} construits à partir de scissions obtenues à l'aide d'informations *a priori*. De plus, la structure hiérarchique des modèles LM_{Stat} est très intéressante pour maîtriser le niveau de spécialisation des modèles.

La réduction du taux d'erreurs envisageable est tributaire d'une sélection efficace des modèles spécialisés. Les méthodes de sélection proposées ici ne permettent pas d'utiliser pertinemment les modèles spécialisés. Il semble que les informations recueillies lors d'une première passe de reconnaissance sont insuffisantes.

Dans le cadre de notre étude, le module de reconnaissance de la parole et le gestionnaire de dialogue étaient totalement indépendants. Nous avons travaillé à la sélection dynamique de modèles de langage au cours d'une conversation entre un homme et une machine sans aucune connaissance sur l'état du dialogue.

Une solution pour améliorer la sélection des modèles spécialisés consisterait à donner au module de reconnaissance des informations provenant du gestionnaire de dialogue. Ces informations sont porteuses de connaissance sur l'historique et l'état du dialogue. Elles semblent donc pertinentes pour anticiper le type de phrase que l'utilisateur va prononcer, et leur utilisation combinée aux informations obtenues en première passe de reconnaissance avec un modèle général apporterait certainement en fiabilité pour la sélection des modèles de langage spécialisés. Il est évident qu'une coopération plus étroite entre le module de reconnaissance et le gestionnaire de dialogue serait bénéfique à l'ensemble du système de dialogue.

Chapitre 6

Utilisation d'hypothèses de reconnaissance issues de systèmes différents : choix, validation, rejet

Sommaire

6.1 ROVER 114
6.1.1 Présentation générale 114
6.1.2 Alignement de plusieurs hypothèses 114
6.1.3 Vote 116
6.1.4 Quelques résultats 117
6.2 Proposition 117
6.2.1 Informations utilisées 118
6.2.2 Arbre de décision 119
6.3 Expérimentation 122
6.3.1 Utilisation de la technique dite de *leave-one-out* 122
6.3.2 Rappel des performances des différents modèles de langage présentés dans ce mémoire 123
6.3.3 Prise de décision pour deux systèmes de reconnaissance mis en concurrence (modèles *trigrams*) 124
6.3.4 Prise de décision pour les quatre systèmes de reconnaissance à base de modèles bigrams mis en concurrence 127
6.4 Discussion 132

Ces dernières années, divers travaux ont montré l'intérêt d'utiliser les sorties générées par divers systèmes de reconnaissance de la parole pour un même signal de parole. La combinaison de ces différentes hypothèses produit une hypothèse finale de meilleure qualité. Le système le plus courant est le système ROVER (Recognizer Output Voting Error Reduction), introduit dans (Fiscus, 1997).

Nous proposons un système de choix de la meilleure hypothèse basé sur un arbre de décision. La construction automatique de cet arbre de décision à partir de critères prédéfinis est non supervisée et utilise certaines informations recueillies pendant le processus de reconnaissance pour chacune des hypothèses initiales. Ce système est fondé sur des principes différents de ceux du système ROVER.

6.1 ROVER

(Fiscus, 1997) a introduit le système ROVER qui considère que le choix de chaque mot de l'hypothèse de reconnaissance finale est issu d'un vote entre les hypothèses initiales, générées par différents systèmes de reconnaissance de la parole.

6.1.1 Présentation générale

Le système ROVER est construit à partir de deux modules :

1. Un module d'alignement qui crée un réseau de transitions de mots à partir des hypothèses produites par plusieurs systèmes de reconnaissance.
2. Un module de vote : s'il existe au moins deux mots différents portés par des transitions entre deux états consécutifs du réseau de transitions créé par le module d'alignements, le module de vote doit choisir le mot qui sera conservé dans l'hypothèse finale. Ce choix est effectué à partir d'un protocole d'élection, où le mot conservé est le mot qui a recueilli le plus grand nombre de suffrages.

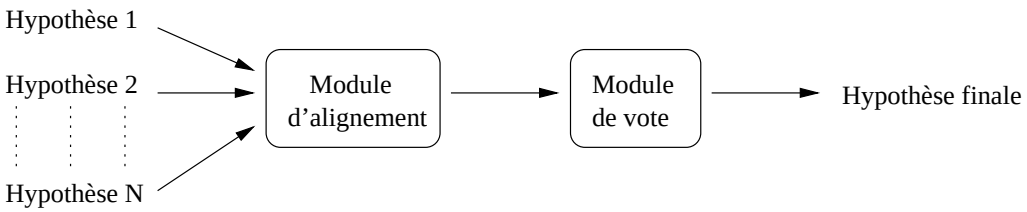


FIG. 6.1 – *Architecture du système ROVER*

La figure 6.1 représente l'architecture générale du système ROVER, qui à partir de plusieurs hypothèses et de ses modules d'alignement et de vote permet d'obtenir une hypothèse finale.

6.1.2 Alignement de plusieurs hypothèses

Dans le système ROVER, l'algorithme d'alignement est un algorithme basé sur la programmation dynamique. Théoriquement, pour obtenir l'alignement optimal l'algorithme de programmation dynamique doit manipuler un espace de recherche hyper-dimensionnel. En pratique, (Fiscus, 1997) propose d'utiliser un algorithme

d'alignement classique de deux réseaux de transitions : cette méthode ne garantit pas d'atteindre l'alignement optimal, mais donne une solution proche dans un temps acceptable.

Plusieurs itérations de cet algorithme sont nécessaires pour un alignement proche de l'alignement optimal.

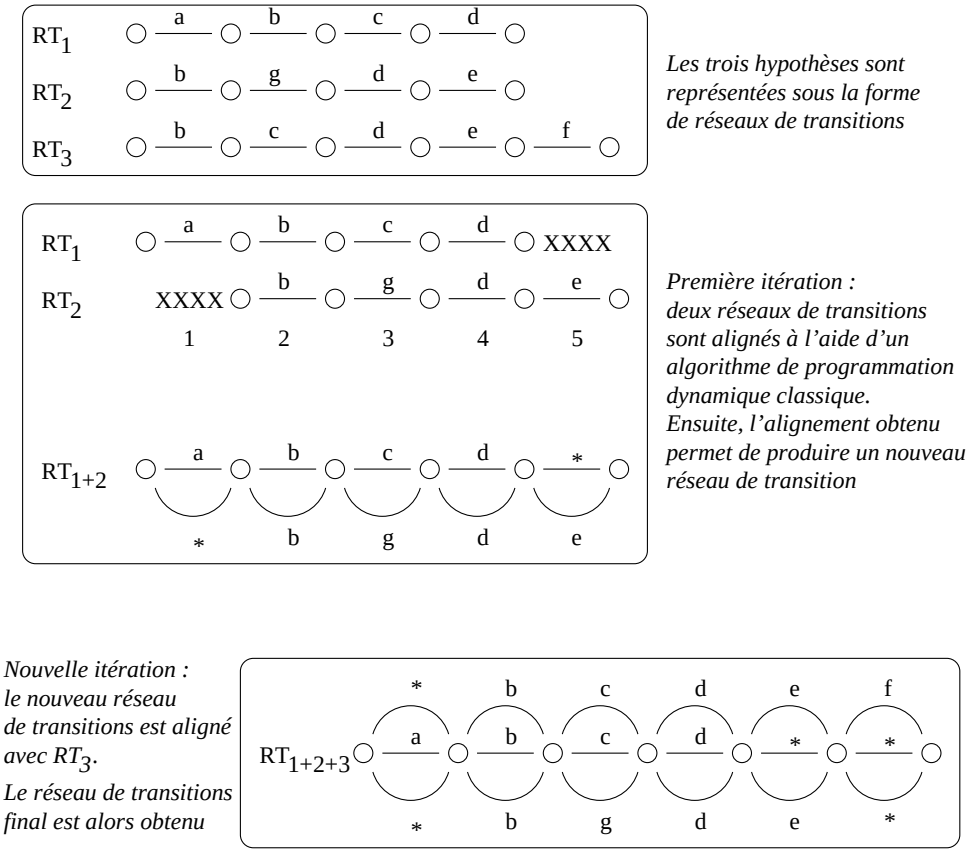


FIG. 6.2 – *Exemple d'alignement de trois hypothèses issues de trois systèmes de reconnaissance de la parole.*

La figure 6.2 illustre un exemple d'alignement de trois hypothèses, représentées chacune sous la forme d'un réseau de transitions.

Tout d'abord, l'algorithme d'alignement de deux réseaux de transitions est utilisé pour les réseaux RT_1 et RT_2, RT_1 étant considéré comme la référence. Ensuite, un nouveau réseau de transitions, appelé RT_{1+2} est créé à partir de l'alignement trouvé entre RT_1 et RT_2. La création de ce réseau de transitions consiste à ajouter des arcs à RT_1 en fonction du résultat de son alignement avec RT_2. Quatre cas sont possibles :

1. RT_1 et RT_2 présentent le même mot au même instant i : une transition portant ce mot est ajoutée dans RT_1 à l'instant i,
2. RT_2 ne présente pas de mot à l'instant i alors que RT_1 en présente un (nous

sommes en présence d'une omission). Un arc portant le mot vide "*" est ajouté à RT_1 à l'instant i.

3. RT_2 présente le mot w à l'instant i alors que RT_1 n'en présente pas (insertion). Un arc portant le mot vide "*" est ajouté à RT_1à l'instant i, ainsi qu'un arc associé au mot w.
4. RT_1 et RT_2 présentent chacun un mot différent au même instant i : le mot présenté par RT_2 est ajouté à RT_1 à l'instant i.

Le réseau de transitions de mots ainsi obtenu, RT_{1+2}, sert alors de référence pour un nouvel alignement avec RT_3. Le même processus est appliqué pour obtenir RT_{1+2+3} .

L'ordre dans lequel les différentes itérations utilisent les réseaux de transitions influe sur le réseau de transitions final.

6.1.3 Vote

Appelons ensemble de correspondances les mots qui se trouvent en concurrence au même instant, c'est-à-dire entre deux noeuds consécutifs dans le réseau de transitions de mots produit par le module d'alignement.

Le module de vote, ou module de *scoring*, permet de choisir un mot pour chaque ensemble de correspondances du réseau de transitions qui lui est présenté. L'ensemble des votes permet de construire l'hypothèse de reconnaissance finale

Plusieurs techniques de vote peuvent être utilisées. Un vote simple, par nombre d'occurrences, consiste à choisir le mot qui apparaît le plus souvent dans un ensemble de correspondances. Si les systèmes de reconnaissance sont en mesure de donner à chaque mot une mesure de confiance, il est également possible de prendre en compte cette mesure en plus de la fréquence d'apparition pour élire le mot qui sera conservé.

La formule générale permettant de donner un score à un mot d'un ensemble de correspondance s'écrit :

$$score(w) = \alpha(N_i(w)/N_s) + (1-\alpha)C_i(w) \tag{6.1}$$

où :

– $N_i(w)$ est le nombre d'occurrences du mot w dans l'ensemble de correspondance de l'instant i.
– N_s est le nombre d'hypothèses de reconnaissance présentées en entrée du système ROVER. Généralement, il est égal au nombre de systèmes de reconnaissance de la parole utilisés.
– $C_i(w)$ est le score de confiance du mot w. Deux valeurs sont envisageables : il s'agit soit de la moyenne des mesures de confiance des différentes transitions portant le mot w dans le même espace de correspondance, soit de la valeur maximale de ces mesures de confiance. Si w est le mot vide, on utilise la valeur $Conf(*)$

qui est une valeur constante représentant le score de confiance du mot vide. Cette valeur est choisie empiriquement sur un corpus de développement. Une mesure de confiance proche de 0 est peu sûre, alors qu'une mesure de confiance proche de 1 est une indication de fiabilité.
– α est un coefficient linéaire compris entre 0 et 1. Lorsque le score ne prend en compte que le nombre d'apparitions d'un mot, alors $\alpha = 1$. Sinon, la valeur de α est déterminée empiriquement sur un corpus de développement, comme $Conf(*)$. Ces deux valeurs sont choisies afin de réduire au plus le taux d'erreurs global sur les mots sur un corpus d'apprentissage.

Pour chaque ensemble de correspondances, le mot retenu est le mot qui obtient le plus grand score.

6.1.4 Quelques résultats

Le système ROVER a été appliqué à partir des soumissions de plusieurs systèmes de reconnaissance de la parole (BBN, CMU, CU, DRAGON, SRI) recueillies lors de l'évaluation NIST, "The LVCSR 1996 Hub-5 Benchmark Test". Les résultats, en terme de taux d'erreurs, des divers systèmes de reconnaissance étaient compris entre $44,9\%$ et $50,2\%$. L'utilisation du système ROVER a permis de baisser le taux d'erreurs à $39,4\%$, soit une réduction relative de $12,5\%$ par rapport au taux d'erreurs sur les mots du meilleur des systèmes de reconnaissance (Fiscus, 1997). D'après (Pallet *et al.*, 1998), l'utilisation de ROVER lors de l'évaluation "1998 DARPA Broadcast News evaluation", a permis d'obtenir une réduction relative du taux d'erreurs sur les mots d'environ $21,4\%$ par rapport au taux d'erreurs du meilleur système (les taux d'erreurs sur les mots des différents systèmes variaient entre $13,5\%$ et $25,7\%$. Avec le système ROVER, le taux d'erreurs est tombé à $10,6\%$).

Quelques travaux pour améliorer le système ROVER ont été publiés. Par exemple, (Schwenk et Gauvain, 2000,) propose de modifier ROVER en y injectant des informations provenant d'un modèle de langage. D'autres travaux, tels (Mangu *et al.*, 1999) ou (Stolcke *et al.*, 1997), proposent d'utiliser ROVER sur des graphes de mots ou des listes de n meilleures phrases.

6.2 Proposition

Nous proposons un système de choix de l'hypothèse retenue différent du système ROVER. À la différence de ce dernier, notre système tente de retenir l'hypothèse qui lui semble la plus correcte dans sa globalité, et ne tente pas de choisir mot par mot l'hypothèse finale. De plus, le rejet de l'ensemble des hypothèses est possible, ce qui peut aider à la gestion du dialogue. Pour une application de dialogue oral homme-machine, il est en effet préférable de faire répéter l'utilisateur plutôt que d'interpréter une phrase mal reconnue : les erreurs d'interprétations sémantiques dues à l'analyse d'une phrase erronée perturbent grandement le gestionnaire de dialogue, et peuvent mener le dialogue dans une impasse.

Le système proposé ici est basé sur l'utilisation d'un arbre de décision. Cet arbre traite diverses informations concernant chacune des hypothèses afin de procéder au choix. La construction de cet arbre nécessite un corpus d'apprentissage constitué des hypothèses de chacun des systèmes de reconnaissance pour chaque enregistrement de phrase du corpus.

La construction de l'arbre de décision a pour but de déterminer le système le plus performant pour une situation observée. Sur le corpus d'apprentissage, les choix de l'arbre de décision doivent minimiser le taux d'erreurs sur les mots.

6.2.1 Informations utilisées

Il est possible d'utiliser n'importe quel type d'information disponible. Ces informations doivent aider à la détection des faiblesses ou des points forts de chaque système de reconnaissance. Par exemple, en comparant le nombre de mots des hypothèses de chaque système pour une phrase de référence donnée, la tendance à la suppression de mots d'un des systèmes peut être mise en évidence.

La comparaison des scores acoustiques et linguistiques permet de distinguer certains comportements de chaque système : faiblesse du modèle de langage pour un système dans certains cas, plus grande précision dans d'autres. Pour détecter le manque de connaissances d'un modèle de langage donné pour un cas particulier, il est intéressant de connaître le nombre de *bigrams*, *trigram*s, [1] non observés lors de l'apprentissage du modèle, mais présents dans l'hypothèse retenue par le système qui l'utilise.

Selon le même principe, d'autres informations peuvent être injectées dans le processus. Si les hypothèses de chaque système peuvent être étiquetées[2], cette information peut être utilisée, tout comme la présence ou non de mots de certaines catégories syntaxiques, comme les verbes. De même, pour les modèles à automates, le nombre d'automates ayant participé à l'émission de chacune des hypothèses de reconnaissance qu'ils proposent peut apporter de l'information quant à la pertinence de l'utilisation de ces modèles pour la reconnaissance de la phrase prononcée par l'utilisateur.

L'utilisation de critères comme la présence ou l'absence de verbes permet d'intégrer implicitement des contraintes de cohérence linguistique. Il est rare qu'une phrase soit constituée de plusieurs mots sans qu'un seul ne soit un verbe. Ce type de contraintes peut être représenté dans l'arbre de décision par un noeud dont le parcours passe par des questions portant sur la présence d'un verbe et le nombre de mots.

Au final, c'est l'algorithme de construction de l'arbre de décision qui fait le tri entre les questions pertinentes et les autres. Tout information disponible concernant les hypothèses de reconnaissance susceptible de définir un cas particulier doit être utilisée.

[1] voir le chapitre 4.

[2] par exemple avec les étiquettes [REQUÊTE, QUESTION, RÉPONSE et AUTRE] de la section 5.2.1.

6.2.2 Arbre de décision

L'arbre de décision utilisé pour sélectionner l'hypothèse la pertinente parmi les différentes hypothèses de reconnaissance proposées est un arbre du type décrit dans (Breiman *et al.*, 1984)[3].

6.2.2.1 Questions

Les questions posées par l'arbre de décision portent sur les caractéristiques de l'ensemble d'hypothèses candidates.

Notons $Cand(S_i)$ l'ensemble des hypothèses de reconnaissance candidates pour la phrase S_i effectivement prononcée .

