Académie d'Aix-Marseille
Université d'Avignon et des Pays de Vaucluse

# THÈSE

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

Spécialité : Informatique

École doctorale : Mathématiques et Informatique

# ÉLÉMENTS POUR UN SYSTÈME DE RECONNAISSANCE AUTOMATIQUE DE LA PAROLE CONTINUE DU FRANÇAIS

par

Stéphane IGOUNET

Soutenue le 20 juillet 1998, devant le jury composé de :

| | |
|---|---|
| M. Henri MÉLONI | Président |
| M. Jean-Marc DOLMAZON | Rapporteur |
| M. Jean-Paul HATON | Rapporteur |
| M. Frédéric BIMBOT | Examinateur |
| M. Pascal NOCERA | Examinateur |
| M. Marc EL-BÈZE | Directeur de Thèse |

# Remerciements

Pour l'intérêt qu'ils ont témoigné à mes activités de recherche, je tiens à exprimer mes remerciements et ma sincère reconnaissance aux membres du jury :

À Jean-Marc Dolmazon, professeur à l'INPG, et Jean-Paul Haton, professeur à l'Université de Nancy I, qui ont tous deux accepté d'être les rapporteurs de cette thèse ;

À Marc El-Bèze, professeur à l'Université d'Avignon et des Pays de Vaucluse, qui a dirigé mes recherches ;

À Henri Méloni, professeur et président de l'Université d'Avignon et des Pays de Vaucluse, qui malgré ses nombreuses responsabilités, a pris le temps d'examiner ce travail ;

À Frédéric Bimbot, chargé de recherche à l'IRISA, et Pascal Nocera, maître de conférence à l'Université d'Avignon et des Pays de Vaucluse, qui ont eu l'amabilité de faire partie de ce jury.

# Sommaire

Remerciements ....................1

Sommaire ....................3

Avant-propos ....................6

Introduction ....................8

Bases théoriques ....................11
  1.1. Caractérisation du problème ....................12
  1.2. Méthodologies de reconnaissance ....................14
  1.3. L'approche globale probabiliste ....................19
  1.4. Modélisation Markovienne des probabilités acoustiques et linguistiques ....................21
  1.5. Stratégies de décodage en parole continue ....................30
  1.6. Conclusion ....................43

Réalisation d'un système de reconnaissance : Décodage Acoustico-Phonétique ....................44
  2.1. Cadre de l'étude ....................45
  2.2. Outils de développement ....................47
  2.3. Bases de données ....................49
  2.4. Module de phonétisation ....................51
  2.5. Paramétrisation des données acoustiques ....................60
  2.6. Alignement des données acoustiques et phonétiques ....................60
  2.7. Modèles de Markov ....................62
  2.8. Apprentissage ....................65
  2.9. Reconnaissances et résultats ....................69
  2.10. Conclusion ....................78

Réalisation d'un système de reconnaissance : Accès Lexical et Modélisation du Langage ....................80
  3.1. Présentation d'un SRAP de référence ....................81
  3.2. Reconnaissance ....................82
  3.3. Reconnaissance et stratégies de décodage ....................86
  3.4. Conclusion ....................96

Modèles de langage : Extraction de Sous-Modèles de Langage ....................99
  4.1. Introduction ....................100
  4.2. Position du problème ....................100
  4.3. Extraction de Sous-Modèles de Langage ....................101
  4.4. Expériences et résultats ....................106
  4.5. Conclusion ....................109

Conclusions et perspectives........................................................................110

Annexes ......................................................................................................114
  Table des abréviations ...........................................................................115
  Alphabet Phonétique International .........................................................117
  Classes syntaxiques ................................................................................119
  Résultats de reconnaissance du DAP.......................................................122

Liste des publications ..................................................................................126

Bibliographie ................................................................................................128

Table des figures et des tableaux .................................................................139

Table des matières .......................................................................................142

Index ............................................................................................................147

# Avant-propos

Reproduire, imiter, simuler les comportements humains de manière artificielle est un vieux rêve de l'homme. Dès l'antiquité, certains textes évoquent déjà des automates pouvant se déplacer de manière autonome grâce à des roulettes d'or (livre XVIII de *l'Iliade*, Homère), des statues si vivantes qu'il faut les empêcher de s'enfuir (*Les statues de Dédale*, Platon) et même des automates à tête parlante (*La Bible*, Ezéchiel, XXI, 26). Les progrès technologiques aidant, diverses réalisations peuvent effectivement voir le jour comme des fontaines animées (XVI$^{e}$ siècle), le joueur de flûte traversière (Vaucanson, XVIII$^{e}$ siècle) ou, plus proche de nous et grâce à l'électronique, l'apparition des premiers ordinateurs. Ce nouvel outil va d'abord être utilisé pour automatiser les calculs. Puis, on se rend vite compte qu'il est aussi possible de manipuler des symboles et, en 1956, Allen Newell, John C. Shaw et Herbert A. Simon présentent leur "Logic Theorist" qui permet de démontrer des théorèmes de la logique des propositions. Cet épisode est un des premiers balbutiements d'une nouvelle discipline scientifique : l'Intelligence Artificielle (ou IA).

Ce vocable désigne un des secteurs de la recherche où sont associées questions théoriques fondamentales et applications pratiques. Les chercheurs en Intelligence Artificielle imaginent un ensemble de méthodologies et d'outils appelés à intervenir dans notre environnement technologique et dans nos méthodes de communication. Des robots aux jeux de l'esprit, de l'apprentissage automatique aux systèmes experts, des langages de programmation aux langues naturelles ou de la vision artificielle au traitement automatique de la parole, les domaines d'investigation de l'Intelligence Artificielle sont variés et concernent de plus en plus directement l'ensemble de notre société.

Ce document présente les travaux réalisés, dans le cadre d'un doctorat en informatique, sur l'un des thèmes de l'Intelligence Artificielle : la reconnaissance automatique de la parole.

# Introduction

Si l'aboutissement de la Reconnaissance Automatique de la Parole (RAP[1]) est que des personnes puissent dialoguer en langage naturel avec une machine, les Systèmes de Reconnaissance Automatique de la Parole (SRAP) actuels en sont encore bien loin.

Les recherches en RAP ont véritablement commencé dans les années soixante-dix, notamment avec le projet ARPA (Advanced Research Projects Agency) [Newell, 73 ; Klatt, 77] de "compréhension de la parole", lancé en 1971. Les conclusions de ce premier projet ARPA marquaient la fin d'une période optimiste où une partie de la communauté imaginait des progrès rapides des SRAP : malgré des directions prometteuses, le sujet nécessite un effort à long terme. Depuis, la RAP est au cœur de nombreuses recherches qui ont permis de repousser petit à petit les limites imposées aux systèmes par la complexité de la tâche. Depuis les SRAP expérimentaux mono-locuteur, en mots isolés et à petit vocabulaire des années quatre-vingt, les laboratoires de recherche travaillent maintenant sur des systèmes multi-locuteurs, en parole continue et à très grand vocabulaire. Par ailleurs, les technologies vocales sortent petit à petit des laboratoires de recherche et permettent la mise au point d'applications comme des serveurs télématiques, des bornes interactives, des systèmes de commande ou de saisie [Néel, 96]. Ces dernières années ont vu, en particulier, la commercialisation de SRAP type machine à dictée. Cantonnées tout d'abord dans des applications spécifiques fermées, telle que la saisie vocale des comptes-rendus médicaux, les dernières versions de ces systèmes permettent la saisie vocale de textes divers en parole continue[2].

Les travaux présentés ici se situent dans le cadre de la réalisation d'un Système de Reconnaissance Automatique de la Parole Continue (SRAPC) du français. L'étendue des domaines concernés et l'impossibilité de réaliser un SRAP dans son intégralité nous ont conduit à adopter certaines restrictions. Notre démarche consiste à étudier, développer et intégrer les éléments constitutifs d'un SRAP. D'une manière générale, la problématique traitée recouvre l'ensemble des problèmes rencontrés lors de l'automatisation des processus de construction d'un moteur de RAP.

Dans une première partie, nous définissons les bases théoriques utilisées en RAP. Après avoir présenté les principaux facteurs caractéristiques du domaine, nous proposons de distinguer les notions de méthodologies et de stratégies de reconnaissance. En fonction de cette distinction, nous décrivons les différentes approches et stratégies pouvant être employées. Nous insistons notamment sur les méthodes probabilistes utilisées dans ce travail de recherche et sur les raisons qui poussent à mettre en œuvre des stratégies dites multi-passes. Nous terminons par un tour d'horizon de SRAPC afin de faire le point sur les différentes stratégies employées actuellement.

---

[1] L'annexe A présente la table des abréviations utilisées dans ce document.

[2] Nous trouvons dans la presse informatique de nombreux articles sur la RAP qui permettent de suivre les évolutions des systèmes proposés au grand public. Nous citons, à titre d'exemples, des articles sur les systèmes de commande [Romain, 94], les bornes interactives [Le Hir, 97], les premiers systèmes de dictée automatique en mots isolés [Cornuau, 93 ; Jovanovic, 94 ; Neuman, 95a ; Neuman, 95b ; Gaveau, 96] puis en parole continue [Neuman, 97].

Nous commençons la deuxième partie en précisant le cadre général de notre travail, ses ambitions, mais aussi ses limites. Puis, nous présentons les composants nécessaires à la réalisation d'un système de Décodage Acoustico-Phonétique (DAP) fondé sur la modélisation Markovienne. Cette présentation sert de fil conducteur pour une description des différents éléments réalisés. En ce qui concerne le DAP, nous étudions particulièrement la phonétisation des corpus et l’impact de l'alignement des données acoustiques et phonétiques sur l'initialisation des modèles probabilistes acoustiques. Des expériences de reconnaissance nous permettent de valider et confronter les approches proposées.

La troisième partie est consacrée à l'accès lexical et à la modèlisation du langage. Tout d'abord, nous présentons le type d'accès lexical et de modélisation linguistique utilisé dans le cadre de la réalisation d'un SRAPC de référence. Puis, nous proposons deux stratégies de reconnaissance alternatives. La première est une stratégie multi-passes fondée sur une simplification du lexique de décodage (lexique homophone) et un processus de filtrage acoustico-linguistique permettant de déterminer l'hypothèse optimale. La seconde stratégie proposée vise à inclure des ressources linguistiques diverses (pour le moment un tri-gramme) directement dans un processus de décodage dérivé de l'algorithme de parcours de graphe A*. Des expériences de reconnaissance sur ces systèmes permettent de comparer les différentes approches proposées.

Dans la quatrième et dernière partie, nous nous intéressons à la fabrication de Modèles de Langage (ML) probabilistes. Nous proposons une méthode qui permet de faciliter la mise au point de ML grâce à un processus d'extraction de Sous-Modèles de Langage (SML) à partir d'un ML calculé au préalable. Nous illustrons cette méthode sur un ML de 64k à partir duquel nous calculons une suite de SML.

# Première partie

# Bases théoriques

L'objectif de cette première partie est de définir les bases théoriques utilisées en Reconnaissance Automatique de la Parole (RAP).

Après avoir caractérisé le problème de la RAP en fonction des critères usuels du domaine (reconnaissance mono / pluri / multi-locuteurs, lexique, mode d'élocution, environnement, connaissances linguistiques), nous proposons de distinguer les notions de méthodologie et de stratégie de reconnaissance. Nous commençons par exposer les différentes méthodologies de reconnaissance existantes : l'approche analytique est principalement présentée à travers l'exemple des Systèmes de Reconnaissance Automatique de la Parole (SRAP) analytiques du Laboratoire Informatique d'Avignon (LIA) ; quant à la description de l'approche globale, elle s'appuie sur une étude bibliographique. Une insistance particulière est portée sur l'approche probabiliste et la modélisation markovienne des composants acoustiques et linguistiques qui seront appliquées dans les deuxième et troisième parties de ce document dans le cadre de la réalisation d’un Système de Reconnaissance de la Parole Continue (SRAPC). Puis, nous étudions les stratégies de décodage en comparant trois stratégies fondatrices avec celle dite du Modèle de Propagation de Jeton. Enfin, un tour d'horizon des SRAPC actuels nous permet de faire le point sur les différents types de stratégies de décodage utilisées actuellement.