Un ensemble $A_{Cand(S_i)}$ d'attributs est associé à $Cand(S_i)$. Chaque attribut $A_{Cand(S_i)}(x)$ de $A_{Cand(S_i)}$ correspond à une caractéristique de $Cand(S_i)$ et peut prendre une valeur dans un ensemble fini d'éléments.

Par exemple, l'attribut $A_{Cand(S_i)}(Nb\_Mots\_H_x)$, qui renseigne sur le nombre de mots de l'hypothèse H_x de l'ensemble $Cand(S_i)$, est défini de manière à prendre pour valeur uniquement un élément de l'ensemble $Val(Nb\_Mots\_H)$ tel que :

$Val(Nb\_Mots\_H) = \{A_{nmH}, B_{nmH}, C_{nmH}, D_{nmH}, E_{nmH}, F_{nmH}\}$.

Chaque élément de $Val(Nb\_Mots\_H)$ représente un intervalle prédéfini de valeurs.

Supposons que A_{nmH} représente la valeur 1, B_{nmH} la valeur 2, C_{nmH} les valeurs 3 et 4, D_{nmH} les valeurs comprises entre 5 et 8 inclus, E_{nmH} les valeurs comprises entre 9 et 15 inclus, et F_{nmH} les valeurs strictement supérieures à 15. Alors si l'hypothèse H_1 appartenant à $Cand(S_i)$ contient 5 mots, nous avons :

$A_{Cand(S_i)}(Nb\_Mots\_H_1) = D_{nmH}$.

L'utilisation d'attributs ne prenant qu'un nombre fini de valeurs permet de travailler dans un espace discret. Cette nécessité vient de la taille finie, et réduite, du corpus d'apprentissage qui exige la généralisation de certains événements proches.

Cette discrétisation s'avère indispensable, par exemple, pour comparer les scores (acoustique, linguistique, total, ...) d'hypothèses de $Cand(S_i)$. Il serait bien entendu possible d'utiliser des valeurs binaires $VRAI$ ou $FAUX$ comme réponses à l'attribut exprimant le résultat d'une comparaison du type : *"le score acoustique de l'hypothèse H_x est supérieur au score acoustique de l'hypothèse H_y"*. Mais de l'information quantitative serait perdue.

La méthode suivante est préférable : notons $A_{Cand(S_i)}(Sc\_Ac\_H_xH_y)$ l'attribut exprimant le résultat de la comparaison des scores acoustiques de H_x et de H_y. Soient SA_x et SA_y les scores acoustiques respectifs des hypothèses H_x et H_y.

Soit $Val(Sc\_Ac\_H_xH_y) = \{A_{saHH}, B_{saHH}, C_{saHH}, D_{saHH}, E_{saHH}\}$ l'ensemble des valeurs que peut prendre l'attribut $A_{Cand(S_i)}(Sc\_Ac\_H_xH_y)$.

Les éléments $\{A_{saHH}, B_{saHH}, C_{saHH}, D_{saHH}, E_{saHH}\}$ de $Val(Sc\_Ac\_H_xH_y)$ sont définis comme suit :

[3] voir section 5.2.2.

A_{saHH}	: SA_x et SA_y sont égaux,
B_{saHH}	: SA_x est plus grand que SA_y mais sa valeur n'excède pas celle de SA_y de plus de 5%
C_{saHH}	: SA_x est plus de 5% plus grand que SA_y
D_{saHH}	: SA_y est plus grand que SA_x mais sa valeur n'excède pas celle de SA_x de plus de 5%
E_{saHH}	: SA_y est plus de 5% plus grand que SA_x

Ce type de définition permet de savoir si le score acoustique d'une hypothèse est supérieur ou inférieur au score d'une autre, mais aussi de connaître les proportions de cette différence.

Il faut cependant se montrer prudent lors de la définition des valeurs pouvant être affectées à un attribut : l'intervalle de valeurs choisi influe sur la construction de l'arbre de décision. Le nombre de valeurs que peut prendre un attribut est variable : c'est au concepteur du système de prise de décision de définir ce nombre et la signification de ces valeurs.

Les questions associées à l'arbre de décision sont construites à partir des attributs définis à l'avance et des ensembles de valeurs auxquels ils sont associés. Les questions sont du type :

"*est-ce que* $A_{Cand(S_i)}(Sc\_Ac\_H_xH_y) = B_{saHH}$ *?*",

"*est-ce que* $A_{Cand(S_i)}(Nb\_Mots\_H_z) = C_{nmH}$ *?*", ...

L'arbre de décision obtenu sera alors un arbre de décision binaire, puisque les seules réponses possibles à ces questions sont oui ou non.

6.2.2.2 Apprentissage

Le corpus d'apprentissage de l'arbre de décision doit être préparé à partir des informations disponibles.

Préparation des données initiales Le corpus d'apprentissage de l'arbre de décision est construit à partir de phrases de référence S_i associées chacune à un ensemble $Cand(S_i)$ d'hypothèses de reconnaissance. Ces hypothèses sont issues des divers systèmes de reconnaissance disponibles. Pour chaque ensemble $Cand(S_i)$ d'hypothèses, l'ensemble $A_{Cand(S_i)}$ de leurs caractéristiques est déterminée : nombre de mots, scores acoustiques, scores linguistiques, scores globaux, nombre de *bigrams*, *trigrams* ou automates non vus, catégorie de chaque hypothèse (REQUÊTE, QUESTION, RÉPONSE ou AUTRE), présence de verbe, ...

Choix du système de reconnaissance et rejet Le taux d'erreurs sur les mots de chaque hypothèse de $Cand(S_i)$ est calculé : pour chaque phrase de référence du corpus d'apprentissage, le système proposant l'hypothèse la plus correcte est

déterminé. On associe à l'ensemble $A_{Cand(S_i)}$ des caractéristiques des hypothèses le nom de ce système.

Dans le cas où plusieurs systèmes offrent les mêmes performances optimales pour un ensemble $A_{Cand(S_i)}$, c'est le système le plus performant sur l'ensemble du corpus d'apprentissage qui est associé à l'ensemble des caractéristiques d'hypothèses.

Enfin, si chaque hypothèse a un taux d'erreurs supérieur à un seuil prédéfini T_{wer}, aucun système n'est associé à $A_{Cand(S_i)}$: cet ensemble de caractéristiques d'hypothèses est associé à l'étiquette REJET. La valeur du seuil T_{wer}, appelé seuil d'acceptation, permet de régler la tolérance du système de prise de décisions : plus le seuil T_{wer} sera bas, moins il y aura de situations de validation.

Corpus d'apprentissage et classification Le corpus d'apprentissage est formé d'ensembles de caractéristiques. Une étiquette est associée à chaque ensemble $Cand(S_i)$: il s'agit soit du nom d'un système, soit de l'étiquette REJET. Le corpus d'apprentissage se présente sous la forme suivante :

$Cand(S_1) \rightarrow Systeme(X)$,

$Cand(S_2) \rightarrow Systeme(Y)$,

$Cand(S_3) \rightarrow Rejet$,

...

$Cand(S_n) \rightarrow Systeme(Y)$

La construction de l'arbre de décision est effectuée à partir de ce corpus d'apprentissage. Les questions sont posées sur les valeurs prises par les attributs des ensembles de caractéristiques $Cand(S_i)$. L'apprentissage vise à créer un arbre dont les questions regroupent au mieux les ensembles de caractéristiques en fonction du système de reconnaissance qui leur est associé. Pour cela, l'index de Gini est utilisé et l'impureté d'un noeud t_i se mesure avec la formule suivante :

$$G(t_i) = 1 - \sum_{s \in \mathbb{S}} P(s|t_i)^2 \tag{6.2}$$

où $\mathbb{S}$ est l'ensemble des systèmes de reconnaissance en concurrence, et $P(s|t_i)$ la fréquence relative d'apparition de l'étiquette du système de reconnaissance s au niveau du noeud t_i, calculée en fonction du nombre d'apparition de l'ensemble des étiquettes des systèmes de S apparaissant au niveau du noeud t_i. Une telle étiquette apparaît sur un noeud lorsque les questions portant sur l'ensemble des caractéristiques d'hypothèses auquel cette étiquette est associé ont permis d'atteindre ce noeud dans l'arbre. La construction de l'arbre de décision est soumise[4] à la contrainte de minoration de $G(t_i)$.

[4]Voir section 5.2.2.1 pour obtenir plus de détails sur l'apprentissage d'un arbre de décision et sur l'index de Gini.

6.2.2.3 Utilisation

Avant d'utiliser l'arbre de décision pour établir un choix entre les hypothèses proposées par les différents systèmes disponibles[5], les données doivent être préparées. Les caractéristiques sont extraites des hypothèses de reconnaissance et représentées sous la même forme que lors de l'apprentissage de l'arbre : les hypothèses sont traitées selon les informations attendues (par exemple, il faut procéder à un étiquetage syntaxique si la présence d'un verbe est une caractéristique recherchée). L'ensemble des caractéristiques est alors présenté à l'arbre de décision qui détermine quel système de reconnaissance est le plus probablement adapté à la situation (ou détermine un rejet). Le choix de l'arbre dépend des réponses de l'ensemble des caractéristiques d'hypothèses à ses questions.

6.3 Expérimentation

Les expériences sont effectuées sur les données fournies par France Télécom R&D et décrites en section 4.6.1.

Les différents systèmes de reconnaissance en concurrence diffèrent uniquement par le modèle de langage qu'ils utilisent. Le même module acoustique est utilisé pour chaque système et propose le même graphe de mots. C'est l'utilisation de modèles de langage différents pour la recherche de l'hypothèse optimale dans le graphe qui mène à des hypothèses de reconnaissance différentes : cette caractéristique est intéressante dans la mesure où seul le *rescoring* d'un graphe de mots est nécessaire, un autre décodage n'étant pas nécessaire.

Les expériences consistent à construire l'arbre de décision permettant d'établir un choix sur l'hypothèse retenue parmi celles présentées par un ensemble prédéfini de différents systèmes de reconnaissance. Ensuite, les performances de l'arbre avec le même ensemble de systèmes de reconnaissance sont mesurées sur un corpus de test.

6.3.1 Utilisation de la technique dite de *leave-one-out*

Afin de construire le système de décision, il est nécessaire de disposer d'une quantité assez importante de résultats pour chacun des systèmes de reconnaissance.

Nos données expérimentales comportent 1 422 graphes de mots. Les expériences de reconnaissance sont menées à partir de ces graphes. Cet ensemble doit être scindé en deux parties : une partie pour l'apprentissage de l'arbre de décision, et une partie pour le test. Cette quantité de graphes s'avère plutôt faible pour obtenir des résultats fiables. Afin de pallier ce manque de données, nous utilisons la technique dite de *leave-one-out*.

[5] Bien entendu, les systèmes de reconnaissance utilisés sont les mêmes que ceux de l'apprentissage.

La technique du *leave-one-out* consiste à retirer une petite partie des données du corpus initial. Les données restantes sont utilisées pour l'apprentissage et les données retirées sont utilisées pour le test. L'originalité de cette technique vient de la ré-injection des données retirées dans le corpus initial après expérimentation, et de la réitération de ce processus de retrait/expérimentation/ré-injection. Il est ainsi possible de mener à bien les procédures de test sur l'intégralité du corpus, tout en respectant l'indépendance des données de test par rapport aux données d'apprentissage. Bien entendu, il y a quelques règles à suivre afin de conserver cette indépendance :

1. La partie retirée à chaque itération pour les tests doit être totalement disjointe des données déjà retirées et réinjectées.
2. Il faut veiller à l'indépendance de certains paramètres du corpus d'apprentissage et du corpus de test pour chaque itération. Pour nos expériences, le locuteur de chaque phrase prononcée est connu : il semble cohérent d'interdire l'existence de phrases d'un même locuteur dans la partie dédiée à l'apprentissage et dans la partie dédiée au test simultanément. En effet, les expériences pourrait être biaisées par les caractéristiques de ce locuteur (expressions particulières, thème particulier, etc.).

Dans un souci de cohérence avec ces règles, à chaque itération du processus de *leave-one-out*, la partie retirée pour notre expérimentation correspond à l'ensemble des graphes associés à un locuteur[6]. Cela représente 7 itérations pour effectuer les tests sur l'ensemble des graphes.

6.3.2 Rappel des performances des différents modèles de langage présentés dans ce mémoire

Avant de présenter les résultats des expériences portant sur l'utilisation du système de décision présenté ici, il est intéressant de rappeler les performances du système de reconnaissance en fonction du modèle de langage utilisé. Le tableau 6.1 dresse le bilan des performances des modèles *bigrams* présentés dans ce mémoire qui sont utilisés pour l'expérimentation à venir, alors que le tableau 6.2 concerne les modèles *trigrams*. Les quatre type de modèles *bigrams* sont, dans l'ordre : le modèle *bigram* généraliste, les modèles spécialisés construits avec un arbre de classification sémantique et choisis lors de la reconnaissance en utilisant le critère de réduction de la perplexité (ici, nous avons choisi $n = 4$ comme taille de la liste des n meilleures hypothèses car cette valeur donne les meilleurs résultats sur les graphes I avec cette méthode), les modèles spécialisés construits par étiquetage des phrases en quatre catégories prédéfinies et le modèle *bigram* à automates.

Les modèles *trigrams* sont le modèle *trigram* généraliste classique et le modèle *trigram* à automates.

[6] le tableau 4.3 de la section 4.6.1.2 précise la répartition des graphes en fonction du locuteur.

Bigrams	Graphes I	Graphes II
bigram général	*24,61*	*23,63*
bigrams spécialisés (PP, 4-best)	24,58	23,5
bigrams spécialisés (Cat.)	24,16	23,49
bigram + Automates	23,71	22,59

TAB. 6.1 – *Tableau récapitulatif des taux d'erreurs (sur les mots) obtenus sur le corpus de test selon le modèle bigram utilisé présenté dans ce mémoire*

Trigrams	Graphes I	Graphes II
trigram général	*21,79%*	*21%*
trigram + Automates	21,34%	20,04%

TAB. 6.2 – *Tableau récapitulatif des taux d'erreurs (sur les mots) obtenus sur le corpus de test selon le modèle trigram utilisé présenté dans ce mémoire*

6.3.3 Prise de décision pour deux systèmes de reconnaissance mis en concurrence (modèles *trigrams*)

Les premières expériences concernent les deux systèmes de reconnaissance construits à partir des deux modèles *trigrams* utilisés pour les expériences de la section 4.6. Par souci de concision, les résultats présentés proviennent uniquement des expériences menées sur les graphes de type I. Pour les graphes de type II, les résultats sont similaires.

Nous faisons varier le seuil d'acceptation[7] utilisé lors de l'apprentissage de l'arbre de décision entre 0 et 100. En cas de proposition identique lors de l'apprentissage de l'arbre, c'est le système le plus performant globalement qui est choisi. Il s'agit ici du système utilisant le modèle *trigram* à automates.

La figure 6.3 montre l'influence de la variation du seuil d'acceptation sur le nombre d'ensembles d'hypothèses rejetés lors de l'utilisation de l'arbre de décision. Lorsque le seuil d'acceptation est nul, un maximum d'hypothèses sont rejetées, qui correspond à presque un tiers des phrases prononcées ($31,15\%$). Le pourcentage d'ensembles d'hypothèses rejetés diminue avec l'augmentation de la valeur du seuil, pour se stabiliser à environ 5% lorsque le seuil d'acceptation atteint 84. Lorsque le seuil atteint 100, le pourcentage de rejet est inférieur à 1% : presque tous les ensembles d'hypothèses fournissent une hypothèse validée.

La valeur du seuil d'acceptation, utilisé lors de l'apprentissage, est donc un bon paramètre pour régler l'importance du rejet lors de l'utilisation de l'arbre de décision sur le corpus de test.

D'après la figure 6.4, qui permet de comparer les taux d'erreurs sur les mots des hypothèses rejetées[8] et des hypothèses validées en fonction du pourcentage d'en-

[7] voir sous-section 6.2.2.2

[8] L'hypothèse utilisée pour calculer ce taux d'erreurs est l'hypothèse fournie par le système utilisant

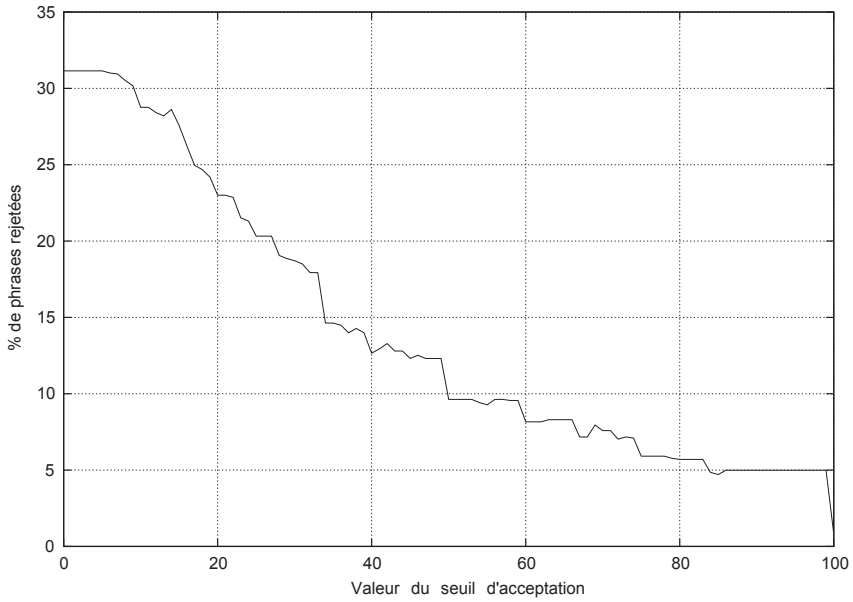


FIG. 6.3 – *Pourcentage d'ensembles d'hypothèses rejetés en fonction de la valeur du seuil d'acceptation utilisé lors de la construction de l'arbre de décision (modèles trigrams, Graphes I)*

semble d'hypothèses rejetés, le taux d'erreurs sur les mots des hypothèses validées est toujours largement inférieur au taux d'erreurs des hypothèses rejetées.

Plus la proportion d'ensembles d'hypothèses rejetés est importante, plus le taux d'erreurs sur les mots des hypothèses validées diminue. Ainsi, ce taux d'erreurs passe de $21,63\%$ lorsque le pourcentage d'hypothèses rejetées est inférieur à 1% pour atteindre $12,41\%$ pour un pourcentage d'ensembles d'hypothèses rejetés de $31,15\%$. Le comportement général du taux d'erreurs sur les mots des hypothèses rejetées est également à la diminution : il oscille entre 51% pour un pourcentage d'hypothèses rejetées de l'ordre de 8% et 34% pour un pourcentage d'hypothèses rejetées supérieur à 30%.

Une étude du même type, mais à partir du taux d'erreurs sur les phrases[9] est illustrée par la figure 6.5.

Les résultats obtenus montrent que quelle que soit la proportion d'ensembles d'hypothèses rejetés, le taux d'erreurs sur les phrases des hypothèses rejetées est de l'ordre de 80% alors que plus la proportion de phrases rejetées augmente, plus le taux d'erreurs sur les phrases des hypothèses validées décroît : il passe de $41,8\%$ à 23%. Cependant, plus il y a d'ensemble d'hypothèses rejetés, moins ces hypothèses contiennent d'erreurs sur les mots.

le modèle le plus performant en général. En l'occurrence, il s'agit du modèle *trigram* à automates stochastiques à états finis.

[9]Il s'agit du pourcentage de phrases contenant au moins une erreur sur un mot par rapport au nombre de phrases total. Il est noté s.e.r, pour *sentence error rate*, en anglais.

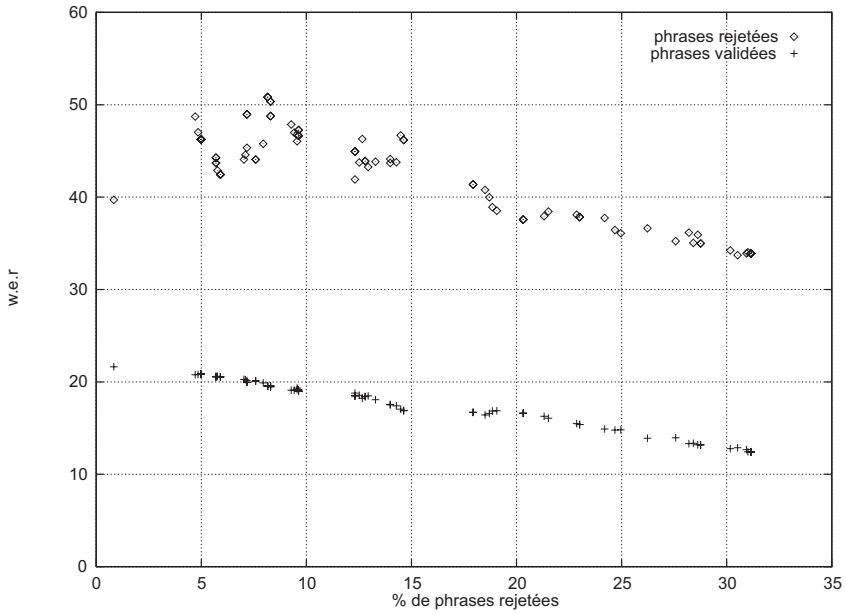


FIG. 6.4 – *Comparaison du taux d'erreurs sur les mots des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'ensemble d'hypothèses rejetés (Modèles trigrams, Graphes I)*

Au sein des hypothèses validées, nous pouvons considérer deux types d'hypothèses :

1. Les hypothèses pour lesquelles les deux systèmes sont d'accord. Appelons-les hypothèses de niveau 1.
2. Les hypothèses pour lesquelles le système de décision doit établir un choix car les deux propositions sont différentes. Appelons-les hypothèses de niveau 2.

La figure 6.6 illustre la proportion d' hypothèses de niveau 1 et de niveau 2 par rapport au nombre total de phrases prononcées et en fonction de la proportion d'hypothèses rejetées[10].

Il est très intéressant de noter que la proportion d'hypothèses de niveau 1 est supérieure à celle des hypothèses de niveau 2 et devient d'autant plus importante que le nombre d'hypothèses rejetées augmente. La figure 6.7 indique clairement que le taux d'erreurs sur les mots des hypothèses de niveau 1 est largement inférieure à celle des hypothèses de niveau 2 : il y a donc une majorité d'hypothèses pour lesquelles le système de décision peut prédire qu'elles sont de bonne qualité.

Ces expériences montrent qu'en acceptant de rejeter 15 à 20% des hypothèses, il est possible de faire baisser le taux global d'erreurs sur les mots de 21,34% (qui est le taux d'erreurs obtenu avec le modèle *trigram* à automates) à environ 17%. Parmi les hypothèses validées, il est possible de définir deux niveaux de confiance :

[10]Il est évidemment possible de considérer l'ensemble des hypothèses rejetées comme l'ensemble des hypothèses de niveau 3.

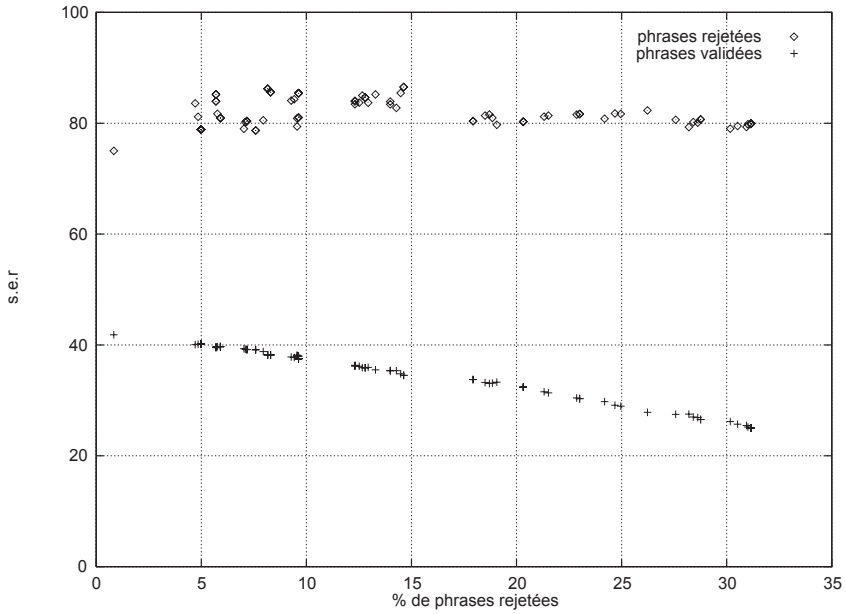


FIG. 6.5 – *Comparaison du taux d'erreurs sur les phrases des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles trigrams, Graphes I)*

1. un niveau de confiance élevé pour les hypothèses de niveau 1, qui représentent environ 80% des hypothèses validées et dont le taux d'erreurs sur les mots est inférieur à 16% pour 15 à 20% d'ensembles d'hypothèses rejetés,
2. un niveau de confiance moyen pour les hypothèses de niveau 2, dont le taux d'erreurs sur les mots varie entre 20,3 et 19% dans les mêmes circonstances.

6.3.4 Prise de décision pour les quatre systèmes de reconnaissance à base de modèles bigrams mis en concurrence

Les mêmes expériences ont été menées avec cette fois quatre systèmes de reconnaissance. Seul le modèle de langage utilisé différencie les systèmes entre eux. Les modèles de langage sont les modèles *bigrams* présentés précédemment dans ce mémoire. Il s'agit :

1. d'un modèle *bigram* classique généraliste,
2. d'un modèle bigram à automates[11], qui est le modèle considéré comme le plus performant et donc le modèle choisi par défaut en cas de propositions identiques de la part de chaque système de reconnaissance pendant la phase de construction de l'arbre,
3. de l'utilisation de modèles *bigrams* spécialisés, construits à partir de connaissance *a priori*[12],

[11] voir chapitre 4.
[12] voir section 5.2.1.

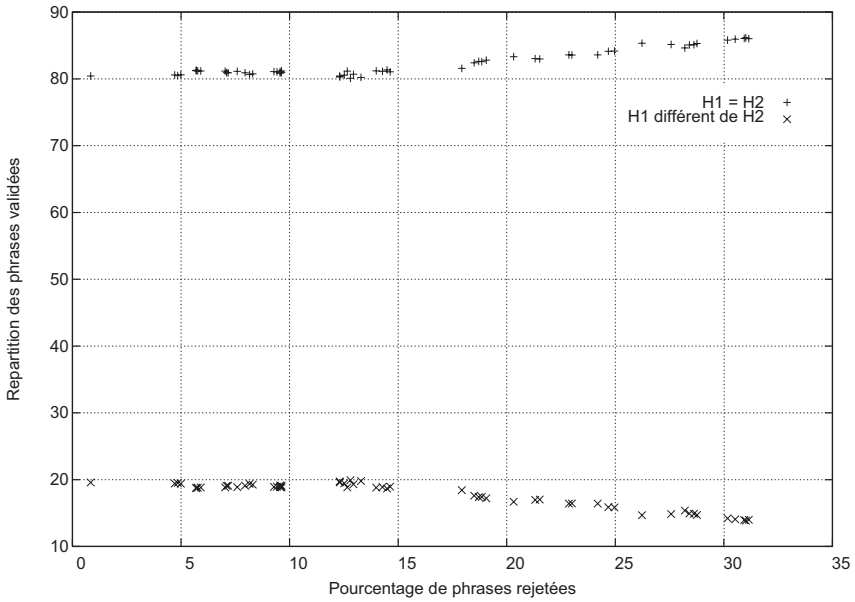


FIG. 6.6 – *Proportion d'hypothèses de niveau* 1 *et d'hypothèses de niveau* 2 *validées en fonction du pourcentage d'hypothèses rejetées (Modèles trigrams, Graphes I)*

4. de l'utilisation de modèles *bigrams* spécialisés, construits à l'aide d'un arbre de classification sémantique[13], et utilisant le critère de réduction de la perplexité sur une liste de 4-best issue d'une première passe à l'aide du modèle *bigram* généraliste pour effectuer la sélection du modèle utilisé[14].

Les résultats obtenus lors des expériences avec les quatre modèles *bigrams* confirment le comportement du système de décision aperçu lors des expériences effectuées à partir des deux modèles *trigrams*.

La proportion d'ensembles d'hypothèses rejetés est légèrement supérieure avec les quatre modèles *bigrams* qu'avec les deux modèles *trigrams*. Comme pour les expériences précédentes, cette proportion diminue en fonction du seuil d'acceptation, ce qu'indique le comportement de la courbe de la figure 6.8.

La figure 6.9 montre que le taux d'erreurs sur les mots des hypothèses rejetées, qui varie entre 60 et 32% en fonction du pourcentage d'ensembles d'hypothèses rejetés, est très largement supérieur au taux d'erreurs sur les mots des phrases validées qui varie entre 24 et 15%.

En terme de taux d'erreurs sur les phrases, la figure 6.10 confirme le taux très élevé d'hypothèses comportant des erreurs parmi les hypothèses rejetées : le taux d'erreurs sur les phrases est constamment supérieur à 70%. La seule exception apparaît lorsque seulement 2,5% des ensembles d'hypothèses est rejeté. Et dans ce cas, le taux d'erreurs sur les phrases reste supérieure à 60%. Plus il y a d'hypothèses

[13]voir section 5.2.2.1.
[14]voir section 5.3.

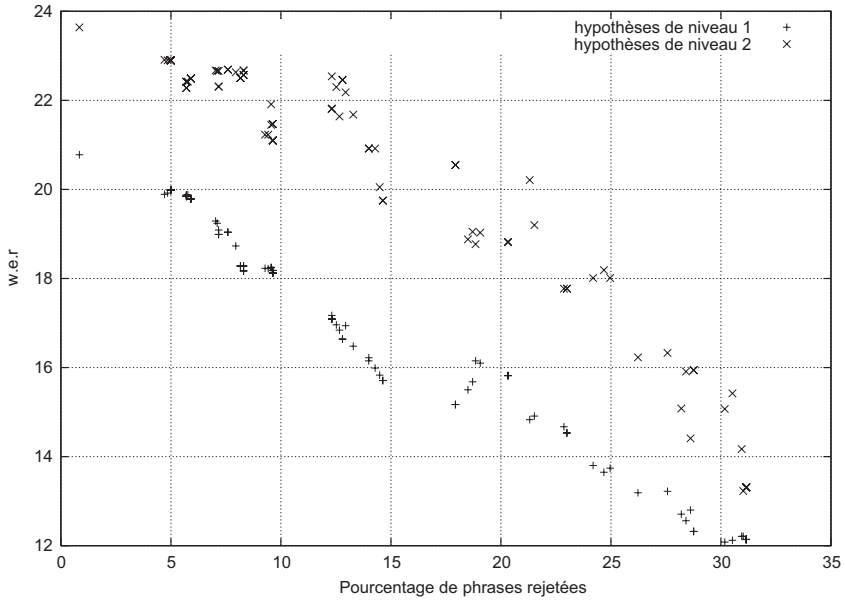


FIG. 6.7 – *Taux d'erreurs sur les mots des phrases de niveau* 1 *(phrases validées et toutes les hypothèses sont identiques) et des phrases de niveau* 2 *(phrases validées, mais un choix a du être fait sur l'hypothèse finale) en fonction du pourcentage de phrases rejetées (Modèles trigrams, Graphes I)*

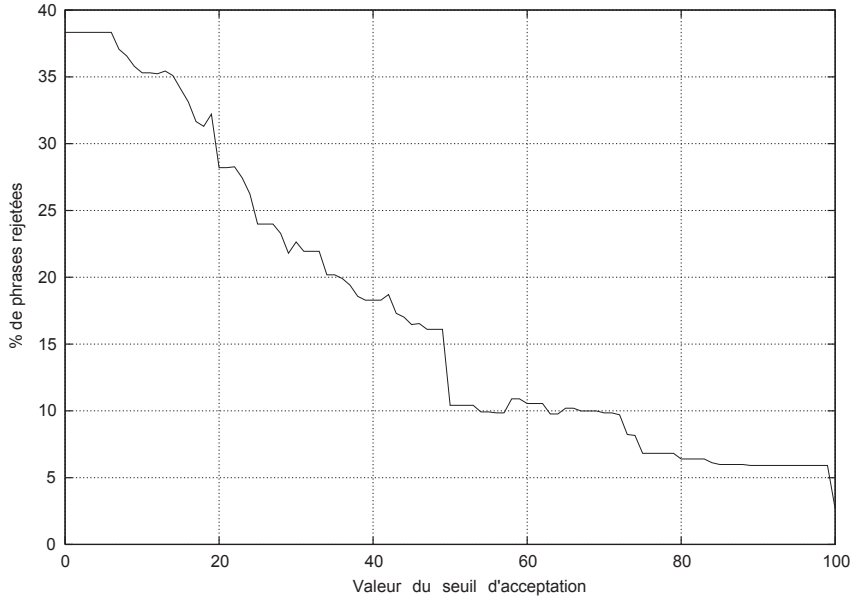


FIG. 6.8 – *Pourcentage d'ensembles d'hypothèses rejetés en fonction de la valeur du seuil d'acceptation utilisé lors de la construction de l'arbre de décision (Modèles bigrams, Graphes I)*

rejetées, plus le taux d'erreurs sur les phrases des hypothèses validées décroît : il passe de 42% pour $2,5\%$ d'hypothèses rejetées à 26% pour 38% de phrases rejetées.

La proportion d'hypothèses de niveau 1 est moins importante que pour les expériences effectuées à l'aide des deux modèles *trigrams*. Malgré cela, les hypothèses de niveau 1 restent majoritaires (voir la figure 6.11) : leur proportion augmente de 64 à 80% quand la proportion d'hypothèses rejetées augmente de $2,5$ à 37%.

Cependant, les résultats des expériences effectuées à partir des quatre modèles *bigrams*, présentés dans la figure 6.12 montrent que le taux d'erreurs sur les mots des hypothèses de niveau 1 est plus faible qu'en utilisant deux modèles *trigrams*. Ceci s'explique facilement : comme les hypothèses de niveau 1 doivent être proposées identiquement par chaque système de reconnaissance, l'utilisation de quatre systèmes au lieu de deux impose des contraintes beaucoup plus strictes. Ces contraintes sont à l'origine du plus petit nombre d'hypothèses de niveau 1, mais aussi de l'augmentation de leur qualité.

Par conséquent, le taux d'erreurs des phrases de niveau 2 est plus mauvais que lors de l'utilisation de deux modèles *trigrams* (expériences précédentes), et ces hypothèses sont plus nombreuses : leur taux d'erreurs sur les mots décroît de 32 à 20% en fonction du nombre d'ensembles d'hypothèses rejetés.