## 1.1. Caractérisation du problème

La réalisation d'un SRAP est particulièrement difficile et compliquée : le signal de parole est soumis à une combinaison de nombreux facteurs complexes (redondance, très grande variabilité, effets de coarticulation, etc.). Par ailleurs, les performances demandées aux systèmes ont atteint un niveau très élevé. En définitive, on trouve peu de points communs entre, d'une part, un système capable de traiter la parole pour un locuteur donné, sur un petit vocabulaire spécifique, en mots isolés et dans un environnement protégé ; et, d'autre part, un système multi-locuteur, vocabulaire illimité, parole continue et environnement bruité.

Les travaux présentés dans ce document se situent dans le cadre de la réalisation d'un SRAPC du français[1]. Dans ce premier chapitre, nous définissons la problématique de la RAP en décrivant les principales caractéristiques et contraintes qui définissent les possibilités et les limites des SRAP [Calliope, 89 ; Mariani, 90 ; Haton, 91]. Nous profitons de ces définitions pour exposer les choix efféctués *a priori* dans nos travaux :

**Reconnaissance mono-locuteur, pluri-locuteur ou multi-locuteur**
Un SRAP peut être utilisé par un seul locuteur (mono-locuteur), par plusieurs locuteurs référencés (pluri-locuteur) ou être indépendant du locuteur (multi-locuteur).

Les phénomènes acoustiques liés à la production de parole sont extrêmement variables. Ils peuvent rendre deux prononciations d'un même énoncé très différentes. En général, on distingue deux types de variabilité : la variabilité intra-locuteur (pour un même locuteur) et la variabilité inter-locuteur (entre plusieurs locuteurs). La variabilité intra-locuteur est due aux variations naturelles de la voix (nous ne prononçons jamais exactement de la même façon un même mot) mais aussi à l'état physique et psychologique du locuteur (fatigue, stress, rhume, etc.) ou encore au bruit de fond qui modifie la façon de parler du locuteur[2]. La variabilité inter-locuteur est, *a priori*, plus importante. Elle est causée par les différences physiologiques entre locuteurs (conduit vocal, cavités résonantes, etc.) et par les habitudes de prononciation liées aux milieux sociaux et géographiques (accents).

Ces phénomènes de variabilité sont pris en compte par le processus de modélisation acoustique. Cette modélisation consiste à définir l'ensemble de références acoustiques qui servira au processus de reconnaissance. Les modèles acoustiques réalisés, dans la deuxième partie de ce document, sont indépendants du locuteur.

---

[1] Nous présentons le cadre général de notre étude au début de la deuxième partie de ce document (cf. *2.1. Cadre de notre étude*).

[2] Le réflexe qui pousse un locuteur à augmenter son effort vocal en présence de bruit est appelé effet Lombard [Anglade, 94].

### Lexique

Le lexique est une ressource essentielle car c'est à partir de lui que le SRAP va pouvoir décoder les mots [Lamel, 96]. Sa taille (le nombre de mots), sa couverture, la prise en compte des Mots Hors Vocabulaire (MHV), sa phonétisation, son taux d'homophonie[1], sa segmentation (mots composés), la prise en compte des apostrophes, des liaisons et des variantes de prononciation sont les principales caractéristiques qui permettent de définir le lexique d'une application. Par ailleurs, la relation entretenue par les différentes entités qui composent le lexique conditionne en partie la difficulté de la tâche[2].

Nous abordons les problèmatiques lexicales dans la troisième partie de ce document.

### Mode d'élocution : mots isolés ou parole continue

En mots isolés, les mots prononcés sont séparés par des pauses. Ainsi, les frontières entres les mots sont aisément identifiables (il existe diverses méthodes de détection des silences). En revanche, en mode "parole continue", ces frontières s'estompent. Il faut alors faire de très nombreuses hypothèses de fin de mots dans le processus de décodage, ce qui complexifie considérablement la tâche. De plus, les phénomènes de coarticulation sont beaucoup plus importants.

Notons que l'ensemble de ce travail s'inscrit dans le cadre de la parole continue.

### Environnement

L'environnement acoustique (condition d'enregistrement, milieu bruité, etc.) peut fortement influencer les performances des SRAP. À ce propos, nous noterons que la capacité de résistance des systèmes aux modifications de l'environnement (notion de robustesse) est l'une des problématiques les plus sensibles du moment.

Les problèmes liés à l'environnement acoustique sortent du cadre de la présente étude.

### Connaissances linguistiques : grammaire formelle et/ou modèle de langage probabiliste

---

[1] Au-delà du taux d'homophonie, il faut également se soucier du degré de parenté phonétique entre les entrées du lexique.

[2] Comme, par exemple, la reconnaissance des lettres de l'alphabet qui est une tâche difficile car, si le vocabulaire est petit, les entrées lexicales sont courtes et phonétiquement proches.

Le langage utilisé peut être contraint par une syntaxe artificielle (grammaire formelle) ou naturelle (dans ce cas, des modèles de langage probabilistes sont généralement utilisés). Par ailleurs, le domaine d'application peut être fermé (domaine d'application défini précisément [Cornuau, 93 ; Jovanovic, 94]) ou ouvert (pas de domaine *a priori*).

La mise en place de contraintes linguistiques est indispensable pour obtenir des taux de reconnaissance acceptables. Elles permettent l'ajout de connaissances supplémentaires dans le processus de reconnaissance. Nous proposons une approche de ce type dans les deuxième et troisième parties de ce document avec l'utilisation de modèles de langage probabilistes.

## 1.2. Méthodologies de reconnaissance

Traditionnellement, deux méthodologies sont proposées en reconnaissance de la parole : l'approche **analytique** et l'approche **globale**. La première approche consiste à décomposer le problème : à partir d'une étape de décodage acoustico-phonétique, une suite de modules de niveau supérieur va permettre de traiter le signal de parole. Notons que l'approche analytique est, en règle générale, fondée sur des bases de connaissances formelles (de type système expert). Par contraste, la seconde approche cherche à identifier globalement des segments de parole en les comparant avec des références enregistrées préalablement.

**Remarque : distinction entre méthodologie de reconnaissance et stratégie de reconnaissance**

Nous avons choisi de distinguer les notions de *méthodologie de reconnaissance* d'une part, et de *stratégie de reconnaissance* d'autre part. Nous considérons la *méthodologie de reconnaissance* comme un point de vue dont résultent différentes méthodes et techniques qui permettent d'identifier **un segment** de parole. La *stratégie de reconnaissance* désigne alors la façon de combiner ces méthodes et techniques afin de prendre en compte la parole dans son ensemble, comme **une suite de segments**.

### 1.2.1. L'approche analytique

Les caractéristiques principales de l'approche analytique sont une segmentation *a priori* du signal acoustique, une organisation modulaire hiérarchique et l'utilisation de bases de connaissances formelles.

À titre d'exemple, nous décrirons succinctement les fondements des SRAP analytiques développés au LIA jusqu'au début des années quatre-vingt-dix[1].

#### 1.2.1.1. Schéma général

La figure 1.1 présente les différents modules et interactions qui constituent le fondement des SRAP analytiques du LIA.

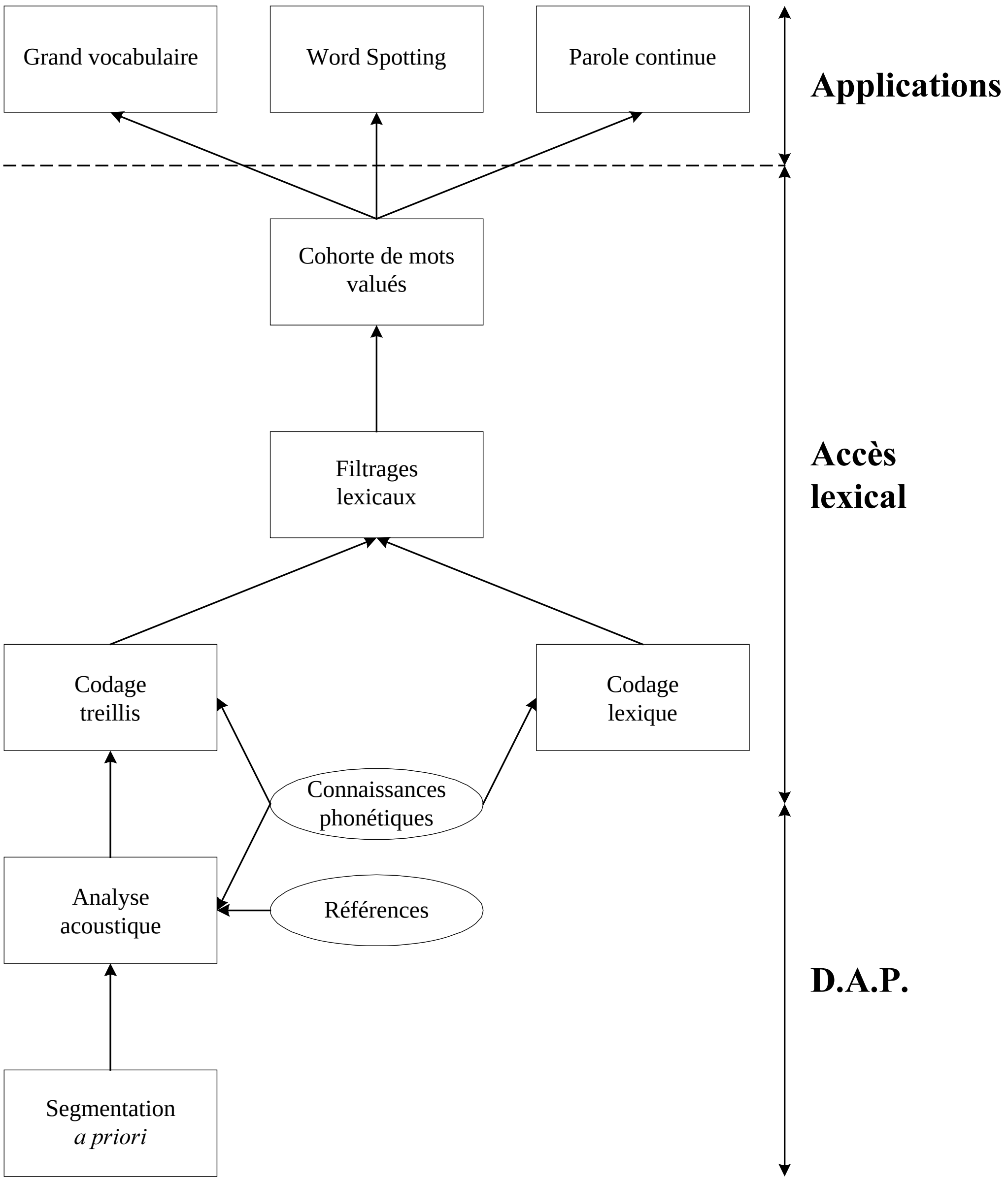

Grand vocabulaire
Word Spotting
Parole continue
Applications
Cohorte de mots valués
Filtrages lexicaux
Accès lexical
Codage treillis
Codage lexique
Connaissances phonétiques
Analyse acoustique
Références
D.A.P.
Segmentation *a priori*

Figure 1.1 : Synoptique d'un SRAP utilisant une approche analytique
Schéma général des modules analytiques du LIA

---

[1] Pour d'autres exemples de SRAP analytiques, se référer à [Klatt, 77] pour des systèmes développés lors du projet ARPA, [De Mori, 80] pour des systèmes utilisant les ensembles flous, [Bigorne, 88] pour des systèmes hiérarchisés, [De Mori, 85 ; Mercier, 85] pour des systèmes liés aux techniques issues de l'Intelligence Artificielle et, pour un historique plus complet, [Calliope, 89 ; Mariani, 90].