Ces expériences, à partir de deux systèmes utilisant des modèles *trigrams* ou de quatre systèmes utilisant des modèles *bigrams*, indiquent qu'il est possible de distinguer trois niveaux de confiance sur les hypothèses de reconnaissance pour aider le gestionnaire de dialogue. Par contre, il n'a pas été possible de mesurer la qualité du choix d'une hypothèse lorsque les hypothèses proposées par les différents

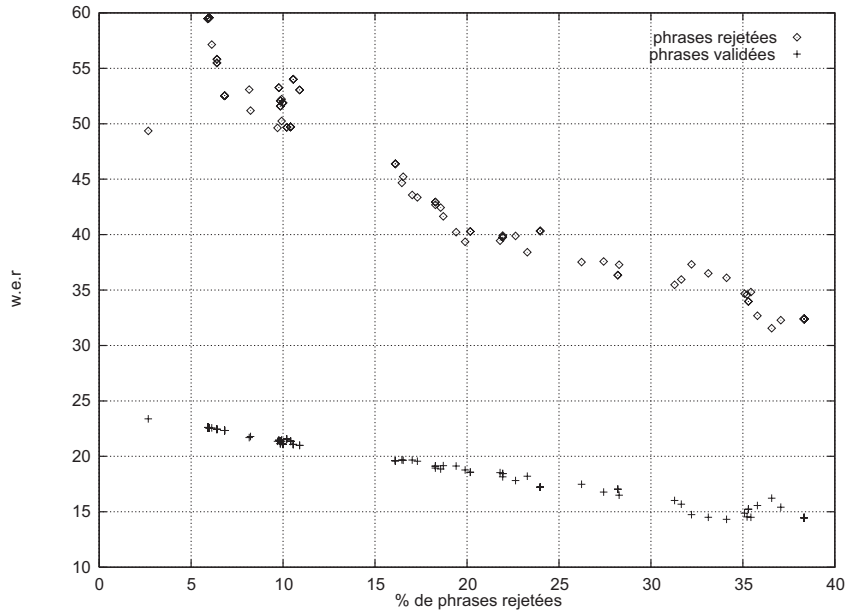


FIG. 6.9 – *Comparaison du taux d'erreurs sur les mots des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)*

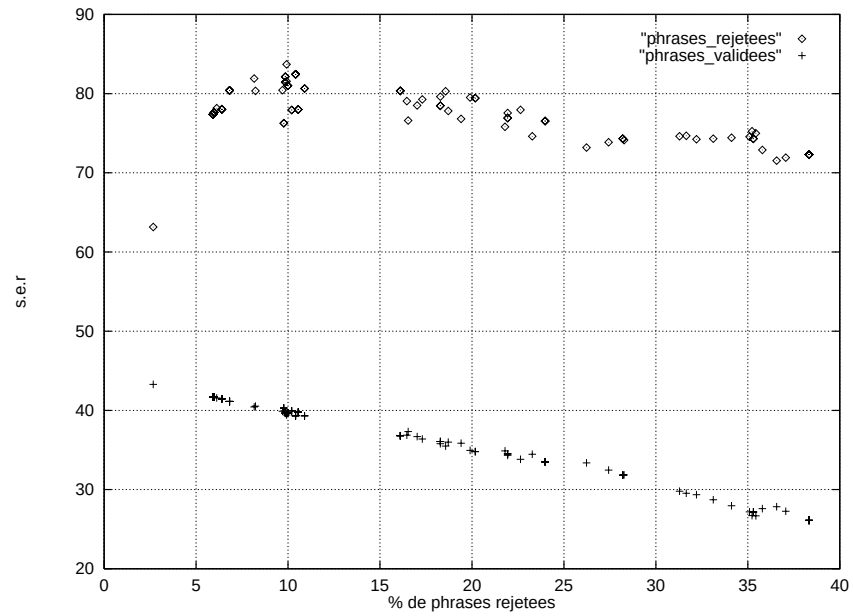


FIG. 6.10 – *Comparaison du taux d'erreurs sur les phrases des hypothèses validées et des hypothèses rejetées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)*

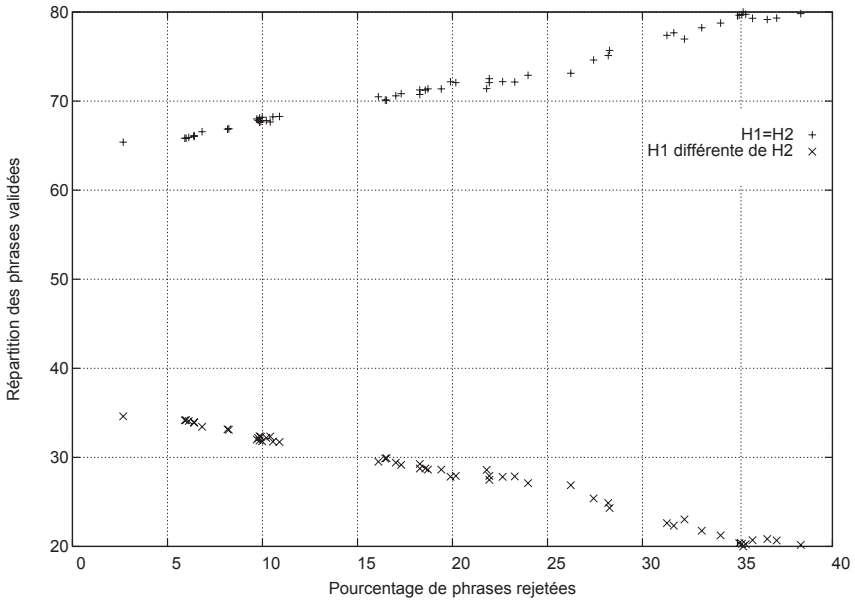


FIG. 6.11 – *Proportion d'hypothèses de niveau* 1 *et d'hypothèses de niveau* 2 *validées en fonction du pourcentage d'hypothèses rejetées (Modèles bigrams, Graphes I)*

systèmes n'étaient pas identiques : dans cette situation, l'hypothèse fournie par le modèle le plus performant est généralement choisie, et généralement avec raison.

La capacité à choisir l'hypothèse la plus pertinente de notre système de décision serait certainement plus facile à mesurer si les différents systèmes de reconnaissance étaient plus différents (modèles acoustiques différents, algorithmes et paramètres de recherche différents, etc.). Dans cette situation, chaque système de reconnaissance a des faiblesses et des qualités propres que notre système de décision cherche à déterminer en analysant les différentes hypothèses proposées en fonction des caractéristiques que lui attribuent chaque système. Dans nos expériences, seules les caractéristiques du modèle de langage influaient différemment sur la recherche de l'hypothèse de reconnaissance la plus vraisemblable pour chacun des systèmes.

6.4 Discussion

Le système de décision présenté ici permet d'établir trois niveaux de confiance sur les hypothèses de reconnaissance, dont un niveau assimilé à un rejet. La prise de décision est effectuée au niveau de la phrase, au contraire du système ROVER qui intervient au niveau des mots. Il serait intéressant de combiner ces deux approches. En effet, les expériences montrent que le système de rejet n'est pas infaillible, et exclut des phrases correctes. De plus, même si certaines hypothèses rejetées comportent des erreurs, elles contiennent également de l'information viable. La combinaison du système ROVER et de l'approche basée sur un arbre de décision comparant les caractéristiques des hypothèses des différents systèmes de

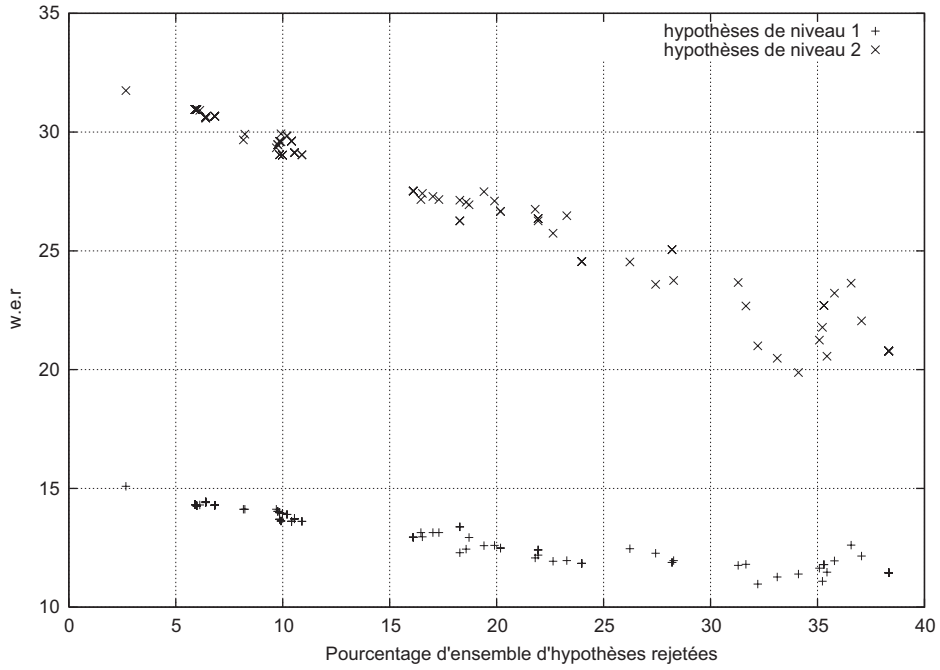


FIG. 6.12 – *Taux d'erreurs sur les mots des phrases de niveau* 1 *(phrases validées et toutes les hypothèses sont identiques) et des phrases de niveau* 2 *(phrases validées, mais un choix a du être fait sur l'hypothèse finale) en fonction du pourcentage de phrases rejetées (Modèles bigrams, Graphes I)*

reconnaissance permettrait de choisir l'hypothèse la plus pertinente, et de déterminer dans cette hypothèse les mots reconnus les plus sûrs.

Au niveau du gestionnaire de dialogue, une telle combinaison autoriserait une conversation intelligente avec l'utilisateur afin d'éclaircir les zones floues de la reconnaissance en s'appuyant sur les mots reconnus avec certitude. Ainsi, au lieu de demander à l'utilisateur de répéter entièrement la phrase qu'il a déjà prononcée, il serait possible de lui poser une question plus directe sur la partie de la phrase mal comprise.

Au niveau du module de reconnaissance, la possibilité de détecter les hypothèses de reconnaissance incorrectes ouvre de nouvelles perspectives. En analysant les hypothèses défectueuses et en vérifiant la consistance syntaxique ou sémantique de ces hypothèses, il est possible d'effectuer une nouvelle phase de *rescoring*. Cette phase de *rescoring* utilise des modèles de langage spécifiques aux problèmes détectés. Le chapitre suivant traite de l'utilisation stratégique de tels modèles de langage.

Chapitre 7

Utilisation stratégique de modèles de langage

Sommaire

7.1 Stratégies 136
7.1.1 Raisonnement sur les résultats et détermination de la stratégie 136
7.1.2 Mesure de consistance 137
7.1.3 Apprentissage par l'exemple *(Explanation-based learning)* . . . 138
7.1.4 Déclencheurs 139
7.2 Modèles stratégiques 139
7.2.1 Modèles *n-grams* adaptés par génération de *n-grams* plausibles 140
7.2.2 Une variante : la dépréciation des *n-grams* peu plausibles . . . 141
7.2.3 Modèles *n-grams* adaptés par augmentation de données 141
7.2.4 Modèles *n-grams* à automates spécifiques 142
7.2.5 Combinaison d'un modèle *n-gram* et d'un arbre de classification sémantique 143
7.3 Expérimentations 145
7.3.1 Définition des contraintes de consistance 146
7.3.2 Bilan 147

Jusqu'à présent, notre étude a porté sur la présentation de nouveaux types de modèles de langage : les modèles *n-grams* à automates et les modèles *n-grams* spécialisés, en particulier les modèles spécialisés hiérarchiques. Aucun de ces modèles, comme aucun des modèles de langage proposés depuis plus de vingt-cinq ans, n'apporte une réduction du taux d'erreurs suffisamment importante pour détrôner l'utilisation des modèles *n-grams* dans les systèmes de reconnaissance. Nous avons également étudié l'utilisation des modèles de langage dans un système de choix/rejet/validation d'hypothèses de reconnaissance. Les résultats expérimentaux de ce système de décision montrent l'utilité d'exploiter les ressources de modèles de langage différents.

À chaque niveau de notre étude, nous avons tenté d'utiliser des connaissances autres que statistiques :

– règles de grammaires pour le modèle hybride,
– expressions régulières ou connaissances linguistiques diverses pour les modèles spécialisés,
– connaissances linguistiques ou statistiques sortant du contexte de la modélisation du langage pour le système de décision.

Dans ce chapitre, nous ouvrons la voie à une nouvelle manière d'utiliser les modèles de langage dans un système de reconnaissance de la parole, en coopération avec l'utilisation de critères de cohérence linguistique. L'étude qui est proposée ici s'inscrit surtout comme une première étape vers des travaux de recherche à venir.

Nous proposons de conserver l'utilisation des modèles *trigrams* puisque ceux-ci donnent de très bons résultats étant donné le coût de leur utilisation, mais en ayant conscience des faiblesses de ces modèles afin d'y pallier. Par exemple, les données d'apprentissage disponibles pour estimer un modèle *trigram* pour une application de dialogue sont rarement suffisantes pour obtenir un modèle robuste. L'utilisation de techniques de lissage, de repli, d'interpolation, diminuent les nuisances de ce manque de données, mais ne les suppriment pas. De plus, les modèles *n-grams* ne modélisent pas de contraintes à longue distance.

Nous proposons d'utiliser les modèles *trigrams* en collaboration avec d'autres modèles de langage (Estève *et al.*, 2002). Chacun de ces modèles de langage n'est utilisé que lorsque son usage est jugé nécessaire. Il ne s'agit pas de proposer de nouveaux modèles dont le comportement général permet d'égaler ou d'améliorer légèrement les performances globales d'un modèle *trigram*, mais d'utiliser des modèles qui, appliqués à des cas particuliers, donnent des résultats bien plus fiables que ceux d'un modèle *trigram*. La somme des gains de performance de chacun de ces modèles stratégiques permet d'élever globalement et substantiellement les performances d'un système de reconnaissance. L'utilisation de ces modèles est déterminée par des critères de cohérence syntaxique et sémantique (De Mori *et al.*, 2002), ou par une mesure de consistance du modèle *trigram*. A l'instar du système présenté dans le chapitre précédent ces critères de consistance autorise le rejet d'hypothèses de reconnaissance jugées incorrectes.

Les modèles de langage stratégiques ne sont donc utilisés que lorsque le modèle *trigram* semble avoir échoué.

7.1 Stratégies

7.1.1 Raisonnement sur les résultats et détermination de la stratégie

7.1.1.1 Raisonnement sur les résultats

Dans le chapitre 6, il a été vu que l'utilisation en parallèle de plusieurs modèles de langage pour la reconnaissance de la parole permettait, en cas de consensus sur l'hypothèse retenue, de préciser que cette hypothèse contenait peu ou pas d'erreurs. Ce type de raisonnement basé sur des résultats a déjà fait l'objet d'une étude dans (Waldinger et Levit, 1974,).

Cette stratégie de mesure de la fiabilité des hypothèses de reconnaissance est utile dans une application de dialogue pour guider le gestionnaire de dialogue en l'informant d'une éventuelle déficience de la reconnaissance : le gestionnaire peut alors adapter les interventions du système afin de pallier les problèmes de la reconnaissance (répétitions, demandes de confirmation de points sensibles, ...).

En fonction du niveau de confiance associé aux hypothèses de reconnaissance, une stratégie de *rescoring* est mise en place. Cette stratégie consiste à utiliser le cas échéant le modèle de langage pouvant traiter efficacement le problème.

7.1.1.2 Détermination de la stratégie

La stratégie de *rescoring* est basée sur deux points fondamentaux :

1. La détection d'une hypothèse de reconnaissance contenant ou risquant de contenir des erreurs.
2. La détection du type d'erreurs rencontrées, ou du type d'ambiguïtés. Ce point est crucial pour le choix du modèle de langage à utiliser.

Le premier point a déjà été étudié dans le chapitre 6. L'utilisation de contraintes de consistance sera abordée, en particulier dans la section 7.3, qui traite des expériences. Pour la détection du type d'erreurs, plusieurs approches sont possibles.

7.1.2 Mesure de consistance

Une première approche consiste à utiliser une mesure de consistance du modèle *trigram* LM qui a généré une première hypothèse de reconnaissance H_1. Cette mesure, notée $CONS(LM)$ est définie ainsi :

$$CONS(LM) = \frac{n_{trigram}(app_{LM} \cap H_1)}{n_{trigram}(H_1)} \tag{7.1}$$

où $n_{trigram}(app_{LM} \cap H_1)$ est le nombre de *trigrams* de l'hypothèse H_1 qui ont été observés au moins une fois dans le corpus d'apprentissage du modèle *trigram* LM, et $n_{trigram}(H_1)$ est le nombre de *trigrams* que contient l'hypothèse H_1. Cette mesure de consistance est comprise entre 0 et 1 : 1 correspond à la mesure de consistance optimale.

Cette mesure de consistance, simple à calculer, donne des résultats très intéressants pour la détection des hypothèses de reconnaissance erronées. Une expérience menée sur le corpus de test à notre disposition[1] a permis d'obtenir les résultats présentés dans le tableau 7.1.

Ces résultats montrent que la mesure de consistance $CONS(LM)$ est très intéressante pour prédire le taux d'erreurs sur les mots des hypothèses de reconnaissance.

[1] voir section 4.6.

$CONS(LM)$	nb de phrases	taux d'erreurs sur les mots
1	1011	$11,72$
$0,75 \leq CONS(LM) < 1$	125	$27,11$
$0,5 \leq CONS(LM) < 0,75$	198	$36,06$
$0,25 \leq CONS(LM) < 0,5$	51	$55,58$
$CONS(LM) < 0,25$	37	$69,24$

TAB. 7.1 – *Taux d'erreurs sur les mots et nombre de phrases en fonction de la mesure de consistance* $CONS(LM)$

Le second intérêt de cette mesure de consistance réside dans le type d'erreurs que cette mesure permet de détecter : il s'agit principalement d'erreurs dues au manque de données d'apprentissage caractérisées par la présence de *trigrams* non vus dans l'hypothèse de reconnaissance, ou bien de graphes de mots ne proposant pas d'hypothèses de vraisemblance acoustique élevée contenant des *trigrams* observés dans le corpus d'apprentissage.

Une solution pour pallier l'absence remarquée de *trigrams* non vus dans l'hypothèse de reconnaissance H_1 est de construire un modèle *trigram* spécialement adapté. La section suivante proposera deux types d'adaptation de modèle de langage pour du *rescoring* basée sur la génération de *n-grams* plausibles ou sur l'augmentation de données.

7.1.3 Apprentissage par l'exemple *(Explanation-based learning)*

Une autre approche, appelée *explanation-based learning* consiste à généraliser des informations tirées d'observations spécifiques (Mitchell *et al.*, 1986). En analysant certaines erreurs produites par le système de reconnaissance sur un corpus de développement, il est possible de proposer une stratégie plus générale pour prévenir l'apparition d'erreurs du même type.

Ainsi, l'apparition incorrecte d'une séquence de mots m dans un contexte particulier (a, b) peut se corriger par le remplacement de m par une autre séquence de mots n. Le remplacement de m par n doit bien entendu répondre à des contraintes strictes. Par exemple, si la séquence de mots m est interdite d'apparition dans le contexte (a, b) car m ne contient pas de verbe, la séquence de mots n susceptible de la remplacer doit contenir un verbe et être consistante syntaxiquement et sémantique avec le contexte (a, b). La figure 7.1 est une représentation sous forme d'arbre logique de cette analyse. Chaque expression associé à un noeud de cet arbre est vraie si les expressions associées à ces fils sont vraies. Par exemple, le noeud racine de la figure 7.1 est associé au remplacement de la séquence de mots m par la séquence de mots n dans le contexte (a, b) : ceci est effectif si la séquence m n'est pas consistante dans le contexte (a, b) alors que la séquence de mots n est consistante en présence du même contexte.

À l'aide de cet arbre, des mots du lexique de l'application, et d'un corpus d'apprentissage, il est possible d'établir manuellement ou automatiquement l'ensemble des

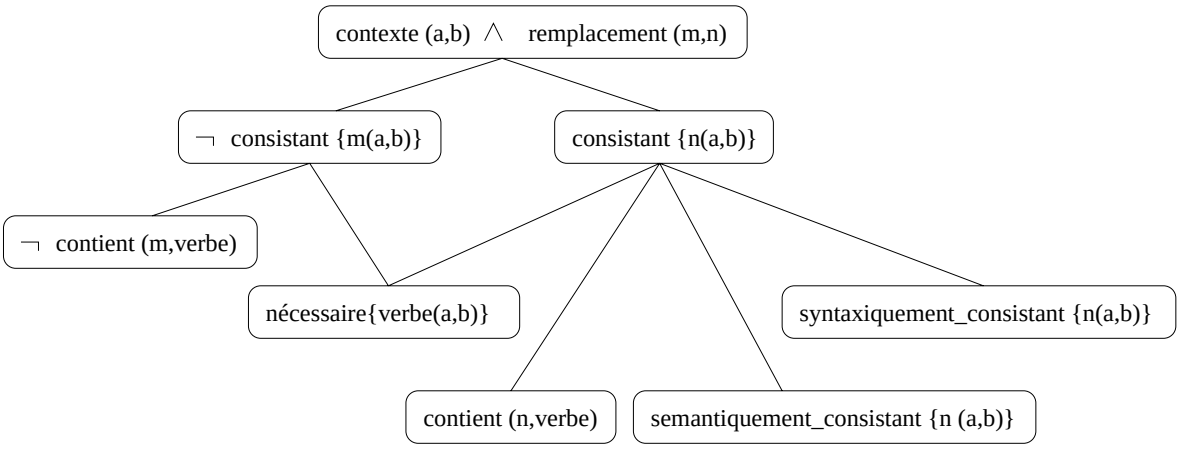


FIG. 7.1 – *Arbre logique de résolution du remplacement de la séquence de mots m par la séquence de mots n dans le contexte (a, b)*

séquences de mots n pouvant remplacer la séquence m.

7.1.4 Déclencheurs

La détection de mots ou de séquences de mots impliqués régulièrement dans l'apparition d'une erreur peut servir de déclencheur à l'utilisation d'un modèle de langage chargé de résoudre le problème lors de la phase de *rescoring*.

C'est le cas par exemple avec les erreurs dues aux mots homophones, ou quasi-homophones, très courantes en français. Les modèles de langage intégrant des contraintes plus longues que les modèles *n-grams* permettent dans certaines situations de désambiguïser les mots homophones. C'est le cas du modèle basé sur la combinaison d'un modèle *n-gram* et d'un arbre de classification sémantique proposé dans la section suivante.

7.2 Modèles stratégiques

Les modèles de langage que nous proposons ici sont destinés à résoudre des problèmes de reconnaissance spécifiques. Nous les présentons brièvement.

La liste de ces modèles n'est pas exhaustive : d'autres modèles de langage peuvent être utilisés qui répondent à des exigences différentes. La principale condition de l'utilisation d'un modèle de langage dit stratégique est sa capacité à résoudre des problèmes devant lesquels les modèles généralistes, et en particulier les modèles *n-grams* classiques, sont peu efficaces.

Une première solution consiste à adapter le modèle *n-gram* lorsque le problème rencontré est un problème de manque de données.

7.2.1 Modèles *n-grams* adaptés par génération de *n-grams* plausibles

En utilisant des informations linguistiques et des informations sémantiques, il est possible de produire ou de donner plus de poids à des *n-grams* plausibles. Ces *n-grams*, qui n'apparaissaient pas dans le corpus d'apprentissage ou qui y sont sous-représentés, sont modifiés au niveau de leurs comptes avant le calcul des probabilités du modèle adapté : la cohérence du modèle est sauvegardée.

Soient les trois *trigrams* suivants dans lesquels le mot x apparaît : $t_g = xw_cw_d$, $t_c = w_gxw_d$ et $t_d = w_gw_cx$. Soit $c(t_q)$ le compte du *trigram* t_q, avec $q \in \{g, d, r\}$. Il est possible de dériver par analogie les comptes des *trigrams* $t'_g = yw_cw_d$, $t'_c = w_gyw_d$ et $t'_d = w_gw_cy$ à partir des comptes des *trigrams* t_g, t_c et t_d si les mots x et y respectent le prédicat $analogue$ défini ainsi :

$$analogue(x, y) = [Syntax(x) = Syntax(y)] \wedge [SemComp(x, y)] \tag{7.2}$$

où $Syntax(x)$ est la classe syntaxique du mot x et $SemComp(x, y)$ indique que x et y sont sémantiquement compatibles. La notion de compatibilité sémantique entre deux mots peut être approchée avec l'utilisation d'une mesure de distance entre deux mots. (Janiszek *et al.*, 2000) propose une mesure de distance euclidienne basée sur la manipulation de vecteurs d'historiques de mots obtenus par décomposition en valeurs singulières, décrite dans (Bellegarda, 1998) ou (Berry, 1992). Concrètement, la proximité de deux mots est calculée à partir de la distribution de leurs historiques. Il est intéressant de noter que ces historiques peuvent être étudiés sur un corpus de données différent de celui de l'application de dialogue visée. En particulier, un corpus beaucoup plus important composé d'articles de presse écrite semble satisfaisant (Janiszek *et al.*, 2000).

Ainsi, nous avons :

$$SemComp(x, y) \text{ vraie } \Leftrightarrow d(x, y) < \delta$$

où $d(x, y)$ est la distance entre les mots x et y définie dans (Janiszek *et al.*, 2000). δ est un seuil fixé à l'avance.

L'acquisition de connaissance par analogie a été présentée dans (Polya, 1954) ou (Brown, 1997). Plus récemment, elle a été utilisée dans (Yvon, 1996) dans le cadre de la synthèse de la parole.

Une fois que l'analogie entre les mots x et y est établie, il y a plusieurs moyens d'initialiser ou de réévaluer les comptes $c(t'_q)$.

Par exemple, il est possible de déterminer les comptes des *trigrams* t'_q les plus bas en fonction des comptes $c(t_q)$ les plus grands :

$$c(t'_g) = \begin{cases} c(yw_cw_d) & \text{si } c(yw_cw_d) > \vartheta \\ \alpha \left\{ \max_{z|analogue(z,y)} c(zw_cw_d) \right\} e^{-d\left(y, \operatorname{argmax}_{z|analogue(z,y)} c(zw_cw_d)\right)} & \text{sinon} \end{cases} \tag{7.3}$$

où d est la mesure de distance proposée dans (Janiszek *et al.*, 2000).

On pourrait croire que l'augmentation des comptes des *trigrams* selon ce principe revient à utiliser des classes de mots, comme pour les modèles *n-classes*. Mais ce n'est pas le cas, puisque les comptes modifiés des mots y analogues à x ne sont pas toujours les mêmes et dépendent du contexte. De plus, les probabilités obtenues sont différentes de celles obtenues avec des modèles à base de classe existants. Elles varient selon la distribution initiale des comptes des *trigrams* et la distance entre les mots.

7.2.2 Une variante : la dépréciation des *n-grams* peu plausibles

Il est également intéressant, à l'inverse de ce qui vient d'être proposé, de pénaliser les comptes de *trigrams* peu plausibles. Cette dépréciation est particulièrement utile pour se débarrasser des *trigrams* apparaissant dans l'hypothèse de première passe grâce à un lissage trop favorable : il arrive ainsi que des *trigrams* non vus dans le corpus d'apprentissage soient favorisés au détriment de *trigrams* observés dans une très faible mesure dans le corpus d'apprentissage.

En faisant l'hypothèse que l'ensemble des *trigrams* d'étiquettes syntaxiques observés sur un corpus d'apprentissage de très grande taille est très proche de l'ensemble des *trigrams* d'étiquette syntaxiques acceptables dans ce langage, il est alors possible de considérer que tout *trigram* d'étiquettes syntaxiques n'apparaissant pas dans cet ensemble est peu plausible. Cette hypothèse semble vraisemblable en raison de la relative stabilité des structures syntaxiques d'un langage.

Lorsqu'une hypothèse de reconnaissance de première passe est associée à une mesure de consistance du modèle de langage $CONS(LM)$ inférieure à 1, l'étiquetage de cette hypothèse à l'aide des noms des classes syntaxiques permet de repérer les *trigrams* de classes syntaxiques peu plausibles. Dès lors, les *trigrams* de mots correspondants sont pénalisés et une phase de *rescoring* est effectuée avec le modèle adapté.

$RCONS(LM)$ est la nouvelle valeur de $CONS(LM)$ du modèle LM sur l'hypothèse obtenue après dépréciation des *trigrams* peu plausibles existant auparavant dans LM.

7.2.3 Modèles *n-grams* adaptés par augmentation de données

En présentant la mesure de distance entre les mots citée précédemment, (Janiszek *et al.*, 2000) propose d'augmenter les comptes des *bigrams* dont les historiques sont proches. La notion de proximité des historiques est définie en fonction de cette distance. Par cette méthode, aucun *bigram* n'a de compte nul et la probabilité de chaque *bigram* est estimée sans lissage.

A partir de calculs matriciels basés sur la décomposition en valeurs singulières (Berry, 1992), (Janiszek *et al.*, 2000) manipule des vecteurs représentant les historiques de mots.

Soit c_{ij} le nombre de fois que le mot w_i a été observé avec l'historique h_j. c_{ij} est le compte de w_i avec le contexte h_j.

Soit d_{jk} la distance entre les vecteurs représentant les historiques h_j et h_k. Soit Γ_j^α l'ensemble des vecteurs représentant les historiques dont les vecteurs sont proches du vecteur représentant h_i.

Soit a_{ij} le compte de w_i avec le contexte h_j obtenu par augmentation de données. Le compte augmenté a_{ij} est obtenu en supposant qu'un historique h_k similaire à h_j participe au compte de la séquence $[(h_j = w_j)w_i]$ en fonction du niveau de similarité existant entre h_j et h_k :

$$a_{ij} = c_{ij} + \sum_{h_k \in \Gamma_j^\alpha} c_{ik} f(d_{jk}) \tag{7.4}$$

où $f(d_{jk}) = 1$ quand $d_{jk} = 0$ et décroît quand d_{jk} augmente. En posant :

$$f(d_{jk}) = e^{-\frac{d_{jk}}{D}} \tag{7.5}$$

$f(d_{jk})$ respecte ces propriétés. D est une valeur prédéfinie qui permet de régler le comportement de l'augmentation de données. Dans nos expériences, $D = 1$.

7.2.4 Modèles *n-grams* à automates spécifiques

L'approche de correction d'erreurs par *explanation-based learning*[2] s'utilise très bien avec des modèles à automates, comme ceux présentés dans le chapitre 4. Ces modèles sont construits à l'aide d'automates spécifiques selon le type d'erreurs détectées.

Supposons que l'erreur suivante ait été détectée sur le corpus de développement : l'hypothèse "la météo Paris" est proposée par le système de reconnaissance alors que la phrase prononcée est "la météo pour Paris".

Une correction est alors nécessaire. Il est possible de généraliser ce type d'erreurs en passant au niveau des classes syntaxiques et en déclarant que si l'hypothèse de reconnaissance propose un nom commun suivi directement d'un nom de ville ou de région alors une erreur est suspectée.

Le modèle de langage généré est alors composé d'un seul automate stochastique créé spécifiquement pour ce type d'incohérence syntaxique. Cet automate est constitué de l'ensemble des séquences de mots apparaissant dans le corpus d'apprentissage et correspondant à la séquence suivante : "N pour XVILLE", où N est un nom commun et XVILLE le nom d'une ville. Dans ce cas, tous les noms de ville du lexique peuvent également être utilisés. La probabilité d'apparition de l'automate dans le contexte attendu d'après l'hypothèse de première passe est augmentée artificiellement en agissant sur les comptes des *trigrams*. Le but de cette manipulation est de forcer l'algorithme de recherche à passer dans l'automate stochastique spécifique

[2] voir section 7.1.3.

lors de la phase de *rescoring* sur le graphe de mots. Seule la séquence de mots erronée qui a été détectée dans la première passe doit être modifiée.

Par contre, si aucune hypothèse acoustique ne propose une séquence de mots appartenant à l'automate, l'hypothèse de reconnaissance obtenue ne contiendra pas de sous-séquence de mots acceptée par l'automate et ne devra pas être retenue.

7.2.5 Combinaison d'un modèle *n-gram* et d'un arbre de classification sémantique

Une des spécificités du français est l'importance du nombre de mots homophones ou quasi-homophones. La désambiguisation des homophones en reconnaissance de la parole est un problème qui peut être résolu par un modèle de langage. Un problème de quasi-homophones récurrent est le problème qui concerne l'accord du nombre : dans un syntagme, il n'y a souvent qu'un seul mot qui permet de différencier acoustiquement le singulier du pluriel. Par exemple, les phrases "Je voudrais savoir si ce serveur concerne Marseille" et "Je voudrais savoir si ces serveurs concernent Marseille" ne diffèrent acoustiquement que pour un seul mot. Or, le choix de ce mot influe directement sur le choix de deux autres mots.

Même si une erreur de nombre n'est pas décisive dans la gestion d'un dialogue oral homme-machine, il entraîne toutefois une légère modification du sens d'une phrase.

Pour résoudre ce type de problèmes, nous proposons d'utiliser un arbre de classification sémantique[3] pour les séquences de mots observées le plus fréquemment dans le corpus de test et pour lesquelles il y a le plus grand nombre de confusions. Lorsque l'hypothèse de reconnaissance de première passe contient une de ces séquences de mots, l'arbre est utilisé.

Comme nous l'avons vu, une confusion existe entre "ce serveur" et "ces serveurs". Un arbre de classification sémantique est construit à partir de toutes les phrases du corpus d'apprentissage qui contiennent les syntagmes "ce serveur" et "ces serveurs". Il est alors possible d'utiliser des contraintes portant sur toute la longueur de la phrase pour effectuer un choix entre ces deux syntagmes.

La figure 7.2 représente un arbre de classification sémantique construit à partir des 271 phrases sur 9 842 du corpus d'apprentissage qui contiennent "ce serveur " ou "ces serveurs". L'arbre n'est pas bâti sur des expressions régulières mais sur des questions portant sur les mots ou classes syntaxiques contenus avant ou après l'indécision que l'arbre doit aider à lever. Pour chaque noeud de l'arbre, la probabilité d'apparition de chaque syntagme est calculée.

Notons $P(y|BE)$ la probabilité donnée par l'arbre de classification sémantique pour l'apparition d'un syntagme en fonction de contextes gauche et droit : y représente le syntagme, B est l'ensemble des mots qui précèdent y dans la phrase et E l'ensemble des mots qui le suivent.