L'indépendance relative des modules permet de modifier l'architecture du système en fonction de l'application désirée. Sur cette base ont été réalisés : un SRAP continu avec traitement formel de connaissances linguistiques [Spriet, 93], un SRAP de mots isolés sur de grands vocabulaires [Béchet, 94a], ou encore une application de reconnaissance de mots clés dans la parole continue (ou *Word-Spotting*) [Igounet, 93 ; Béchet, 94c].

Nous noterons que, dans l'approche analytique, les niveaux et types de connaissances (phonétiques, lexicales, syntaxiques, etc.) mis en œuvre s'appliquent, par l'intermédiaire des différents modules, les uns à la suite des autres.

### 1.2.1.2. Le Décodage Acoustico-Phonétique

Le processus de Décodage Acoustico-Phonétique (DAP) [Méloni, 91 ; Gilles, 93] permet d'obtenir un treillis d'unités phonétiques valuées (cf. figure 1.2).

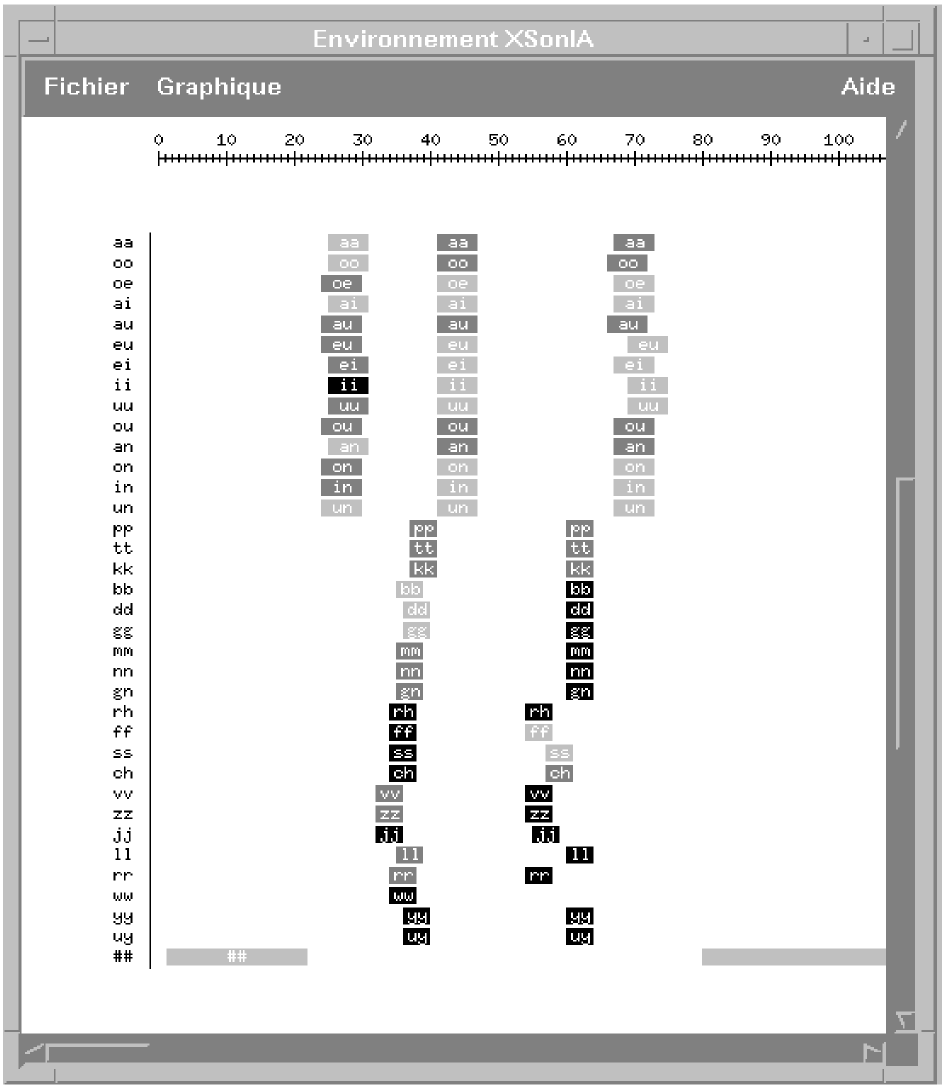

Environnement XSonIA
Fichier
Graphique
Aide
0
10
20
30
40
50
60
70
80
90
100
aa
oo
oe
ai
au
eu
ei
ii
uu
ou
an
on
in
un
pp
tt
kk
bb
dd
gg
mm
nn
gn
rh
ff
ss
ch
vv
zz
jj
ll
rr
ww
yy
uy
##

Figure 1.2 : Exemple d'un treillis de phonèmes suite au processus de DAP sur le mot "abaisser" (les phonèmes les plus clairs sont les plus vraisemblables)

Il s'effectue en deux temps. Dans un premier temps, après une phase de traitements acoustiques (numérisation, paramétrisation, densité de passages par zéro, etc.), une segmentation voyelles-consonnes est réalisée au moyen d'outils pour la reconnaissance de formes sur divers types de critères tels que l'énergie ou la stabilité spectrale. Dans un second temps, sur les zones vocaliques et consonantiques ainsi déterminées, les phonèmes correspondants sont identifiés et values par comparaisons des paramètres acoustiques calculés à des formes ou des seuils de référence. Ces comparaisons sont effectuées à l'aide de distances spectrales en fonction de la structure particulière des phonèmes concernés et de diverses informations contextuelles contenues dans les trames adjacentes du signal (variation d'énergie, fonctions d'instabilité spectrale, etc.). L'identification exacte étant en général impossible, le DAP propose plusieurs hypothèses, affectées d'un score acoustique, pour chacun des segments voyelles-consonnes. On obtient alors un treillis d'unités phonétiques valuées (cf. figure 1.2) qui contient l'ensemble des sons vraisemblables et "essentiels" (les voyelles et la plupart des consonnes intervocaliques) permettant d'accéder à un sous-ensemble raisonnablement limité du lexique.

#### 1.2.1.3. Accès lexical

La partie lexicale [Béchet, 94a ; Béchet, 94b] est constituée de modules de codage de l'information et par une étape d'accès lexical. Le rôle du module de codage est de trouver une fonction de correspondance liant les unités phonétiques identifiées par le DAP avec les différents mots du lexique. La structure du treillis phonétique et les règles limitatives du décodage phonétique [Gilles, 93] empêchent de réaliser un alignement direct des mots cibles du lexique sur le treillis. Une représentation intermédiaire est réalisée à partir de macro-classes décrivant des sons ayant des traits acoustiques distinctifs. Les macro-classes permettent de représenter les données lexicales et les hypothèses phonétiques issues du DAP sous la forme d'une structure commune. L'accès lexical consiste alors à affiner cette structure jusqu'à la décomposition phonologique complète de chaque item par l'intermédiaire de filtres lexicaux successifs. Cela permet de réduire progressivement le nombre d'hypothèses lexicales pour donner, en solution, une cohorte de mots valués.

### 1.2.2. L'approche globale

L'approche globale évite toute segmentation *a priori* et ne fait pas d'hypothèses sur le type des éléments à traiter. Elle effectue des comparaisons sur un ensemble de références en traitant les données et les connaissances dans leur globalité.

Le principe de base consiste à donner au système de reconnaissance au moins une image de chacune des unités qu'il est censé devoir identifier par la suite. Cette opération est faite lors de la phase d'apprentissage qui permet de constituer la base de données de référence du système. Le processus de décodage consiste alors à comparer l'image de l'unité à identifier avec celles de la base de référence. L'unité dont la référence est la plus proche est déclarée reconnue. Par ailleurs, dans la mesure où les données à traiter sont constituées d'une suite d'unités de base (comme, par exemple, la reconnaissance d'une phrase composée de mots), plusieurs types de stratégies de reconnaissance peuvent alors être appliqués.

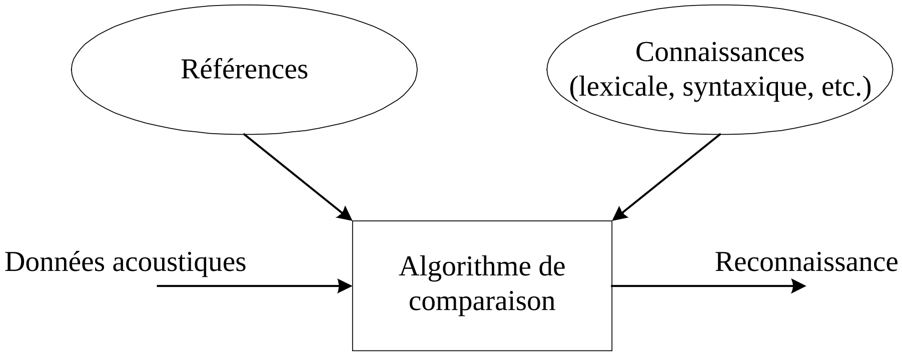


Figure 1.3 : Synoptique d'un SRAP utilisant une approche globale

La figure 1.3. donne en exemple le schéma d'un SRAP utilisant une approche globale. Remarquons que les connaissances disponibles pour le système sont utilisées simultanément.

Les algorithmes de comparaison utilisés dans l'approche globale consistent soit en une mesure de similarité par rapport à des formes de référence [Itakura, 75 ; Sakoe, 78 ; Rabiner, 79 ; Bridle, 79], soit en une mesure de probabilité par rapport à des modèles probabilistes.

### 1.2.3. Commentaires

Les approches analytique et globale telles que nous les avons décrites dans les sections précédentes sont à l'origine des premiers SRAP. Ces systèmes mettaient en œuvre un certain nombre d'hypothèses simplificatrices : un petit vocabulaire et/ou une prononciation mot à mot et/ou un seul locuteur. L'apport le plus important de ces systèmes se situe sans doute dans le domaine des architectures logicielles de systèmes complexes à plusieurs sources et niveaux de connaissances, plutôt que dans le traitement de la parole proprement dit [Calliope, 89].

À partir des années quatre-vingt, la majorité des SRAP s'orientent petit à petit vers les techniques les plus prometteuses : l'approche probabiliste et les modèles de Markov cachés. Cependant, la distinction entre méthodologies globales et analytiques est de plus en plus discutable. En effet, les méthodes statistiques à base de chaînes de Markov cachées sont des méthodes globales exploitant des unités acoustiques sub-lexicales et des modules linguistiques. De plus, comme nous le verrons dans le chapitre consacré aux stratégies de décodage, ces SRAP utilisent des stratégies multi-passes qui ressemblent fortement aux successions de modules de traitement des SRAP analytiques conçus dans le passé.

Pour notre part, c'est après avoir travaillé sur des systèmes de reconnaissance de type analytique [Igounet, 93 ; Béchet, 94b ; Béchet, 94c] que nous nous sommes intéressés aux processus de construction d'un SRAP fondé sur l'approche globale probabiliste.

## 1.3. L'approche globale probabiliste

J.-K. Baker et F. Jelinek proposent une formulation probabiliste du problème de la RAP [Baker, 75 ; Jelinek, 76]. Issue des travaux de Shannon sur la théorie de l'information, cette approche est aujourd'hui classique et incontournable. Elle est à la base de la majorité des SRAP actuels.

Dans le cadre de la RAP, la chaîne de traitement de l'information est la suivante [Jouvet, 96] (figure 1.4) : une suite de mots *M* est énoncée par un locuteur, transformée en ondes électriques par l'intermédiaire d'un microphone et numérisée sous la forme d'un signal acoustique *S*. Puis, un processeur acoustique transforme le signal *S* en unités de temps (typiquement de 10 ms chacune) sur lesquelles une suite d'observations acoustiques *O* est calculée par des techniques classiques de traitement du signal. Enfin, un décodeur transforme cette suite d'observations acoustiques *O* en une suite de mots résultats $\hat{M}$.