[3] voir section 5.2.2

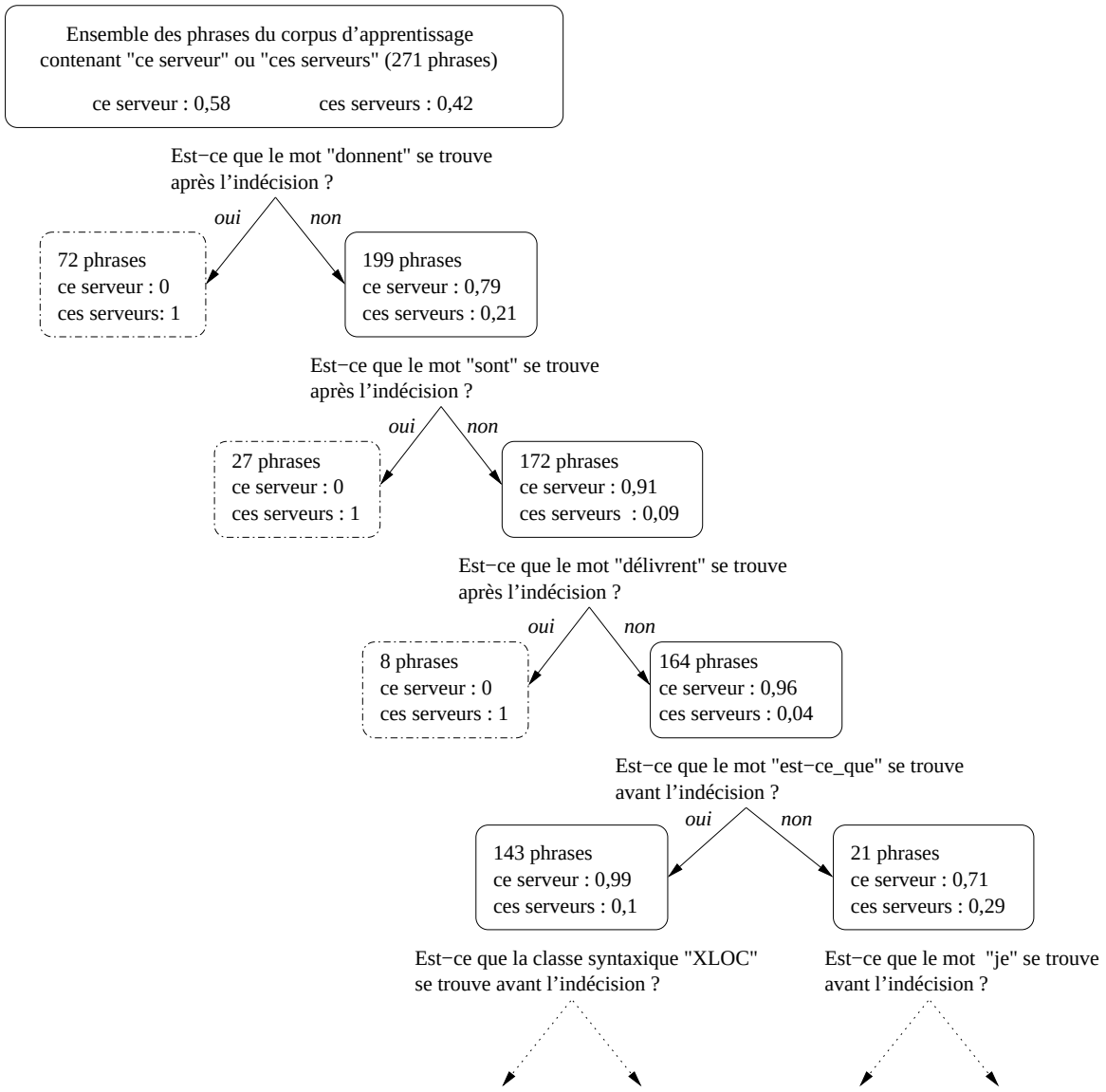


FIG. 7.2 – *Exemple d'arbre de classification utilisé pour la désambiguisation de syntagmes quasi-homophones*

Soit φ le contexte sémantique d'une phrase W : φ correspond à la trame détectée par l'arbre de classification, c'est-à-dire au chemin parcouru dans l'arbre pour aller de la racine à la feuille associée à la phrase.

La probabilité de la phrase W en fonction du contexte φ est donnée par la formule suivante :

$$P(W|\varphi) = P_g(B)\,\{P_s(y|BE)P(s|\varphi) + P_g(y|h)P(g|\varphi)\}\,\{P_s(E|B)P(s|\varphi) + P_g(E|By)P(g|\varphi)\} \tag{7.6}$$

où P_g représente la probabilité donnée par le modèle *n-gram* général et P_s celle donnée par le modèle basé sur l'arbre de classification. h est l'historique utilisé par le modèle *n-gram*.

En pratique, nous utilisons cette probabilité pour décider d'une correction. Pour des questions d'indécisions entre le singulier ou le pluriel, les deux hypothèses sont envisagées et leurs probabilités sont comparées. L'hypothèse retenue est celle ayant la plus grande valeur de probabilité.

7.3 Expérimentations

Les expériences ont été menées en vue de mesurer l'efficacité des contraintes de consistance pour la validation des hypothèses obtenues en première passe d'une stratégie de reconnaissance de la parole.

Nous avons également testé les performances des modèles de langage présentés dans ce chapitre lorsque ces modèles ont été utilisés.

La figure 7.3 décrit la stratégie générale de validation des hypothèses de reconnaissance. Le taux d'erreurs sur les mots pour chaque ensemble de validation est noté $w.e.r$. Le taux d'erreurs sur les mots de la totalité des hypothèses validées depuis le début du processus jusqu'à un niveau donné de la stratégie est noté $G\ w.e.r$.

Le point de départ est constitué par l'ensemble des hypothèses de reconnaissance de première passe. Comme nous avons pu le voir dans les descriptions des expériences des chapitres précédents, elles sont au nombre[4] de $1\,422$. Les expériences sont menées sur les ensembles des graphes de mots les plus élagués, que nous avons appelés graphes de type I.

En utilisant le modèle *trigram* T_g estimé sur le corpus d'apprentissage pour la première passe de reconnaissance, ces hypothèses ont un taux d'erreurs sur les mots de $21,79\%$.

Parallèlement, deux autres modèles de langage sont utilisés afin d'obtenir deux autres hypothèses de reconnaissance pour la même phrase prononcée. Un des modèles est le modèle *bigram* B_a adapté par augmentation de données présenté en section 7.2.3. Le dernier modèle est le modèle *trigram* adapté par génération de

[4] voir section 4.6.1 pour une description détaillée des données d'apprentissage.

trigrams plausibles T_a présenté en section 7.2.2. Notons $H(LM)$ l'hypothèse de reconnaissance obtenue avec un modèle LM.

Les modèles B_a et T_a utilisés seuls ont des taux d'erreurs supérieurs au taux d'erreurs du modèle T_g. Le modèle B_a apporte la robustesse relative d'un modèle *bigram* adapté par augmentation de données, alors que le modèle T_a met en évidence les faiblesses du modèle T_g, en particulier lorsque les hypothèses de reconnaissance obtenues avec T_g comportent des *trigrams* non vus ou sous-estimés sur le corpus d'apprentissage.

7.3.1 Définition des contraintes de consistance

Sur le corpus d'apprentissage, une première règle est obtenue par *explanation-based learning*. Pour la générer, nous sommes partis de l'hypothèse qu'une phrase contenant un nombre n de mots consécutifs sans verbe ne pouvait pas être correcte. Sur le corpus d'apprentissage, il n'a pas été observé de phrase telle que $n > 5$ sans la présence d'un verbe.

La contrainte C_0 est définie à partir de ce constat. Si une hypothèse de reconnaissance contient plus de cinq mots consécutifs sans verbe, alors elle ne satisfait pas C_0.

La contrainte C_1 est définie par le consensus entre les trois modèles de langage T_g, T_a et B_g et par une valeur de la mesure de consistance $CONS(LM)$ égale à 1 pour T_g et T_a (pour B_g, tous les *bigrams* ont un compte non nul et $CONS(B_g)$ est toujours égale à 1) :

$$C_1 : [H(T_g) = H(B_a) = H(T_a)] \wedge [CONS(T_g) = CONS(T_a) = 1]$$

Il y a 870 hypothèses de reconnaissance sur les 1422 (soit $61,2\%$) qui satisfont la contrainte C_1. Le taux d'erreurs sur les mots de ces 870 hypothèses est de $8,54\%$.

En imposant la contrainte C_0 avant la contrainte C_1, il reste 848 hypothèses qui satisfont C_1 pour un taux d'erreurs sur les mots de $7,44\%$.

Définissons les contraintes suivantes :

$$C_2 : [H(T_g) = H(T_a)] \wedge [CONS(T_g) = CONS(T_a) = 1]$$

$$C_3 : [H(T_g) = H(B_a)] \wedge [CONS(T_g) = 1]$$

Ces contraintes sont introduites afin de permettre l'acceptation d'un plus grand nombre d'hypothèses. Les hypothèses qui satisfont C_2 et pas C_1 sont environ 90 pour un taux d'erreurs de 21%. Les hypothèses qui satisfont C_3 sans satisfaire C_1 ni C_2 sont moins d'une dizaine pour un taux d'erreurs sur les mots de 18%.

Les hypothèses qui ne satisfont aucune des contraintes précédentes font l'objet d'un *rescoring* après dépréciation des *trigrams* peu plausibles, comme indiqué en section 7.2.2. La contrainte C_4 est définie ainsi :

$$C_4 : RCONS(T_g) \wedge CONS(T_g) < 1$$

où $RCONS(T_g)$ est la mesure de consistance de T_g sur la nouvelle hypothèse de reconnaissance obtenue après dépréciation des *trigrams* peu plausibles.

45 hypothèses satisfont C_4 sans satisfaire C_1, C_2 ni C_3. Avant la phase *rescoring*, ces hypothèses avaient un taux d'erreurs sur les mots de $40,69\%$. Après correction par *rescoring*, le taux d'erreurs tombe à 32%, soit une diminution relative de $21,3\%$ du taux d'erreurs sur les mots.

La contrainte C_5 est définie ainsi :

$$C_5 : [H(T_g) = H(B_a) = H_g(T_a)] \wedge [CONS(T_g) \geq 0,5]$$

À ce niveau d'acceptation, il y a $1\,015$ hypothèses validées, dont certaines ont été corrigées, pour un taux d'erreurs de $11,7\%$.

Les modèles *n-grams* à automates stochastiques définis par *explanation-based learning* sont alors utilisés en fonction de l'inconsistance détectée. Si une solution est trouvée pendant la phase de rescoring à l'aide de l'automate intégré dans le modèle *n-gram* à cet effet, l'hypothèse est validée. 33 hypothèses sont alors validées, pour un taux d'erreurs sur les mots de $21,31\%$.

Afin d'augmenter encore le nombre d'hypothèses validées, il est toujours possible d'imposer des contraintes moins strictes, comme les contraintes C_7 et C_8 définies ainsi :

$$C_7 : CONS(T_g) > 0,5$$

$$C_8 : H(T_g) = H(B_a) = H(T_a)$$

7.3.2 Bilan

7.3.2.1 Bilan général

Au final 1244 hypothèses (soit $87,5\%$ des hypothèses) ont été validées après avoir été quelquefois modifiées. Ces hypothèses ont un taux d'erreurs global sur les mots de $15,9\%$. Les hypothèses rejetées représentent moins de 13% des hypothèses, pour un taux d'erreur sur les mots de $49,18\%$.

Lorsque le système de choix de l'hypothèse retenue présenté dans le chapitre 6 rejette la même proportion d'hypothèses, le taux d'erreurs sur les mots des hypothèses validées est supérieur à 18%.

La réduction du taux d'erreurs sur les hypothèses validées amenée par la méthode basée sur les contraintes de consistance linguistique provient des corrections produites par les modèles *n-grams* à automates et par le *rescoring* de certaines hypothèses à l'aide de modèles *trigrams* dont certains *trigrams* ont été dépréciés.

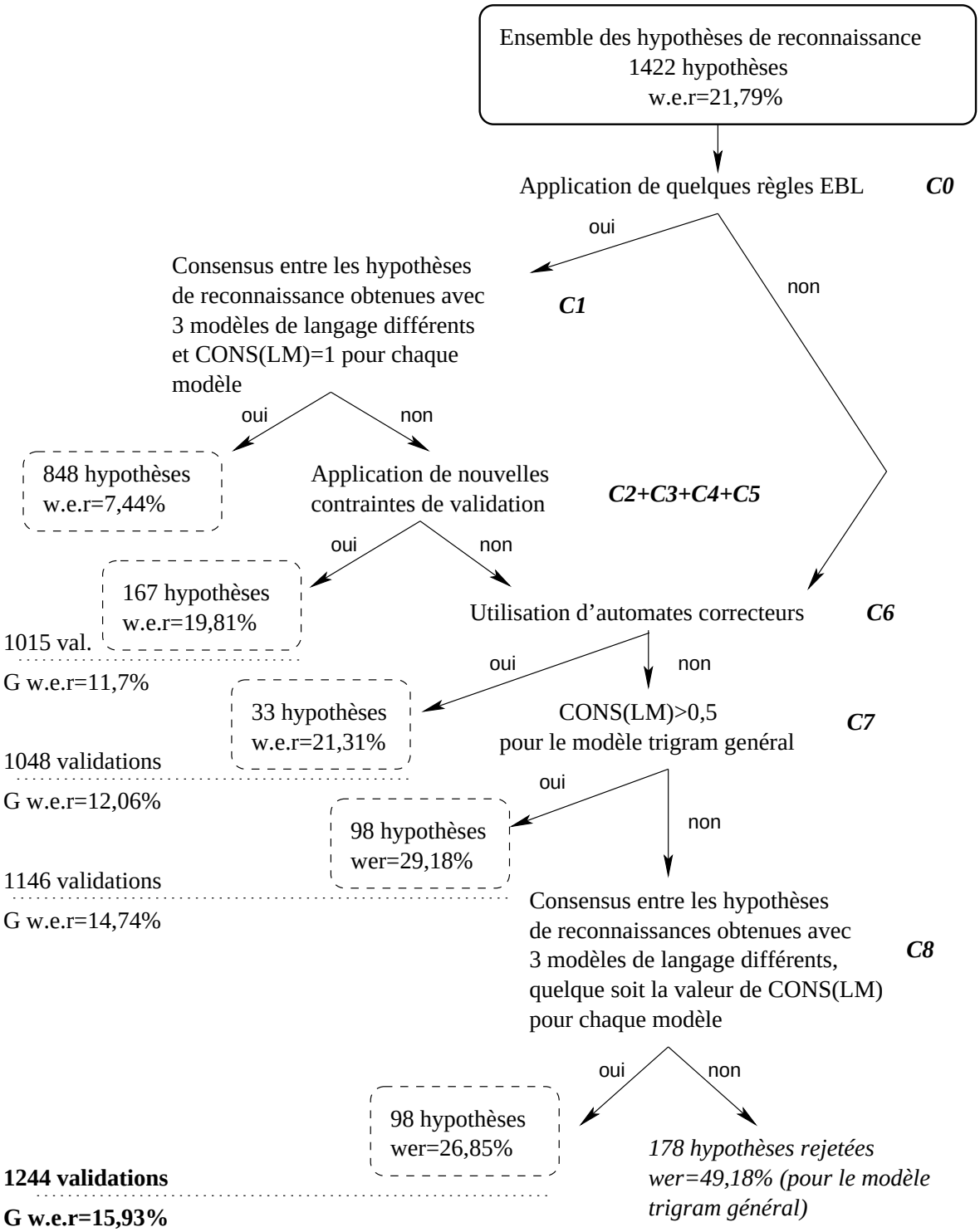


FIG. 7.3 – *Résultats des expériences sur l'application de contraintes de consistance et l'utilisation de divers types de modèles de langage*

Cette réduction s'explique aussi par un meilleur choix des hypothèses rejetées. Le taux d'erreurs des hypothèses rejetées est plus élevé avec l'utilisation de contraintes de consistance qu'avec le système de choix du chapitre 6, ce qui montre la meilleure efficacité des contraintes de consistance comme critères de rejet. Les hypothèses rejetées par l'utilisation de contraintes de consistance ont un taux d'erreurs de $49,18\%$, soit presque 10% de plus que le taux d'erreurs des hypothèses rejetées par l'autre système pour un nombre équivalent de rejets. Quand il rejette 13% des hypothèses, le système de choix du chapitre 6 accuse un taux d'erreurs sur les mots d'environ 45% pour les hypothèses rejetées.

Cependant, il faut noter que pour les expériences du chapitre précédent la technique de leave-one-out a dû être utilisée. Les conditions expérimentales différentes ne permettent pas de valider la comparaison des résultats des deux systèmes.

7.3.2.2 Modèles stratégiques

Chaque modèle stratégique est appliqué sur trop peu de données pour qu'il soit possible de discuter de ses performances. En particulier, les modèles *n-grams* combinés aux arbres de classification sémantique n'ont pas été utilisés pour le calcul des taux d'erreurs. Sur le corpus de test, ces modèles ne s'appliquent que très rarement et ont une influence globale minime sur le taux d'erreurs. Par exemple, le modèle permettant de choisir entre "ce serveur" et "ces serveurs" illustré par la figure 7.2 ne corrige que 13 des 1422 hypothèses du corpus de test (sans ajouter d'erreurs à d'autres hypothèses). Sur ces 13 hypothèses, 9 satisfont la contrainte C_1. Le taux d'erreurs sur les mots de ces 9 phrases passe de $22,2\%$ à $8,3\%$ après application du modèle *n-gram* combiné avec l'arbre de classification sémantique. Le taux d'erreurs de l'ensemble des 848 hypothèses validées par la contrainte C_1 diminue seulement de $7,44\%$ à $7,29\%$.

7.3.2.3 Conclusion

Les expériences qui ont été réalisées montre le potentiel et la faisabilité de l'approche par modèles stratégiques.

Pour obtenir une réduction globale importante du taux d'erreurs, il faudrait un plus grand nombre de modèles comme ceux à base d'arbre de classification ou de modèle *n-gram* à automates. Ces modèles apportent une forte réduction du taux d'erreurs sur les mots dans les cas où ils sont appliqués.

Une qualité de ces modèles est la possibilité de réutiliser un automate ou un arbre de classification sémantique pour une autre application. Au fur et à mesure des développements d'applications, un grand nombre d'automates et d'arbres de classification sémantique pourrait être collecté et réutilisé.

Des travaux complémentaires sont à réaliser afin de compléter cette étude. De nouveaux modèles de langage sont à proposer, différents problèmes de reconnaissance doivent être recensés et généralisés si c'est possible. L'aptitude à ré-exploiter les modèles d'une application à l'autre doit également être explorée.

Chapitre 8

Conclusions et Perspectives

8.1 Bilan

Le principal objet de cette étude a été la recherche de l'amélioration des résultats d'un système de reconnaissance de la parole en agissant sur la modélisation du langage. En particulier, l'accent a été mis sur la diversité des connaissances utilisées pour la création ou l'utilisation d'un modèle de langage. Plus généralement, cette étude s'est appliquée à donner au module de reconnaissance les moyens de fournir à un gestionnaire de dialogue des hypothèses de reconnaissance de meilleure qualité.

L'utilisation de diverses sources de connaissances est bénéfique à un système de reconnaissance de la parole. Ces diverses informations sont exploitables à plusieurs niveaux de la chaîne de traitements où interviennent des modèles de langage probabilistes : au niveau de leur architecture, de leur estimation, de leur utilisation ou du contrôle de la qualité des hypothèses de reconnaissance qu'ils ont aidé à obtenir.

Combinaisons d'automates stochastiques à états finis et d'un modèle *n-gram* : modèle hybride. Dans un premier temps, nous avons proposé un nouveau modèle de langage mixte, combinant un modèle de langage *n-gram* avec des grammaires régulières locales représentées sous la forme d'automates stochastiques à états finis. Ce modèle est intéressant dans la mesure où il se situe entre une approche statistique pure et une approche formelle. Il permet d'intégrer dans un modèle *n-gram* des contraintes sur de longues distances, de modéliser des événements non vus dans le corpus d'apprentissage et d'utiliser des informations linguistiques *a priori*. Il est facilement exploitable pour le décodage d'un graphe de mots dans la mesure où l'algorithme de recherche utilisé avec un modèle *n-gram* classique n'est pas modifié.

Les expériences effectuées avec ce modèle hybride ont montré qu'il aboutissait à de meilleures performances de reconnaissance qu'un modèle de langage *n-gram* : la réduction du taux d'erreurs sur les mots obtenue est statistiquement significative (en relatif, la réduction du taux d'erreurs peut atteindre $4,5\%$).

Utilisation de connaissances pour l'estimation et l'exploitation de modèles *n-grams* : modèles spécialisés. Nous avons poursuivi nos efforts sur la construction de modèles de langage en proposant l'usage de modèles spécialisés pour certains types de phrases apparaissant pendant un dialogue oral homme-machine. Ces modèles sont des modèles *n-grams*. Leur particularité vient de leur estimation sur des sous-corpora spécifiques. Ces sous-corpora sont obtenus par scission du corpus d'apprentissage initial. Ces scissions sont élaborées à l'aide de connaissances *a priori* ou d'informations purement statistiques. En particulier, nous avons proposé une structure hiérarchique des modèles spécialisés en fonction de leur degré de spécialisation.

L'utilisation de ces modèles nécessite une sélection du modèle le plus approprié à la situation du dialogue. Comme aucune information sur l'état du dialogue n'était disponible pour notre étude, nous avons utilisé l'hypothèse de reconnaissance obtenue en première passe de reconnaissance à l'aide d'un modèle de langage généraliste pour effectuer ce choix.

Le potentiel de ces modèles pour la reconnaissance de la parole a été démontré par les résultats de leur expérimentation. Malheureusement, les méthodes de sélection proposées ne sont pas efficaces : elles ne permettent pas d'exploiter correctement ces modèles spécialisés.

Utilisation de connaissances pour le choix, la validation ou le rejet d'hypothèses de reconnaissance : système de décision. Plusieurs modèles de langage peuvent être utilisés parallèlement pour effectuer une recherche d'hypothèse optimale sur un graphe de mots. Diverses hypothèses de reconnaissance sont alors obtenues. La comparaison des caractéristiques de ces hypothèses au sein du système de décision que nous proposons permet de choisir l'hypothèse qui semble la plus pertinente et, le cas échéant, de rejeter la totalité des hypothèses. Certaines caractéristiques des hypothèses de reconnaissance sont issues de traitements divers, comme une analyse grammaticale partielle, ou un étiquetage de la phrase. Les comparaisons de ces caractéristiques sont effectuées à l'aide d'un arbre de décision qui permet d'établir un choix parmi les diverses hypothèses de reconnaissance ou un rejet.

Les expériences ont montré que cette méthode autorise une amélioration de la qualité des hypothèses fournies au gestionnaire de dialogue. La sensibilité du rejet peut être réglée à l'aide d'un seuil d'acceptation : la proportion d'hypothèses rejetées est plus ou moins élevée en fonction de la valeur de ce seuil.

Intégration de connaissances pour l'exploitation de modèles de langage stratégiques. La partie de notre étude consacrée aux modèles de langage stratégiques constitue en quelque sorte la synthèse de nos travaux. Les modèles *n-grams* sont difficilement remplaçables dans les systèmes de reconnaissance. Le gain potentiel de précision obtenu avec d'autres modèles plus évolués ne semble pas satisfaisant au vu de la simplicité d'utilisation des modèles *n-grams*. Nous avons également constaté que certains modèles de langage avaient des performances générales plus

mauvaises que les modèles *n-grams*, mais que ces modèles pouvaient supplanter les modèles *n-grams* dans des cas très particuliers. Ainsi, nous souhaitons conserver le comportement général relativement efficace des modèles *n-grams* et procéder à des ajustements locaux à l'aide de modèles stratégiques dans une phase de *rescoring*. Les modèles dit stratégiques peuvent être des modèles de nature diverse, comme des modèles *n-grams* aux comptes modifiés, des modèles à automates, des modèles utilisant des informations recueillies par un arbre de classification sémantique, etc.

La création de ces modèles provient d'analyses précises des problèmes de reconnaissance les plus fréquents. Leur utilisation est soumise à l'étude de l'hypothèse de reconnaissance obtenue avec le modèle *n-gram* généraliste. Ces diverses analyses permettent de proposer des contraintes de consistance linguistique adaptées aux faiblesses des modèles de langage *n-gram*. Ces contraintes, combinées à d'autres éléments issus de l'hypothèse de première passe, décident du modèle de langage stratégique à mettre en place.

Notre étude sur l'utilisation des modèles stratégiques et des contraintes de consistance ouvre la voie à de futurs travaux. Les premières expériences que nous avons menées ont montré que des contraintes de consistance permettent de distinguer efficacement les hypothèses les plus correctes des plus erronées. L'utilisation des modèles stratégiques permet d'améliorer très fortement la qualité des hypothèses de reconnaissance pour lesquelles ils sont utilisés.

8.2 Perspectives

Coopération entre le module de reconnaissance et le gestionnaire de dialogue. La mise en place d'une coopération effective entre le module de reconnaissance et le gestionnaire de dialogue serait bénéfique à l'ensemble du système. En particulier, il faciliterait la sélection des modèles spécialisés que nous avons proposés. L'utilisation de l'état du dialogue, combinée aux informations contenues dans une hypothèse de reconnaissance de première passe permettrait la mise en place d'un système de sélection dynamique des modèles de langage spécialisés efficace. Étant donné le potentiel dont ces modèles semblent pourvus, leur utilisation en coopération avec le gestionnaire de dialogue s'annonce très intéressante.

Combinaison du système de choix avec un système de type ROVER et des critères de consistance linguistique. La combinaison du système de choix de l'hypothèse que nous proposons, d'un système de vote de type ROVER (dont les votes portent sur les mots et non pas sur l'ensemble des hypothèses de reconnaissance) et de contraintes de consistance linguistique semble également prometteuse. Le fait de connaître les mots douteux d'une hypothèse de reconnaissance validée (ou rejetée) aiderait le gestionnaire de dialogue dans sa conduite de la conversation. Notre système de décision gagnerait également à manipuler des informations sur l'état du dialogue. La coopération entre le module de reconnaissance et le gestionnaire de dialogue permettrait donc non seulement d'améliorer la reconnaissance,

mais aussi de rendre le dialogue plus naturel lorsque des erreurs ou des difficultés de reconnaissance sont détectées.

Ré-exploitation de certaines ressources. Un sujet d'étude intéressant concerne la ré-exploitation des ressources développées pour une application. Le problème se pose par exemple pour les automates stochastiques créés pour le modèle de langage hybride que nous avons présenté. Certains de ces automates sont réutilisables pour de nouvelles applications qui ne disposent pas de données d'apprentissage en quantité importante. Cette pénurie de données d'apprentissage est malheureusement très courante lors de nouveaux développements d'applications de reconnaissance de la parole. La possibilité de réutiliser certaines ressources antérieures semble être une voie intéressante pour compenser le manque de données.

Modèles stratégiques. Nos travaux sur les modèles stratégiques n'en sont qu'à leur début. De nouveaux types d'erreurs récurrentes sont à étudier, de nouveaux modèles sont à créer pour corriger ces erreurs, et de nouveaux critères de consistance sont à proposer. Nos travaux dans ce domaine ont montré qu'il s'agissait d'une approche prometteuse, en particulier quand il est possible de ré-exploiter les ressources utilisées (critères de consistance et modèles stratégiques) pour le développement de nouvelles applications.

Annexe A

Liste des étiquettes syntaxiques utilisées et leur signification

ADV adverbe
ADVNE ne
ADVPAS pas
AFP adjectif féminin pluriel
AFS adjectif féminin singulier
AIND… adjectif indéfini
AMP adjectif masculin pluriel
AMS adjectif masculin singulier
CHIF chiffre ou nombre
COCO conjonction de coordination
COSUB conjonction de subordination
DET… déterminant
DINT… déterminant interrogatif
MOTINC mot inconnu
NFP nom féminin pluriel
NFS nom féminin singulier
NMP nom masculin pluriel
NMS nom masculin singulier
PDEM… pronom démonstratif
PIND… pronom indéfini
PINT… pronom interrogatif
PPER… pronom personnel
PPOBJ… pronom personnel objet

PREF... pronom réfléchi
PREL... pronom relatif
PREP préposition
PREPADE à de
PREPAU au
PREPAUX aux
PREPDES des
PREPDU du
V... verbe
VA... auxiliaire avoir conjugué
VAINF auxiliaire avoir à l'infinitif
VE... auxiliaire être conjugué
VEINF auxiliaire être à l'infinitif
VINF verbe à l'infinitif
VPP... participe passé
VPPRE participe présent
XFAMIL nom propre : nom de famille
XPAY... nom propre : nom de pays
XPREF prénom féminin
XPREM prénom masculin
XSOC nom propre : société
XVILLE nom propre : localité
ZTRM début ou fin de phrase

N.B. : les '...' désignent les différentes sous catégories (féminin pluriel, féminin singulier, masculin pluriel, masculin, singulier pour les déterminants, adjectifs et pronoms et les différentes personnes pour les pronoms personnels et les verbes conjugués).

Annexe B

Grammaire utilisée pour l'analyse grammaticale partielle du corpus d'apprentissage

Les mots du corpus d'apprentissage sont d'abord étiquetés à l'aide des étiquettes syntaxiques présentées dans l' annexe A.

Simplifications

PPERS -> PPER1S

PPERS -> PPER2S

PPERS -> PPER3FS

PPERS -> PPER3MS

PPERP -> PPER1P

PPERP -> PPER2P

PPERP -> PPER3FP

PPERP -> PPER3MP

NFS -> XPAYFS

NMS -> XPAYMS

NFP -> XPAYFP

NMP -> XPAYMP

NFS -> XVILLE

NFS -> XSOC

PREP -> PREPADE

PREP -> PREPDU

PREP -> PREPDES

PREP -> PREPAU

PREP -> PREPAUX

PREF -> PREFFS

PREF -> PREFFP

PREF -> PREFMS

PREF -> PREFMP

PPOBJ -> PPOBJFS

PPOBJ -> PPOBJFP

PPOBJ -> PPOBJMS

PPOBJ -> PPOBJMP

PRELS -> PRELFS

PRELS -> PRELMS

PRELP -> PRELFP

PRELP -> PRELMP

VS -> V1S

VS -> V2S

VS -> V3S

VP -> V1P

VP -> V2P

VP -> V3P

VS -> VA1S

VS -> VA2S

VS -> VA3S

VP -> VA1P

VP -> VA2P

VP -> VA3P

VS -> VE1S

VS -> VE2S

VS -> VE3S

VP -> VE1P

VP -> VE2P

VP -> VE3P

PPS -> VPPFS

PPS -> VPPMS

PPP -> VPPFP

PPP -> VPPMP

ADV -> ADVNE

ADV -> ADVPAS

VINF -> VEINF

VINF -> VAINF

Nucleus adverbial

ADV -> ADV ADV

Nucleus participial

PPS -> ADV PPS

PPS -> PPS ADV

PPP -> PPP ADV

PPP -> ADV PPP

Nucleus adjectival

AFS -> ADV AFS

AMS -> ADV AMS

AFP -> ADV AFP

AMP -> ADV AMP

Juxtaposition d'adjectifs (exemple : un serveur météorologique agricole)

AFS -> AFS AFS

AMS -> AMS AMS

AFP -> AFP AFP

AMP -> AMP AMP

Nucleus nominal

CHIF -> CHIF CHIF

Noms propres (exemple : Monsieur Jean Dupont, Président Bernard Dupond)

NMS -> NMS XPREM XFAMIL

NMS -> NMS XPREM MOTINC

NMS -> NMS XFAMIL

NMF -> NMF XPREM XFAMIL

NMF -> NMF XPREM MOTINC

NMF -> NMF XFAMIL

Apposition : accord en nombre mais pas forcément en genre (c'est le premier qui marque le genre du syntagme)

NFS -> NFS NFS

NFS -> NFS NMS

NFP -> NFP NFP

NFP -> NFP NMP

NMS -> NMS NMS

NMS -> NMS NFS

NMP -> NMP NMP

NMP -> NMP NFP

Epithète : accord en genre et en nombre

NFS -> NFS AFS

NFS -> AFS NFS

NFP -> NFP AFP

NFP -> AFP NFP

NMS -> NMS AMS

NMS -> AMS NMS

NMP -> NMP AMP

NMP -> AMP NMP

Adjectif indefini : accord en genre et en nombre

NFS -> AINDFS NFS

NFP -> AINDFP NFP

NMS -> AINDMS NMS

NMP -> AINDMP NMP

Déterminant : accord en genre et en nombre

NFP -> CHIF NFP

NMP -> CHIF NMP

NFS -> DETFS NFS

NFP -> DETFP NFP

NMS -> DETMS NMS

NMP -> DETMP NMP

Nucleus verbal

Infinitif reflechi

VINF -> PREF VINF

Pronom clitique

VS -> PPOBJ VS

VP -> PPOBJ VP

Pronom réflechi

VS -> PREF VS

VP -> PREF VP

Adverbes dont négation

VS -> ADV VS

VS -> VS ADV

VP -> ADV VP

VP -> VP ADV

Verbe copulé

VS -> VS AFS

VS -> VS AMS

Infinitif

VS -> VS VINF

VP -> VP VINF

Participe passé, temps composé ou passif

VS -> VS PPP

VS -> VS PPS

VP -> VP PPP

VP -> VP PPS

Sujet verbe

P -> PPERP VP

P -> PPERS VS

Bibliographie

A. V. AHO et J. D. ULLMAN. 1972. *The theory of parsing, translation and compiling.* New Jersey, USA : Prentice-Halls.

A.V. AHO et T.G. PETERSON. 1972. A minimum distance error-correcting parser for context-free languages. *SIAM Journal of Computing.*

L.R. BAHL, J.K. BAKER, P.S. COHEN, F. JELINEK, B.L. LEWIS et R.L. MERCER. 1978. Recognition of a continuously read natural corpus. *In : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* Tulsa.

P. BAHL, P. BROWN, P. DE SOUZA et R. MERCER. 1990. A tree-based statistical language model for natural language speech recognition. *Pages 507–514 of :* A. WAIBEL et K.-F. LEE (eds), *Readings in Speech Recognition.* Morgan-Kaufmann.

F. BÉCHET, A. NASR, T. SPRIET et R. DE MORI. 1999. Large span statistical language models : application to homophone disambiguation for large vocabulary speech recognition in French. Budapest, Hongrie.

F. BÉCHET, Y. ESTÈVE et R. DE MORI. 2001a. Modèles de langage hiérarchiques pour les applications de dialogue en parole spontanée. *Pages 327–332 of : 8ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles*, vol. 1. Tours, France.

F. BÉCHET, Y. ESTÈVE et R. DE MORI. 2001b. Tree-based language model dedicated to natural spoken dialog systems. *In : ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition.* Sophia-Antipolis, France.

C. BEAUJARD et M. JARDINO. 1999. Language modeling on automatic word concatenations. *Pages 1563–1566 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 4. Budapest, Hongrie.

J. BELLEGARDA. 1998. Multi-span statistical language modeling for large vocabulary speech recognition. *IEEE Transitions on Speech and Audio Processing,* **SAP-6**(5), 456–467.