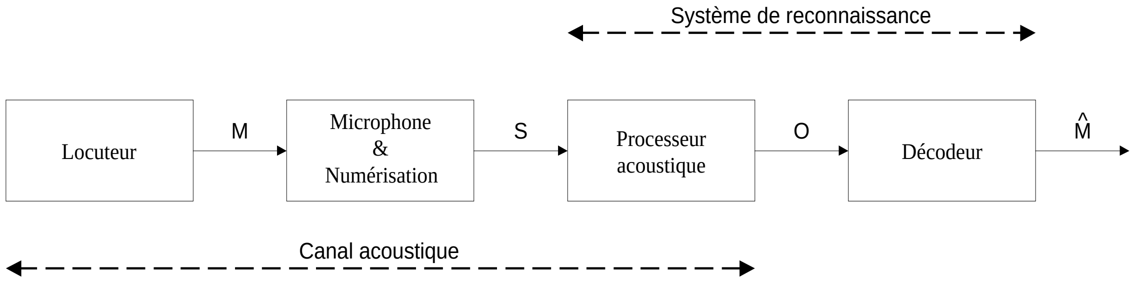


Figure 1.4 : Chaîne de traitement de l'information

Soit $M = m_1\, m_2 \ldots.\, m_n$ une suite de *n* mots prononcés par un locuteur.

Soit $O_T = o_1\, o_2 \ldots.\, o_t$ la suite d'observations acoustiques issues de la transformation de $M_n$ par le processeur acoustique.

La tâche de la RAP consiste à déterminer, parmi toutes les séquences de mots $M$ possibles, la séquence $\hat{M}$ la plus probable connaissant les observations acoustiques $O_T$ :

$$\hat{M} = \underset{M}{Argmax}\, P(M/O) \qquad \text{(Eq. 1.1)}$$

Grâce à la règle de Bayes, il est possible d'écrire :

$$P(M/O) = \frac{P(O/M) \times P(M)}{P(O)} \qquad \text{(Eq. 1.2)}$$

Avec :

Le terme *P(O/M)* représente la probabilité *a posteriori* d'émission de l'observation acoustique $O$ pour une phrase $M$ donnée. Son calcul repose sur une modélisation du canal acoustique.

Le terme *P(M)* représente la probabilité *a priori* d'une séquence de mots $M$. Son calcul repose sur le modèle de langage utilisé.

Le terme *P(O)* représente la probabilité d'occurrence de la suite d'observations acoustiques $O$. Elle est indépendante de $M$ et reste donc constante lorsque $M$ varie.

*P(O)* ne dépendant pas de M dans l'équation 1.2, nous pouvons l'omettre et écrire :

$$\hat{M} = \underset{M}{Argmax}\, P(O/M) \times P(M) \qquad \text{(Eq. 1.3)}$$

Cette formulation fait clairement apparaître le principe probabiliste de la reconnaissance comme le calcul puis le produit des deux termes[1] *P(O/M)* et *P(M)*. Notons que cela permet de considérer la tâche de la RAP globalement en intégrant les modules acoustiques *P(O/M)* et linguistiques *P(M)* dans un processus de décodage synchrone.

---

[1] *P(O/M)* et *P(M)* sont respectivement connues sous les noms de probabilité du modèle acoustique et probabilité du modèle de langage.

## 1.4. Modélisation Markovienne des probabilités acoustiques et linguistiques

Que ce soit au niveau acoustique ou linguistique, l'outil de la théorie de l'information le plus communément utilisé en RAP est le modèle de Markov caché (Hidden Markov Model ou HMM). Dans ce cadre, nous pouvons considérer les mots et les observations acoustiques comme des symboles émis par deux sources différentes ; ces dernières étant modélisées par des processus Markoviens [Calliope, 89].

Pour un examen approfondi des points étudiés dans ce chapitre, le lecteur pourra se reporter aux ouvrages et articles référencés suivants : [Jelinek, 82 ; Bahl, 83[1] ; Levinson, 83 ; Poritz, 88 ; Jouvet, 88 ; Calliope, 89 ; Rabiner, 89].

### 1.4.1. Modélisation acoustique probabiliste : les modèles de Markov cachés

Les modèles de Markov cachés résultent de l'association d'un ensemble de fonctions de densité de probabilité (ou distributions de probabilité) et d'une chaîne de Markov. Les fonctions de densité de probabilité donnent les probabilités sur l'ensemble des observations acoustiques et la chaîne de Markov sert de support aux distributions. Un modèle de Markov peut être défini par un automate probabiliste d'états finis[2].

#### 1.4.1.1. Présentation

Succinctement, nous pouvons décrire un HMM (cf. figure 1.5) par un ensemble de noeuds (ou états) et de transitions (ou arcs) reliant ces états. À chaque état $i$ est associée une distribution de probabilité $b_i(o)$ et à chaque transition de l'état $i$ à l'état $j$ est associée une probabilité de transition $a_{ij}$. La distribution de probabilité $b_i(o)$ représente la probabilité d'émission sur l'état $i$ de l'observation acoustique $o$. La probabilité de transition $a_{ij}$ représente la probabilité de passer de l'état $i$ vers l'état $j$ en empruntant la transition $a_{ij}$.

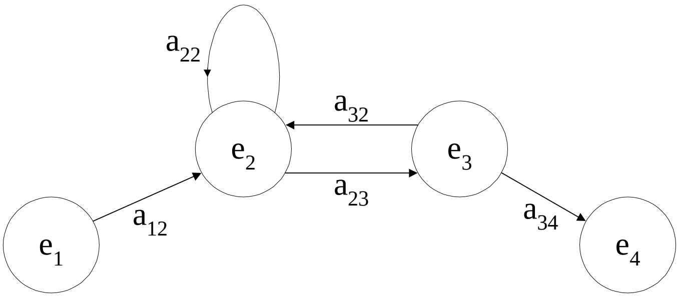


Figure 1.5 : Exemple de HMM

---

[1] Nous remarquons que les références [Jelinek, 82] et [Bahl, 83] désignent, en fait, les mêmes articles.

[2] Pour une définition formelle (et en français) des modèles de Markov, nous proposons de se référer à [Jouvet, 88].

À chaque instant $t$, un processus aléatoire passe d'état en état en fonction des observations acoustiques $O_T = o_1\ o_2\ ....\ o_t$ observées. L'état de l'automate n'est pas directement observable (ce sont les symboles émis qui sont observables). Ceci implique que la suite des états (le chemin) empruntés par l'automate est inconnue, elle est cachée. C'est pourquoi, ces modèles sont appelés modèles de Markov cachés.

Les HMM qui utilisent des probabilités de transition $a_{ij}$ ne dépendant que de l'état de départ $i$ sont dits[1] d'ordre 1. Ajoutons qu'il existe différentes classes de distributions de probabilités (discrètes ou continues) qui conditionnent le type du modèle de Markov caché selon l'ensemble des observations (fini ou infini). On parle alors de HMM discrets ou de HMM continus. Par ailleurs, des approches dites hybrides proposent l'utilisation de réseaux neuromimétiques pour estimer les probabilités d'émission sur les états [Hochberg, 94 ; Morgan, 95].

Dans le cadre des modèles acoustiques réalisés lors de ce travail, nous choisissons des HMM à distributions de probabilités continues (cf. *2.7. Modèles de Markov*).

**Remarque**

> Dans cette présentation des modèles de Markov cachés, nous avons choisi de placer l'émission des observations acoustiques sur les états des modèles. Remarquons que sur un plan théorique, l'émission des observations acoustiques peut se faire indifféremment sur les états ou sur les transitions des HMM.

### 1.4.1.2. Topologie des HMM pour la RAP

Pour la commodité de l'exposé, nous nous plaçons ici dans le cadre de la reconnaissance d'unités acoustiques isolées ; une unité acoustique donnée étant représentée par un HMM. Nous abordons le problème de la combinaison des unités de base dans le chapitre consacré aux stratégies de décodage. Notons cependant que les unités acoustiques généralement utilisées en RAP sont hiérarchiquement combinées entre elles pour former des séquences de niveau supérieur. Par exemple, l'unité acoustique de base peut être définie par le phonème. La concaténation de phonèmes forme alors un mot et la concaténation de mots une phrase.

La parole est un phénomène dont la dimension temporelle ne peut être ignorée. Les HMM utilisés pour la représenter sont, la plupart du temps, des modèles "gauche-droit" qui ne permettent pas de "retour en arrière". C'est-à-dire que l'automate probabiliste d'états finis définissant la chaîne de Markov associée ne contient pas de transition $a_{ij}$

---

[1] Pour l'utilisation des modèles de Markov de second ordre en RAP, se reporter à [Mari, 94 ; Mari, 96].

entre états *i* et *j* de telle sorte que :

$$i > j \Rightarrow a_{ij} = 0 \qquad \text{(Eq. 1.4)}$$

Dans ce cadre, R. Bakis [Bakis, 1976] a proposé un modèle type pour représenter un mot qui permet le bouclage sur l'état courant (progression acoustique stationnaire), le passage à l'état suivant (progression acoustique "standard") ou le saut d'un état (progression acoustique "rapide") (cf. figure 1.6). Le nombre d'états du modèle est alors proportionnel à la durée moyenne de ce mot.

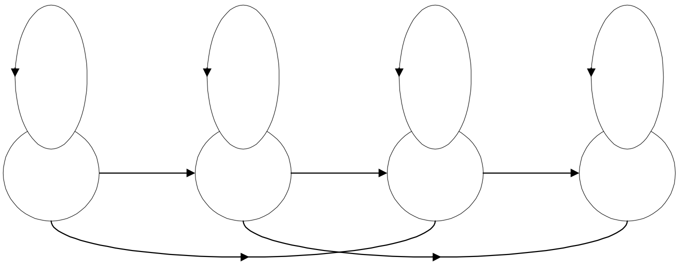


Figure 1.6 : Exemple de HMM de Bakis

Ce type de modèle est devenu générique dans le domaine de la RAP. Il est utilisé dans de nombreux systèmes pour modéliser les unités acoustiques de base à partir de concepts empruntés à la linguistique. Nous citons, notamment, les cas des phonèmes élémentaires ou contextuels (diphones et triphones [Cravero, 86]). Dans le cas des phonèmes contextuels, une solution au problème posé par le grand nombre de paramètres à estimer consiste à définir des classes d'équivalences (classes de phonèmes, classes d'états, etc). Ces classes peuvent être choisies selon des critères linguistiques [Derouault, 87a ; Derouault, 87b] ou obtenues par classification automatique [Lee, 89 ; Lee, 90a]. Nous précisons d'autres perfectionnements possibles dans la modélisation acoustique par HMM dans le chapitre *2.7. Modèles de Markov* ; nous y présentons le type de modèle adopté dans le SRAP réalisé lors de ce travail.

#### 1.4.1.3. Reconnaissance d'un modèle

Soit le terme $P(O/M)$ de l'équation 1.3 représentant la composante acoustique d'un SRAP probabiliste. Le calcul de ce terme implique l'estimation de $P(O/M)$ pour chaque mot $M$ possible, c'est-à-dire pour chaque HMM modélisant les mots définis dans le vocabulaire du SRAP. Mais la probabilité que la suite de paramètres acoustiques $O_T = o_1\, o_2 \ldots.\, o_t$ soit produite par le HMM n'est pas directement calculable car le chemin emprunté par le processus aléatoire est *a priori* inconnu.

Soit une nouvelle formalisation de *P(O/M)* comme la somme des probabilités sur tous les chemins possibles Q :

$$P(O/M) = \sum_{\substack{\text{Tous les}\\ \text{chemins } Q}} P(O,Q\,/\,M) \qquad \text{(Eq. 1.5)}$$

Le calcul direct de *P(O/M)* avec l'équation 1.5 est inapplicable pour des graphes de recherche un tant soit peu importants car le nombre de chemins possibles devient rapidement impossible à décrire. L'algorithme itératif dit *Forward-Pass* (estimation directe) permet de calculer récursivement *P(O/M)*, c'est-à-dire la probabilité d'émission des observations cumulée sur toutes les séquences d'états possibles, sans choisir un chemin particulier [Baum, 1972].

L'algorithme de Viterbi [Viterbi, 67 ; Forney, 73] est utilisé pour trouver le chemin de probabilité optimale $\hat{Q}$ (la séquence d'états cachés la plus probable et sa probabilité d'émission associée). De cette façon, nous obtenons l'alignement optimal d'une suite de paramètres acoustiques ainsi que sa probabilité pour un modèle de Markov donné ; c'est-à-dire le chemin qui conduit à la plus forte probabilité d'émission pour cette suite de paramètres acoustiques.