J. BELLEGARDA. 2001. An overview of statistical language model adaptation. *Pages 165–174 of : Proceeding of the ISCA ITR-Workshop on Adaptation Methods for Speech Recognition.* Sophia-Antipolis, France.

M.W. BERRY. 1992. Large-scale sparse singular value computations. *Int. J. Supercomp. Appl.*, **6**(1), 13–49.

B. BIGI et R. DE MORI. 2000. A fuzzy decision strategy for topic identification and dynamic selection of language models. *Signal Processing Journal*, **80**(6), 1085–1097.

L. BREIMAN, J. FRIEDMAN, R. OLSHEN et C. STONE. 1984. *Classification and Regression Trees*. Wadsworth.

P. F. BROWN, V. J. DELLA PIETRA, P.V. DE SOUZA, J. C. LAI et R. L. MERCER. 1992. Class-based n-gram models of natural language. *Computationnal Linguistics*, **18**(4), 467–479.

R. BROWN. 1997. *Use of analogy to achieve new expertise*. Tech. rept. AI-TE-403. Artificial Intelligence Laboratory.

H. CERF-DANON et M. EL-BÈZE. 1991. Three different probabilistic language models : comparison and combination. *Pages 297–300 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Toronto, Canada.

M. CETTOLO, R. GRETTER et R. DE MORI. 1998. Search and Generation of Word Hypotheses. *Chap. 9, pages 257–309 of :* R. DE MORI (ed), *Spoken Dialogues with Computers*. Academic Press.

E. CHARNIAK et G. CARROL. 1994. Context-sensitive statistics for improved grammatical language models. *Pages 728–733 of : Proceedings of Twelfth National Conference on Artificial Intelligence*. Seattle, Washington, USA : AAAI Press / MIT Press. Seattle, Washington, USA.

C. CHELBA. 2001. Portability of syntactic structure for language modeling. *In : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, Utah, USA.

C. CHELBA et F. JELINEK. 2000. Structured language modeling. *Computer, Speech and Language*, **14**(4), 283–332.

S. F.AND GOODMAN J. CHEN. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. *Pages 310–318 of :* A. JOSHI et M. PALMER (eds), *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*. San Francisco, USA : Morgan Kaufmann Publishers. San Francisco, USA.

L. CHEN et T. HUANG. 1999. An improved MAP method for language model adaptation. *Pages 1923–1926 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 5. Budapest, Hongrie.

S. F. CHEN et R. ROSENFELD. 2000. A survey of smoothing techniques for ME methods. *IEEE Transactions on Speech and Audio Processing*, **8**(1), 37–50.

N. CHOMSKY. 1957. *Syntactic structures*. The Hague : Mouton.

N. CHOMSKY. 1965. *Aspects of the theory of syntax*. Cambridge : MIT Press.

K. W. CHURCH et W. GALE. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English. *Computer, Speech and Language*, **5**, 19–54.

A. CORAZZA, R. DE MORI, R. GRETTER et G. SATTA. 1991. Computation of probabilities for island-driven parsers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(9), 936–950.

T. M. COVER et J.A. THOMAS. 1991. *Elements of information theory*. Wiley Series in Telecommunications.

G. DAMNATI. 1999. Integration of several information sources for robust class-based statistical language modelling. *Pages 1579–1582 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 4. Budapest, Hongrie.

G. DAMNATI. 2000. *Modèles de langage et classification automatique pour la reconnaissance de la parole continue dans un contexte de dialogue oral homme-machine*. Ph.D. thesis, Université d'Avignon et des Pays de Vaucluse, Avignon, France.

R. DE MORI et M. FEDERICO. 1999. Language model adaptation. *Pages 280–303 of :* KEITH PONTING (ed), *Computational Models of Speech Pattern Processing*. F : Computer and Systems Sciences, vol. 169. NATO ASI Series.

R. DE MORI, Y. ESTÈVE et C. RAYMOND. 2002. On the use of structures in language models for dialogue. *In : Proceedings of the International Conference on Spoken Language Processing*. Denver, Colorado, USA.

S. DELIGNE et F. BIMBOT. 1995. Language modeling by variable length sequences : theoretical formulation and evaluation of multigrams. *In : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Détroit, USA.

S. A. DELLA PIETRA, V. J. DELLA PIETRA, R. MERCER et S. ROUKOS. 1992. Adaptive language model estimation using minimum discrimination estimation. *Pages 633–636 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. San Francisco, Californie, USA.

R. O. DUDA et P.E. HART. 1973. *Pattern Classification and Scene Analysis*. New York, USA : Wiley-Interscience.

M. EL-BÈZE et A.-M. DEROUAULT. 1990. A morphological model for large vocabulary speech recognition. *Pages 577–580 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Albuquerque, New Mexico, USA.

M. EL-BÈZE et T. SPRIET. 1995. Stratégie mixte d'étiquetage syntaxique : Statistiques et connaissances. *Revue TAL*, **36**(1-2), 47–66.

Y. ESTÈVE, F. BÉCHET et R. DE MORI. 2000. Dynamic selection of language models in a dialog system. *Pages 214–217 of : Proceedings of the International Conference on Spoken Language Processing*, vol. 1. Beijing, China.

Y. ESTÈVE, F. BÉCHET, A. NASR et R. DE MORI. 2001. Stochastic finite state automata language model triggered by dialogue states. *Pages 725–728 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 1. Aalborg, Danemark.

Y. ESTÈVE, C. RAYMOND et R. DE MORI. 2002. On the use of structure in language models for dialogue : specific solutions for specific problems. *In : ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments*. Kloster Irseen, Allemagne.

A. FARHAT, J.-F. ISABELLE et D. O'SHAUGNESSY. 1996. Clustering words for statistical language models based on contextual word similarity. *Pages 180–183 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Atlanta, USA.

M. FEDERICO. 1996. Bayesian estimation methods for n-gram language model adaptation. *Pages 240–243 of : Proceedings of the International Conference on Spoken Language Processing*. Philadelphia, Pennsylvanie, USA.

M. FEDERICO et R. DE MORI. 1998a. Interpolation and backing-off LMs. *Chap. 7, pages 210–219 of :* R. DE MORI (ed), *Spoken Dialogue with Computers*. Academic Press.

M. FEDERICO et R. DE MORI. 1998b. Language Modelling. *Chap. 7, pages 204–210 of :* R. DE MORI (ed), *Spoken Dialogue with Computers*. Academic Press.

J. G. FISCUS. 1997. A post-processing system to yield reduced error word rates : Recognizer Ouput Voting Error Reduction. *Pages 347–354 of : Proceedings IEEE Workshop on Automatic Speech Recognition and Understanding*. Santa Barbara, Californie, USA.

D. FOHR, J.-P. HATON, J.-F. MARI, K. SMAILI et I. ZITOUNI. 1997. Towards an oral interface for data entry : the MAUD system. *In : Third ERCIM Workshop on "User Interface for All"*. Obernai, France.

J.-L. GAUVAIN, L. LAMEL et G. ADDA. 2000. The LIMSI 1999 Hub-4E Transcription System. *In : Proceedings DARPA Speech Transcription Workshop*. Gaithersburg, Maryland, USA.

S. GELFAND, C. RAVISHANKAR et E. DELP. 1991. An iterative growing and pruning algorithm for classification tree design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(2), 163–174.

R. C. GONZALES et M. G. THOMASON. 1978. *Syntactic pattern recognition*. Massachussets, USA : Addison-Wesley Publishing Company.

IRVING J. GOOD. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**(3), 237–264.

P. HART, N. NILSSON et B. RAPHAEL. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*.

D. JANISZEK, F. BÉCHET et R. DE MORI. 2000. Integrating MAP and linear transformation for language model adaptation. *In : Proceedings of the International Conference on Spoken Language Processing*. Beijing, China.

E. T. JAYNES. 1957. Information theory and statistical mechanics. *Physic Reviews*, **106**(4), 620–630.

F. JELINEK. 1990. Self-organized language modeling for speech recognition. *Pages 450–505 of :* A. WAIBEL et K. LEE (eds), *Readings in Speech Recognition*. Los Altos, California, USA : Morgan Kaufmann Publishers.

F. JELINEK. 1991. Up from trigrams ! *In : Proceedings of European Conference on Speech Communication and Technology*.

F. JELINEK. 1997. Elements of Information Theory. *Pages 113–135 of : Statistical Methods for Speech Recognition*. Cambridge, Massachusetts, USA : MIT Press.

S. M. KATZ. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**(3), 400–401.

A. KELLNER. 1998. Initial language models for spoken dialogue systems. *Pages 185–188 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Seattle, Washington, USA.

R. KUHN et R. DE MORI. 1990. A cache-based natural language method for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(6), 570–582.

R. KUHN et R. DE MORI. 1995. The application of semantic classifciation trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(5), 449–460.

R. KUHN et R. DE MORI. 1998. Sentence Interpretation. *Chap. 14, pages 485–522 of :* R. DE MORI (ed), *Spoken Dialogue with Computers*. Academic Press.

R. LAU, R. ROSENFELD et S. ROUKOS. 1993. Trigger-based language models : a maximum entropy approach. *Pages 45–48 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2. Minneapolis, Minnesota, USA.

G. MALTESE, P. BRAVERTI, H. CRÉPY, B. J. GRAINGER, M. HERZOG et F. PALOU. 2001. Combining word- and class- based language models : A comparative study in several languages using automatic and manual word-clustering techniques. *Pages 21–24 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 1. Aalborg, Danemark.

L. MANGU, E. BRILL et A. STOLCKE. 1999. Finding consensus among words : lattice-based word error minimization. *Pages 495–498 of : Proceedings of European Conference on Speech Communication and Technology*. Budapest, Hongrie.

H. MASATAKI, Y. SAGISAKA et T. TAWAHARA. 1997. Task adaptation using MAP estimation in n-gram language model. *Pages 783–786 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Munich, Allemagne.

T. MITCHELL, R. KELLER et S. KEDAR-CABELLI. 1986. Explanation-based generalization : an unified view. *Machine Learning*, **1**, 47–80.

A. NASR, Y. ESTÈVE, F. BÉCHET, T. SPRIET et R. DE MORI. 1999. A language model combining n-grams and stochastic finite state automata. *Pages 2175–2178 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 5. Budapest, Hongrie.

H. NEY et U. ESSEN. 1991. On smoothing techniques for bigram-based natural language modelling. *Pages 825–828 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Toronto, Canada.

H. NEY, D. MERGEL, A. NOLL et A. PAESELER. 1992. Data driven search organization for continuous speech recognition. *IEEE Transactions on Signal Processing*, **40**(2), 272–281.

D. S. PALLET, J. G. FISCUS, J. S. GAROFOLO, A. MARTIN et M. PRZYBOCKI. 1998. 1998 broadcast news benchmark test results : English and non-English word error rate performance measures. *In : DARPA Broadcast News Workshop*. Hernon, VA, USA.

G. POLYA. 1954. Induction and analogy in mathematics. *In : Mathematics and plausible reasoning*. Princetown University Press.

G. PULLUM et G. GAZDAR. 1982. Natural languages and context-free grammars. *Linguistics and Philosophy*, **4**, 471–504.

G. RICCARDI et A. L. GORIN. 2000. Stochastic language adaptation over time and state in natural spoken dialogue systems. *IEEE Transactions on Speech and Audio Processing*, **8**(1), 3–10.

G. RICCARDI, E. BOCCHIERI et R. PIERACCINI. 1995. Non deterministic stochastic language models for speech recognition. *Pages 237–240 of : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Atlanta, USA.

G. RICCARDI, R. PIERACCINI et E. BOCCHIERI. 1996. Stochastic automata for language modeling. *Computer and Language*, **10**, 265–293.

K. RIES, F. DAG BUO et Y.-Y. WANG. 1995. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. *Pages 193–196 of : Improved language modeling by unsupervised acquisition of structure*, vol. 1. Détroit, USA.

R. ROSENFELD. 1996. A maximum entropy approach to adaptative statistical language modeling. *Computer Speech and Language*, **10**, 187–228.

D. SADEK et R. DE MORI. 1998. Dialogue systems. *Chap. 15, pages 523–561 of :* R. DE MORI (ed), *Spoken Dialogue with Computers*. Academic Press.

D. SADEK, A. FERRIEUX, A. COZANNET, P. BRETIER, F. PANAGET et J. SIMONIN. 1996. Effective human-computer cooperative spoken dialogue : the AGS demonstrator. *Pages 546–549 of : Proceedings of the International Conference on Spoken Language Processing*. Philadelphia, Pennsylvanie, USA.

A. SALOMAA. 1969. Probabilistic and weighted grammars. *Information and Control*, **15**, 529–544.

G. SAPORTA. 1990. *Probabilités, analyse des données et statistique*. Editions Technip.

H. SCHWENK et J.-L. GAUVAIN. 2000. Combining Multiple Speech Recognizers using Voting and Language Model Information. *Pages 915–918 of : Proceedings of the International Conference on Spoken Language Processing*, vol. 2. Beijing, Chine.

C. SORIN et R. DE MORI. 1998. *Spoken Dialogue with Computers*. Academic Press. Pages 563–582.

B. SOUVIGNIER, A. KELLNER, B. RUEBER, H. SCHRAMM et F. SEIDE. 2000. The thoughtful Elephant : strategies for spoken dialogue system. *IEEE Transactions on Speech and Audio Processing*, **8**(1), 51–62.

T. SPRIET et M. EL-BÈZE. 1998. Introduction of rules into a stochastic approach for language modelling. *Pages 350–355 of :* K. PONTING (ed), *Computational Models of Speech Pattern Processing*, vol. 169. NATO ASI Series F.

A. STOLCKE, Y. KÖNIG et M. WEINTRAUB. 1997. Explicit word error minimization in n-best list rescoring. *Pages 163–165 of : Proceedings of European Conference on Speech Communication and Technology*. Rhodes, Grèce.

V. N. VAPNIK. 1982. *Estimation of Dependences Based on Empirical Data*. New York, USA : Springer-Verlag.

D. VAUFREYDAZ, L. BESACIER, C. BERGAMINI et R. LAMY. 2001. From generic to task-oriented speech recognition : French experience in the NESPOLE ! European project. *Pages 179–182 of : Proceeding of the ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*. Sophia-Antipolis, France.

R.J. WALDINGER et K.N. LEVIT. 1974. Reasoning about programs. *Artificial Intelligence*, **5**, 235–316.

Y. WANG, M. MAHAJAN et X. HUANG. 2000. A Unified Context-Free Grammar And N-Gram Model for Spoken Language Processing. *In : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Istanbul, Turkey.

I. H. WITTEN et T. C. BELL. 1991. The zero-frequency problem : estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, **37**(4), 1085–1094.

P. C. WOODLAND, M. J. F. GALES, D. PYE et S. J. YOUNG. 1997. The development of the 1996 HTK broadcast news transcription system. *Pages 73–78 of : Proceedings DARPA Speech Recognition Workshop*. Chantilly, Virginie, USA.

H. YOUGNER. 1967. Recognition and parsing of context-free languages on time N^3. *Information & Control*, **10**, 198–208.

F. YVON. 1996. *Prononcer par analogie : motivation, formalisation et évaluation*. Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications.

X. ZHU et R. ROSENFELD. 2001. Improving trigram language modeling with the World Wide Web. *In : Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, Utah, USA.

I. ZITOUNI, K. SMAILI, J.-P. HATON, S. DELIGNE et F. BIMBOT. 1998. A comparative study between polyclass and multiclass models. *In : Proceedings of the International Conference on Spoken Language Processing*. Sydney, Australia.

I. ZITOUNI, K. SMAILI et J.-P. HATON. 2001. Statistical language model based on a hierarchical approach : MC_{η}^{ν}. *Pages 29–32 of : Proceedings of European Conference on Speech Communication and Technology*, vol. 1. Aalborg, Danemark.

Références bibliographiques personnelles

Conférences internationales

NASR, A., ESTÈVE, Y., BÉCHET, F., SPRIET T., & DE MORI, R. 1999. A language model combining *n-grams* and stochastic finite state automata, *pages 2175-2178, Proceedings of European Conference on Speech Communication and Technology*, vol.5.

ESTÈVE, Y., BÉCHET, F., & DE MORI, R. 2000. Dynamic selection of language models in a dialog system, pages 214-217, *Proceedings of the International Conference on Spoken Language Processing*, vol.1.

ESTÈVE, Y., BÉCHET F., NASR, A., & DE MORI, R. 2001. Stochastic finite state automata language model triggered by dialogue states, pages 725-728, *Proceedings of European Conference on Speech Communication and Technology*, vol.1.

DE MORI, R., ESTÈVE, Y., & RAYMOND, C. 2002. On the use of structures in language models for dialogue, *Proceedings of the International Conference on Spoken Language Processing.*

Conférences nationales

ESTÈVE, Y., BÉCHET, F., & DE MORI, R. 2000. Sélection dynamique de modèles de langage dans une application de dialogue, pages 185-188, *XXIIIème Journées d'Etudes sur la Parole.*

BÉCHET, F., ESTÈVE Y., & DE MORI, R. 2001. Modèles de langage hiérarchiques pour les applications de dialogue en parole spontanée, *pages 327-332, 8ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles*, vol. 1.

Workshops internationaux

BÉCHET F., ESTÈVE, Y., & DE MORI, R. 2001. Tree-based language model dedicated to natural spoken dialog systems, *ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition*.

ESTÈVE, Y., RAYMOND, C., & DE MORI, R. 2002. On the use of structure in language models for dialogue : specific solutions for specific problems, *ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments*.