L'algorithme de Viterbi peut être vu comme une modification de l'opérateur *somme* du *Forward-Pass* par l'opérateur *maximum* et un chaînage arrière des meilleures transitions. L'équation 1.5 s'écrit alors :

$$P(O,\hat{Q}/M) = \max_{\substack{\text{Tous les}\\ \text{chemins } Q}} P(O,Q/M) \qquad \text{(Eq. 1.6)}$$

Notons que l'algorithme de Viterbi est, peu ou prou, à la base de la plupart des SRAP utilisant les modèles de Markov cachés.

#### 1.4.1.4. Apprentissage des paramètres d'un modèle

La reconnaissance d'un ensemble d'unités de base nécessite l'apprentissage des paramètres associés à leurs HMM respectifs. Cela suppose l'existence d'un modèle pour chaque unité de base et d'un corpus de données acoustiques correspondant. Nous rappelons que les paramètres d'un HMM sont les fonctions de densités de probabilité et les probabilités de transition ; la topologie de la chaîne de Markov et le type des densités de probabilité (discrètes ou continues) étant, en général, fixés *a priori*[1].

---

[1] Pour la constitution automatique de topologie Markovienne, se référer aux travaux sur les formes fénoniques [Bahl, 86 ; Cerf-Danon, 90 ; El-Bèze, 90].

Habituellement, le processus d'apprentissage consiste d'abord à faire correspondre à chaque unité de base possible un HMM distinct ; puis, à partir d'un corpus d'apprentissage, à estimer les paramètres du HMM selon certains critères.

En RAP, il existe deux principaux types d'apprentissage : l'apprentissage par maximum de vraisemblance (Maximum Likelihood Estimation ou MLE) et l'apprentissage par maximum d'information mutuelle (Maximum Mutual Information ou MMI).

L'apprentissage MLE cherche à déterminer les paramètres optimaux des modèles de Markov $\hat{M}$ qui maximisent la vraisemblance sur l'ensemble des observations acoustiques $O$ du corpus d'apprentissage :

$$\hat{M} = \underset{M}{Argmax} \prod_{O} P(O/M) \qquad \text{(Eq. 1.7)}$$

En général, les paramètres optimaux du modèle de Markov $\hat{M}$ sont obtenus de manière itérative par les algorithmes du *Forward-Backward* ou de Viterbi qui convergent vers un extremum local [Baum, 72]. Les formules de Baum-Welch permettent de ré-estimer les paramètres du nouveau modèle $M_{i+1}$ à partir des paramètres du modèle $M_i$ de l'itération $i$ telles qu'elles améliorent la vraisemblance sur l'ensemble des observations acoustiques $O$ du corpus d'apprentissage :

$$P(O/M_{i+1}) \geq P(O/M_i) \qquad \text{(Eq. 1.8)}$$

Cependant, le critère de maximum de vraisemblance n'est pas totalement satisfaisant. La convergence du modèle nécessite une bonne initialisation des paramètres et de nombreuses données d'apprentissage. Par ailleurs, ce critère ne tient pas compte des autres modèles lors de la maximisation des paramètres d'un HMM donné.

L'apprentissage MMI autorise la ré-estimation des paramètres en tenant compte de tous les modèles. La probabilité *a priori* des observations acoustiques $P(O)$ est décomposée comme une somme des probabilités conjointes sur tous les modèles [Bahl, 86 ; Mérialdo, 88a ; Kapadia, 93]. L'apprentissage MMI permet de maximiser la probabilité d'émission pour le bon modèle tout en la minimisant sur tous les autres. Mais les temps de calcul pour l'apprentissage sont plus longs et la convergence des modèles n'est pas assurée [Normandin, 94].

### 1.4.2. Modélisation linguistique probabiliste

Dans les SRAP, et plus particulièrement dans le cadre des systèmes de type "machine à dictée", l'utilisation de contraintes linguistiques est indispensable pour obtenir des taux de reconnaissance acceptables. Ces contraintes peuvent prendre différentes formes (modèle à syntaxe fixe, modèle probabiliste local, grammaire probabiliste). Dans ce chapitre, nous nous intéressons aux modèles de langage probabilistes locaux qui sont aujourd'hui les plus utilisés dans les SRAP de type "machine à dictée". Ces modèles ont pour but de compléter la décision prise par le système au moyen des observations acoustiques.

#### 1.4.2.1. Présentation des modèles de langage probabilistes locaux

Soit le second terme $P(M)$ de l'équation 1.3 qui, nous le rappelons, représente la probabilité *a priori* d'une séquence de mots $M$. La probabilité d'une phrase $M$ composée des $n$ mots $M = m_1\ m_2\ ....\ m_n$ peut s'écrire comme un produit de probabilités conditionnelles [Jelinek, 90] :

$$P(M) = P(m_1) \times \prod_{i=2}^{n} P(m_i/m_1....m_{i-1}) \qquad \text{(Eq. 1.9)}$$

En pratique, le mode de calcul de $P(M)$ selon l'équation 1.9 n'est pas réaliste. Même pour un nombre de mots $n$ petit, la quantité de données nécessaires à un apprentissage fiable des probabilités de toutes les séquences de $n$ mots possibles et les ressources matérielles informatiques nécessaires sont considérables. Une simplification s'impose.

D'une façon générale, les simplifications mises en œuvre consistent à partager l'historique en classes d'équivalence (notées $H$). Une approximation de l'équation 1.9 peut alors s'écrire :

$$P(M) \cong P(m_1) \times \prod_{i=2}^{h} P(m_i/H(m_1....m_{i-1})) \qquad \text{(Eq. 1.10)}$$

Ce type de modèle est appelé modèle de langage probabiliste local. Il se caractérise par la longueur de l'historique $h$ et le choix de la classe d'équivalence.

Sur des corpus d'apprentissage, on estime les probabilités par comptage (noté $C$) des occurrences de mots :

$$P(m_i/H(m_1....m_{i-1})) = \frac{C(m_i, H(m_1....m_{i-1}))}{C(H(m_1....m_{i-1}))} \qquad \text{(Eq. 1.11)}$$

#### 1.4.2.2. Modèles n-grammes

Dans le cadre des modèles de langage probabiliste locaux, c'est l'approche n-gramme qui est la plus utilisée. Dans cette approche, on suppose que la probabilité du mot suivant $m_i$ ne dépend pas de tous les mots du début de la phrase mais seulement des $n$ derniers mots $m_{i-n}....m_{i-1}$. Deux historiques sont alors dit équivalents s'ils se terminent par *n* mots identiques. Par conséquent, nous pouvons écrire l'approximation suivante :

$$P(m_i/m_1....m_{i-1}) \cong P(m_i/m_{i-n}....m_{i-1}) \quad \text{(Eq. 1.12)}$$

Par exemple, dans le cas d'un tri-gramme où l'on ne prend en compte que les deux derniers mots (*n=2*), nous redéfinissons les équations 1.12 et 1.10 telles que :

$$P(m_i/m_1....m_{i-1}) \cong P(m_i/m_{i-2}....m_{i-1}) \quad \text{(Eq. 1.13)}$$

Et en reportant ce terme dans 1.10 :

$$P(M) \cong P(m_1) \times P(m_2/m_1) \times \prod_{i=3}^{n} P(m_i/m_{i-2}....m_{i-1}) \quad \text{(Eq. 1.14)}$$

Cependant, pour de très grands vocabulaires, le nombre de tri-grammes $(m_{i-2}, m_{i-1}, m_i)$ est encore trop important. Il est impossible d'estimer des statistiques fiables sur leur fréquence d'occurrence. Ainsi [El-Bèze, 90] souligne que pour la langue anglaise, 23% des tri-grammes du corpus de test n'ont pas été rencontrés lors de la phase d'apprentissage sur un corpus de 1,5 millions de mots [Derouault, 84] ; tandis que pour le français, à partir d'un corpus d'apprentissage de 38 millions de mots, c'est 30% des tri-grammes de test qui sont inconnus [Cerf-Danon, 90]. Pour ne pas affecter une probabilité nulle aux tri-grammes non rencontrés dans la phase d'apprentissage, différentes techniques qui exploitent un ou plusieurs modèles d'ordre inférieur sont utilisées. On peut, notamment, citer la technique dite du *back-off* [Katz, 87] ou celle des combinaisons de modèles par interpolation [Jelinek, 80].

#### 1.4.2.3. Autres types de modèles[1]

L'une des variantes possible des modèles de langage probabilistes locaux n-grammes est l'approche n-classes où les probabilités ne dépendent plus des *n* mots mais de la suite de *n* classes syntaxiques.

À partir de cette idée, il est alors possible de combiner ces deux types de modèles : le n-grammes au niveau du mot et le n-classes au niveau des classes syntaxiques [Derouault, 85 ; El-Bèze, 93].

Il existe également des modèles de langage purement syntaxiques qui reposent sur des grammaires formelles définissant l'ensemble des séquences de mots correctes. Dans cette catégorie des modèles de langage purement syntaxiques, nous trouvons, par exemple, les grammaires à structure de phrase [Chomsky, 57 ; Aho, 72 ; Harrison, 78] utilisées dans de nombreux domaines comme la Théorie des Langages et le Traitement de la Langue Naturelle, ou les grammaires d'unification [Gazdar, 89 ; Miller, 90].

---

[1] Pour une topologie des modèles de langage, nous renvoyons à [Dupond, 96] qui constitue notre référence première pour ce paragraphe.

Enfin, les grammaires hors-contexte stochastiques permettent d'ajouter une probabilité sur les règles d'une grammaire à structure de phrase [Dupond, 96].

## 1.4.3. Contraintes linguistiques syntaxiques et décodage acoustique

Dans les SRAP, les contraintes linguistiques syntaxiques sont, en général, modélisées de deux manières différentes. D'une part, elles peuvent être appliquées *a priori* en limitant le graphe de recherche au moment du décodage (les connections possibles entre les modèles de mots) par l'intermédiaire d'une grammaire formelle. Mais cette approche se limite aux applications où le langage peut être contraint. D'autre part, les algorithmes de reconnaissance de mots connectés peuvent prendre en compte des contraintes syntaxiques sous la forme de modèles probabilistes. Notons que ces deux approches peuvent également être combinées. Par ailleurs, dans la plupart des SRAP actuels, une extension de ces algorithmes permet de générer des treillis de mots possibles plutôt que l'unique meilleure solution ; les connaissances syntaxiques pouvant alors être également appliquées dans un processus *a posteriori*[1].

## 1.4.4. Remarques sur l'implémentation

Après avoir présenté les principes généraux de la modélisation Markovienne des probabilités acoustiques et linguistiques, nous décrivons dans ce chapitre, deux éléments remarquables utilisés dans la plupart des implémentations de SRAP.

### 1.4.4.1. Changement d'échelle et accélération des calculs

L'une des particularités des probabilités acoustiques (cf. équation 1.6), linguistiques (cf. équation 1.14) et de leur combinaison (cf. équation 1.3) est leur faible valeur numérique. Par ailleurs, ces équations impliquent que la valeur de la probabilité finale calculée lors de la phase de décodage d'un SRAP décroît en fonction du nombre d'observations acoustiques à traiter. Le problème est que cette valeur finale est si faible qu'elle sort des possibilités de codage numérique habituelle des ordinateurs.

La méthode employée pour résoudre ce problème est de transposer l'ensemble des calculs dans le domaine logarithmique. Les probabilités deviennent alors des probabilités logarithmiques ou *logprob*. De plus, les propriétés du domaine logarithmique permettent de transformer la multiplication de probabilités en addition de *logprob*, ce qui a pour effet une diminution des temps de calcul[1].

---

[1] À propos de l'intégration des modèles de langage dans les algorithmes de décodage, voir [Ney, 97]. Le chapitre suivant *1.5. Stratégie de décodage* traite de la combinaison des connaissances dans les SRAP.

L'équation initiale de l'approche globale probabiliste (cf. équation 1.3) peut alors s'écrire :

$$\hat{M} = \underset{M}{Argmax}\ LogP(O/M) + LogP(M) \qquad \text{(Eq. 1.15)}$$

### 1.4.4.2. Facteurs de pénalités

Dans le chapitre *1.3. L'approche globale probabiliste*, nous avons défini le principe probabiliste de la reconnaissance de la parole comme le calcul, puis le produit des probabilités acoustiques et linguistiques. Les équations 1.3 et 1.15 étant les représentations formelles de ce processus de reconnaissance. Lors de la phase de mise au point d'un SRAP, il est nécessaire de pouvoir agir sur les poids de ces probabilités afin de moduler les influences acoustiques et/ou linguistiques dans le processus de décodage. Ceci peut être réalisé par l'ajout de facteurs de pénalités.

L'équation 1.15 devient alors :

$$\hat{M} = \underset{M}{Argmax}\ LogP(O/M) \times fa + LogP(M) \times fl + fm \qquad \text{(Eq. 1.16)}$$

Avec :

> Le terme *fa* représente le facteur de pénalité acoustique ; il détermine l'influence de la probabilité acoustique.
>
> Le terme *fl* représente le facteur de pénalité linguistique ; il détermine l'influence de la probabilité linguistique.
>
> Le terme *fm* représente le facteur de pénalité mot ; il influe sur le degré de découpage en mots.

## 1.4.5. Commentaires

Les mécanismes de base utilisés pour la modélisation Markovienne des probabilités acoustiques et linguistiques sont abondamment décrits dans la littérature. Par ailleurs, il existe des ensembles logiciels proposant ces fonctionnalités, ce qui peut grandement faciliter leur mise en œuvre [Boite, 97 ; Young, 96 ; Clarkson, 97].

---

[1] L'addition étant une opération plus rapide que la multiplication sur la plupart des calculateurs.

En fait, comme nous le verrons par la suite, la performance des modélisations acoustique et linguistique dépend essentiellement de la qualité (pertinence de l'alignement, initialisation des modèles, taille du corpus, etc.) des données utilisées pour l'apprentissage et des choix effectués *a priori* (unité phonétique, topologie, densité de probabilité, etc.).

Dans le cadre de la modélisation Markoviennne des probabilités acoustiques, après avoir défini la topologie, le processus de reconnaissance et le type d'apprentissage pour les HMM, la réalisation d'un module acoustique probabiliste nécessite de disposer d'un ensemble de données telles que des corpus acoustiques, une phonétisation de référence, une paramétrisation, un alignement de référence, etc.

Nous présentons, dans la deuxième partie de ce document, les éléments nécessaires à l'obtention de ces données dans un processus automatique.

## 1.5. Stratégies de décodage en parole continue

Jusqu'à présent, nous nous sommes attachés à décrire les méthodologies utilisées dans le domaine du traitement automatique de la parole. Ce point de vue nous a permis de faire ressortir les différentes techniques employées pour la reconnaissance de segments de parole. Mais, quelles que soient les méthodologies adoptées, les SRAP modernes doivent pouvoir traiter la parole dans sa continuité, comme un enchaînement de segments élémentaires (les mots dans une phrase, les phonèmes dans un mot, etc.). Pour cela, il est nécessaire de combiner différentes sources et différents niveaux de connaissances (acoustique, linguistique, phonétique, phonologique, sémantique, etc.) : c'est le rôle de la stratégie de décodage.

Nous rappelons que sur le plan des connaissances acoustiques, ces systèmes peuvent principalement utiliser deux types d’approches globales : la programmation dynamique (Dynamic Time Warping ou DTW) [Sakoe, 79 ; Myers, 81] et les chaînes de Markov cachées qui sont maintenant employées dans la grande majorité des SRAP [Rabiner, 85 ; Mérialdo, 88b ; Jouvet, 88 ; Young, 89 ; Lee, 90a ; Cerf-Danon, 91 ; Schwartz, 92, Gauvain, 93]. Sur le plan des connaissances linguistiques, la plupart des systèmes disposent de grammaires formelles [Young, 89] et/ou de grammaires probabilistes [Jelinek, 82 ; El-Bèze, 90]. Ajoutons enfin que les connaissances phonétiques et phonologiques peuvent être modélisées à l'aide de lexiques.

### 1.5.1. Stratégies de décodage fondatrices

L'étude des SRAP met en évidence l'importance des stratégies de décodage. Certaines d'entre elles, mises au point dans les années quatre-vingt, sont abondamment citées dans la littérature. Nous nous proposons d'étudier trois de ces stratégies [Igounet, 95].

Afin de combiner les différentes sources et les différents niveaux de connaissances, nous avons le choix entre deux types de stratégies : les stratégies synchrones et les stratégies asynchrones [Godin, 89 ; Schwartz, 92 ; Méloni, 95]. Les stratégies synchrones utilisent simultanément toutes les connaissances afin de trouver l'hypothèse la plus probable [Jelinek, 82 ; Young, 89 ; Gauvain, 93]. Par opposition, les stratégies asynchrones exploitent les connaissances *petit à petit* afin de réduire localement l'espace de recherche. Dans ce dernier cas, ces connaissances sont combinées les unes à la suite des autres dans un processus multi-passes [Mérialdo, 88b ; Schwartz, 92].

Pour résoudre le problème de la reconnaissance en parole continue, différentes approches ont été développées. Ces approches, plus ou moins complexes, se doivent de conserver l'optimalité de la solution. Elles comparent chaque portion de la phrase à reconnaître à toutes les références possibles pour tous les instants possibles [Jouvet, 88].

À l'instar de [Godin, 89], nous dénombrons trois stratégies principales :
- *Une Passe* (One Pass) [Bridle, 82],
- *Niveaux Successifs* (Level Building) [Myers, 81],
- *Deux Niveaux* (Two Level) [Sakoe, 79].

Ces trois stratégies conservent l'optimalité de la solution. Elles diffèrent seulement par leur complexité[1], leur temps de calcul et leur capacité à intégrer d'autres types de contraintes comme, par exemple, des connaissances linguistiques [Godin, 89].

Nous nous proposons de rappeler les principes qui sous-tendent ces trois stratégies "fondatrices" en montrant ce qui les unit et ce qui les oppose. Pour des raisons de formulation, nous les exposerons dans l'ordre chronologique inverse de leur apparition dans la littérature.

**Définition**

Soit un signal de parole inconnu paramétrisé et constitué de $i=1,...,N$ trames de temps (une trame de temps est représentée par un vecteur de paramètres). La phrase à reconnaître est composée d'unités de base, appelées mots, choisis dans un dictionnaire. Ces mots correspondent à un ensemble de $K$ références dont les modèles se distinguent par l'index $k=1,...,K$. Les unités de temps du modèle $k$ sont données par $j=1,...,J(k)$ où $J(k)$ est la dernière unité de temps du modèle $k$.

L'objectif de la reconnaissance est de déterminer la séquence optimale de mots qui correspond au signal de parole inconnu.

---

[1] Pour une étude de ces algorithmes du point de vue de la complexité, se référer à [Ney, 84].

### 1.5.1.1. Algorithme *Une Passe*

C'est en 1971 que Vintsyuk [Vintsyuk, 71] introduit pour la première fois cet algorithme qui a été implémenté et testé par Bridle [Bridle, 82]. Les articles de référence pour cet algorithme sont [Bridle, 82 ; Ney, 84 ; Godin, 89 ].

> Soit la trame *i* de la chaîne et la trame *j* du modèle de référence *k* qui définissent un ensemble de points *(i, j, k)*. Au temps *i*, la colonne correspondante est composée des points *(i, j, k)* pour *i* donné, avec *j* variant de *1 à J(k)* et *k* variant de *1 à K*. Nous devons trouver le meilleur chemin à travers cet ensemble de points (pour une distance donnée) en respectant certaines règles de transition entre les points : les règles de transition interne aux modèles de référence et les règles de transition entre les frontières des modèles de référence. Le calcul se termine lorsque la dernière trame de la séquence inconnue est atteinte. Le meilleur chemin est alors obtenu par chaînage arrière ; la solution correspond à la concaténation des mots références (cf. figure 1.7).

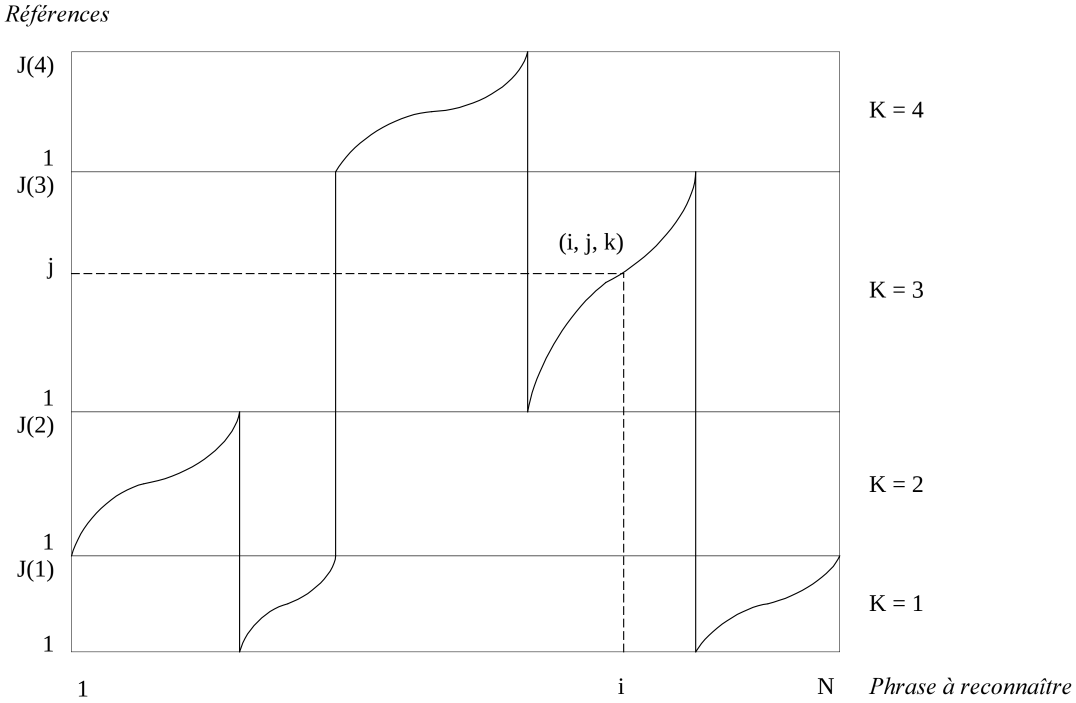


Figure 1.7 : Chemin optimal pour la stratégie *Une Passe*

### 1.5.1.2. Algorithme *Niveaux Successifs*

Cet algorithme a été proposé par [Myers, 81] afin de réduire la complexité de l'algorithme *Deux Niveaux* (cf. *1.5.1.3. Algorithme Deux Niveaux*). Les articles de référence pour cet algorithme sont [Myers, 81 ; Godin, 89].

Soit $L$ le nombre maximum de niveaux qui doit être connu. Le principe de l'algorithme *Niveaux Successifs* est de construire d'une manière récursive les différents niveaux (cf. figure 1.8). Au premier niveau, l'algorithme trouve la meilleure unité de référence $U_1(i)$, parmi les $K$ références, qui commence à la trame $1$ et termine à la trame $i$ pour $i=1,...,N$. Nous remarquons que cette unité peut déjà être le résultat final du décodage si elle se termine à la trame $N$. Puis pour le second niveau, en connaissant ce qui s'est passé au premier, l'algorithme calcule l'unité $U_2(i)$ pour tous les $i$, avec $U_1(i')$, $i'<i$ et $i$ trame de fin. $U_2(i)$ s'aligne donc de la trame $(i'+1)$ jusqu'à la trame $i$. Plus généralement, au niveau $p$, *Niveaux Successifs* trouve pour tout $i$, ce que peut être la meilleure solution $U_p(i)$. C'est-à-dire que, pour chaque niveau $p=1,...,L$, la meilleure séquence d'unités de référence est calculée.

**Remarque**

La stratégie de l'algorithme *Niveaux Successifs* recherche la meilleure séquence d'unités qui contient, au plus, $L$ unités de référence. Si la solution optimale du même problème par l'algorithme *Une Passe* est obtenue par l'algorithme *Niveaux Successifs* au niveau $L^*$, les solutions sont alors identiques pour tout $p>L^*$. En effet, ces deux algorithmes résolvent le même problème : l'algorithme *Une Passe* recherche la séquence optimale d'unités et l'algorithme *Niveaux Successifs* recherche la séquence optimale de au plus $L$ unités [Godin, 89].

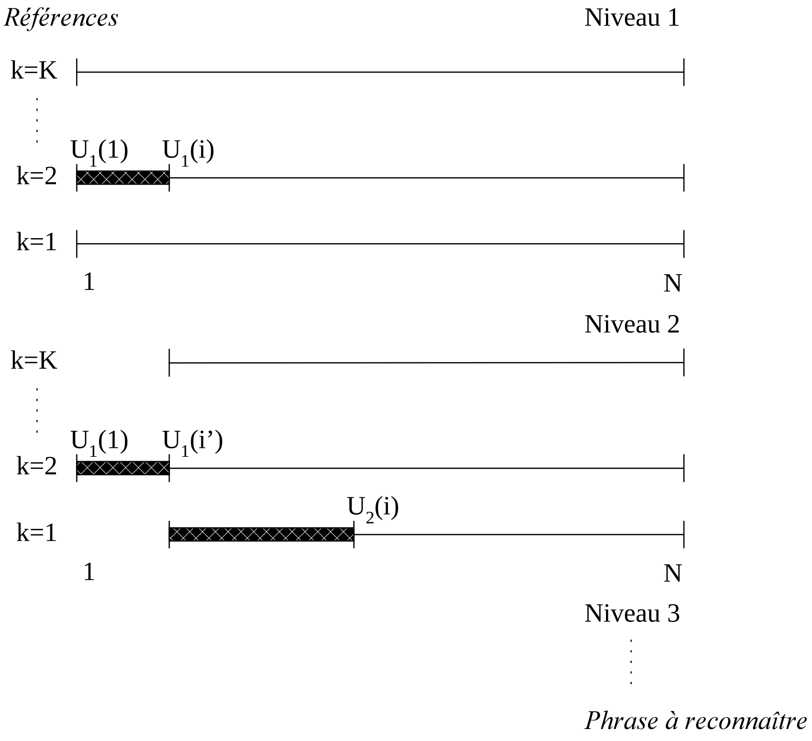

Références
Niveau 1
k=K
U1(1)
U1(i)
k=2
k=1
1
N
Niveau 2
k=K
U1(1)
U1(i')
k=2
U2(i)
k=1
1
N
Niveau 3
Phrase à reconnaître

Figure 1.8 : Chemin optimal pour la stratégie *Niveaux Successifs*

### 1.5.1.3. Algorithme *Deux Niveaux*

L'algorithme *Deux Niveaux* a été introduit par [Sakoe, 79 ] comme une extension à la programmation dynamique (DTW) pour la reconnaissance de mots connectés. Comme son nom l'indique, l'algorithme *Deux Niveaux* est composé de deux niveaux de calcul : le niveau du mot et celui de la phrase. Le processus agissant au niveau de la phrase cherche la meilleure séquence de mots en combinant les hypothèses du processus agissant au niveau des mots. Les articles de référence pour cet algorithme sont [Sakoe, 79 ; Godin, 89].

> Soit $d$ et $f$, respectivement le début et la fin d'un ensemble de trames, et $D(d, f, k)$ la distance cumulative entre les trames $d$ et $f$ de la chaîne de test et une unité de référence $k$. Dans le premier niveau (niveau mot), la distance $D(d, f, k)$ est évaluée pour $1 \leq d \leq f \leq N$ et $1 \leq k \leq K$ ainsi que la meilleure distance telle que :
>
> $$D_*(d, f) = \min_{1 \leq k \leq K} \{D(d, f, k)\}$$
>
> Cette distance doit être calculée pour toutes les références quelles que soient les trames de début et de fin. Dans l'algorithme précédent, *Niveaux Successifs*, la trame de début évolue en fonction du niveau ce qui diminue le nombre de comparaisons de trames nécessaire au processus de décodage.
>
> Puis, dans le second niveau (niveau phrase) la *séquence-p* optimale de mots références est calculée pour $p = 1, \ldots, L$ (à l'instar de la stratégie *Niveaux Successifs*, le maximum de mots à reconnaître doit être défini au préalable) :
>
> $$\text{pour tout } f \in [1,N],\ D_{*,p}(f) = \min_{1 \leq k \leq K} \{D_*(d, f) + D_{*,p-1}(d-1)\}$$
>
> Le résultat correspond à la meilleure *séquence-L* donnée par $D_{*,L}(N)$ :
>
> $$L = \arg\min_{p} \{D_{*,p}(N)\}$$

### 1.5.1.4. Résumé

Résumons maintenant ces trois stratégies de décodage dans le sens plus naturel de l'optimisation des calculs [Jouvet, 88] :

> Tout d'abord, nous avons l'algorithme *Deux Niveaux* qui effectue une programmation dynamique à deux niveaux. Le premier niveau consistant à calculer les distances entre des segments de la phrase à reconnaître et toutes les

références des mots. Le niveau supérieur détermine alors, à partir de ces valeurs, la meilleure séquence possible pour la phrase complète.

L'algorithme *Niveaux Successifs* consiste à rechercher d'abord le meilleur mot correspondant au début de phrase sur toutes les trames possibles. Ensuite, il détermine les meilleurs seconds mots, ce qui permet de connaître pour chaque segment de parole, le meilleur début de phrase correspondant. En répétant la démarche jusqu'à la fin de la phrase, on obtient la meilleure solution possible. Cette méthode nécessite moins de comparaison de trames et moins de mémoire que l'approche *Deux Niveaux*.

Enfin, avec l'algorithme *Une Passe*, au lieu de déterminer la meilleure séquence mot par mot, on peut traiter toutes les références en parallèle et progresser de manière synchrone avec les trames de la phrase à reconnaître. Cet algorithme est plus efficace que ceux mentionnés précédemment. De plus, il se prête mieux à une application temps réel.

### 1.5.2. Le système CUED-HTK et le Modèle de Propagation de Jeton (MPJ)

La stratégie de décodage dite Modèle de Propagation de Jeton (ou *Token Passing Model*) est proposée par S. Young, N. Russel et J. Thornton du département d’ingénierie de l’université de Cambridge [Young, 89]. Cette stratégie est à la base des SRAP du Cambridge University Engineering Departement (CUED) et du Package Hidden markov model ToolKit (HTK) [Young, 93] que nous utilisons au LIA dans le cadre de la réalisation de SRAP.

Le Modèle de Propagation de Jeton (MPJ) propose de voir la reconnaissance de la parole continue comme un processus de décodage qui s’appuie sur le passage de jetons dans un réseau de transitions. Ce point de vue permet notamment de guider la reconnaissance par un ensemble de contraintes grammaticales modélisées au travers de ce réseau de transitions.

Cette stratégie est indépendante du type de méthodologie employée pour la reconnaissance des unités de base. De ce fait, nous pouvons indifféremment appliquer la stratégie du MPJ avec une technologie de type alignement dynamique (DTW) ou bien avec des modèles de Markov cachés (HMM).

### 1.5.2.1. Reconnaissance de mots isolés

Dans un premier temps, nous allons nous intéresser à la reconnaissance d'unités isolées. Dans le MPJ, les unités du vocabulaire sont représentées par des modèles sous la forme d'automates d'états finis classiques (figure 1.9).

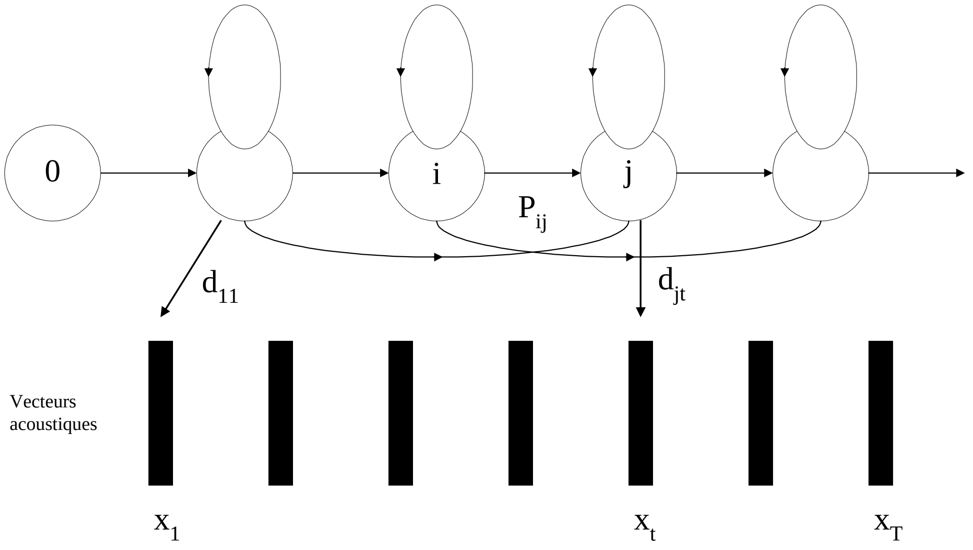


Figure 1.9 : Structure d'un modèle

À chaque paire d'états connectés *i* et *j* de l'automate, est associé un coût de transition noté $p_{ij}$. De plus, chaque état *j* porte également une fonction de coût local notée $d_j(t)$. Le signal de parole étant, au préalable, paramétrisé sous une forme habituelle d'une séquence de vecteurs acoustiques notés $x_1 ... x_T$, cette fonction $d_j(t)$ représente le coût de se trouver dans l'état *j* au temps *t* et donc sur le vecteur associé $x_t$. Chaque séquence possible d'états à travers le modèle - $i = i_0, i_1, ..., i_T$, où $i_0$ est l'état initial - représente un alignement possible du modèle avec le signal de parole et $s(i)$ le coût total de cet alignement.

En reconnaissance de la parole, la ressemblance entre un modèle et le signal de parole est supposée être inversement proportionnelle au coût minimum de l'alignement. Soit $s_j(t)$ le coût minimum de l'alignement entre le segment $x_1 ... x_t$ du signal de parole inconnu et le modèle de référence d'état initial $i_0$ et d'état final $i_T$.

**Remarque**

Cette manière d'exposer le MPJ est très proche de la présentation classique des modèles de Markov cachés. Cependant, nous maintenons la distinction afin de souligner que le MPJ ne s'applique pas seulement aux modèles de Markov cachés. Il peut également être utilisé avec d'autres méthodologies de reconnaissance de formes telles que la DTW.

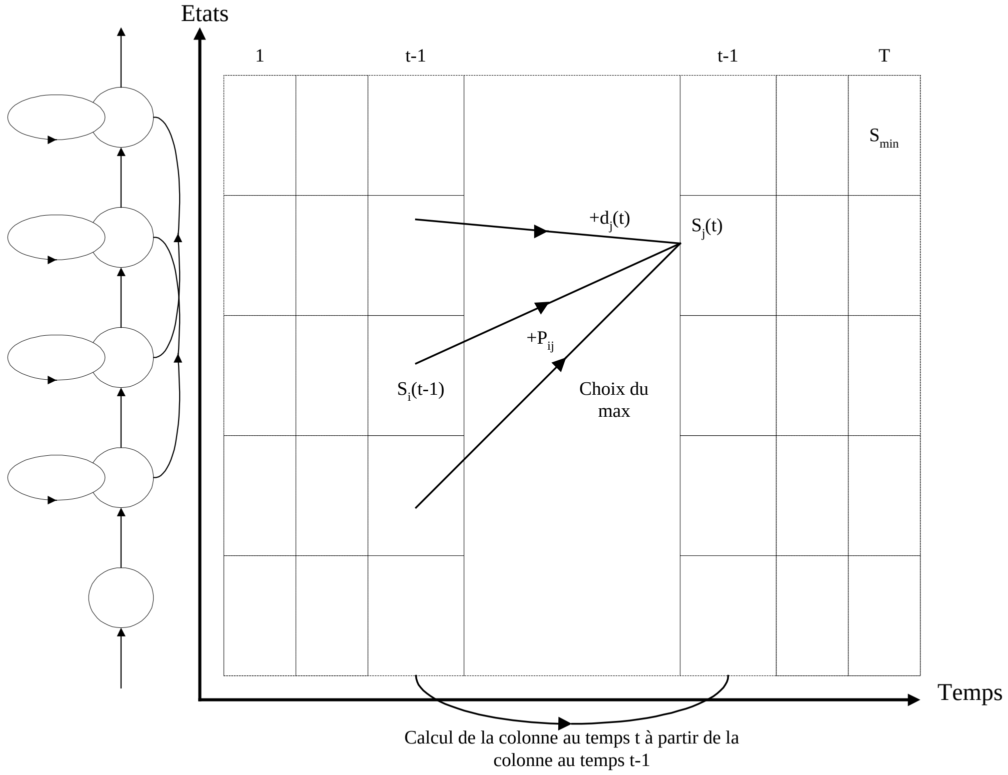


Figure 1.10 : Matrice du calcul des coûts

Habituellement, le coût minimum de l'alignement $s_j(t)$ se calcule au moyen d'une matrice (figure 1.10) dont les éléments sont alors calculés colonne par colonne. L'idée principale du MPJ est que ces calculs peuvent être effectués par le biais d'un jeton passé d'état en état.

Chaque état du modèle de mot porte un jeton mobile. Dans un premier temps, ces jetons ne servent qu'à conserver la valeur du coût de l'alignement. C'est-à-dire qu'au temps $t$, le jeton dans l'état $j$ contient la valeur $s_j(t)$. De cette façon, l'algorithme classique de calcul des coûts peut être reformulé de la façon suivante :

```
Initialisation :
    Les jetons des états initiaux sont initialisés à 0 ;
    Les jetons des autres états sont initialisés à l'infini ;

Algorithme :
    pour $t:=1$ à $T$ faire
        pour chaque état $i$ faire
            Passer une copie du jeton dans l'état $i$ à tous les états connectés, incrémenter sa valeur $s$ par
            $p_{ij}+d_j(t)$ ;
        fin ;
        Effacer le jeton original ;
        pour chaque état $i$ faire
            Trouver le jeton dans l'état $i$ avec la plus petite valeur $s$ et effacer le reste ;
        fin ;
    fin ;
Résultat :
```

> Sur tous les états finaux, le jeton qui contient la plus petite valeur *s* est celui qui donne le coût minimum de reconnaissance.

L'algorithme du MPJ est illustré dans la figure 1.11 Il correspond à une stratégie de recherche synchrone dans le temps où les jetons sont propagés à chaque cycle de la boucle principale.

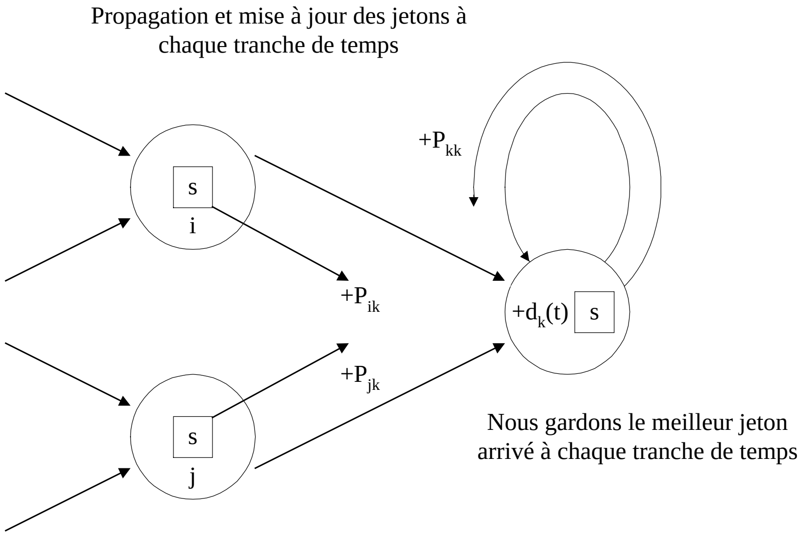

Propagation et mise à jour des jetons à chaque tranche de temps
+P_kk
s
i
+P_ik
+d_k(t)
s
+P_jk
s
j
Nous gardons le meilleur jeton arrivé à chaque tranche de temps

Figure 1.11 : Algorithme de base du Modèle de Propagation de Jeton

Il faut noter ici qu'il est possible d'avoir des états initiaux et terminaux multiples ; cela n'est pas nécessaire pour la reconnaissance d'unités isolées mais c'est utile dans le cas de la reconnaissance de mots connectés.

#### 1.5.2.2. Reconnaissance de mots connectés

À partir de la stratégie de reconnaissance de mots isolés décrite dans le chapitre précédent, l'extension à la reconnaissance de mots connectés est triviale. Les modèles de mots isolés sont simplement connectés entre eux dans un modèle composite en boucle (figure 1.12) . Ils forment un réseau syntaxique sur lequel le même algorithme de passage de jeton peut toujours s'appliquer. La seule différence est la prise en compte de beaucoup plus d'informations pour calculer le score du meilleur jeton quand il atteint la fin du signal de parole.

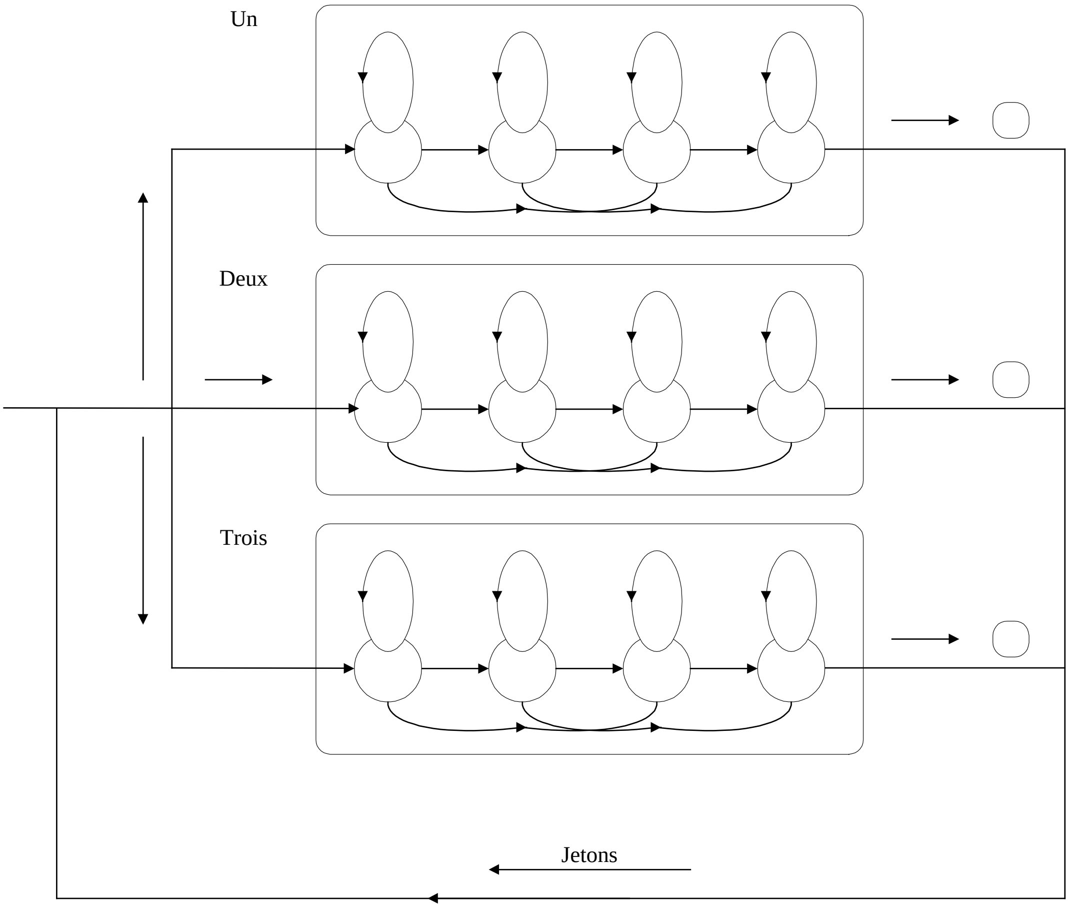


Figure 1.12 : Reconnaissance de mots connectés avec le
Modèle de Propagation de Jeton

Enfin, il faut également retrouver la meilleure séquence de mots et pas seulement le coût d'alignement total. Pour cela, il suffit de mettre en place un processus de chaînage qui permet de sauvegarder les frontières et la suite des mots reconnus.

#### 1.5.2.3. Prise en compte des contraintes syntaxiques dans le MPJ

Dans la section *1.4.3. Contraintes linguistiques syntaxiques et décodage acoustique,* nous présentons les principes généraux pour l'application de contraintes linguistiques syntaxiques dans le cadre d'un SRAP. La stratégie du MPJ permet de prendre en compte des contraintes syntaxiques sous forme de grammaire formelle et/ou de modèles de langage probabiliste. Dans le cas d'une grammaire formelle, il suffit tout simplement de compiler les règles dans un réseau syntaxique *ad hoc* (cf. figure 1.13). Quant au modèle de langage probabiliste, ce sont les transitions du réseau syntaxique qui sont utilisées pour porter les fréquences relatives de chaque occurrence de mot. De cette façon, l'algorithme de base du MPJ peut toujours s'appliquer et les contraintes syntaxiques sont intégrées au cœur même de l'algorithme de décodage.

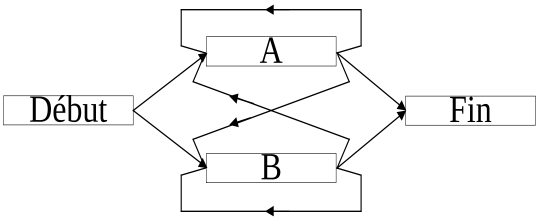


Figure 1.13 : Réseau syntaxique et modèle de langage probabiliste bi-gramme

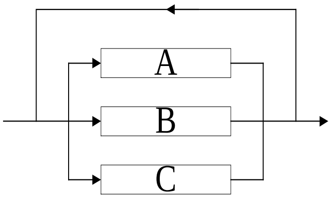
Nous notons que cette approche est très coûteuse du point de vue du temps de calcul et de l'espace mémoire quand le SRAP est à grand vocabulaire. Dans ce cas, le nombre de transitions nécessaires est une fonction de la taille du vocabulaire et de l'ordre du ML.

#### 1.5.2.4. Relation du MPJ avec les stratégies de décodage fondatrices

La relation du MPJ avec les stratégies fondatrices décrites en section *1.5.1. Stratégies de décodage fondatrices* est triviale. En effet, l'application du MPJ sur le modèle en boucle des figures 1.12 et 1.14(a) est équivalent à une recherche de chemin de type *Une Passe* (cf. *figure 1.7 Chemin optimal pour la stratégie Une Passe*). De la même façon, l'algorithme *Niveaux Successifs* peut être représenté en connectant les unités de base dans une séquence gauche-droite comme dans la figure 1.14(b).

Ici, ce sont donc des contraintes syntaxiques traduites dans la topologie du réseau des connexions des unités de base qui déterminent l'adoption de l'une ou l'autre des stratégies. De façon générale, nous pouvons dire que les algorithmes *Une Passe* et *Niveaux Successifs* peuvent être considérés comme des cas particuliers du *Modèle de Propagation de Jeton* selon les contraintes syntaxiques que l'on y introduit.

(a) Une Passe



(b) Niveaux Successifs

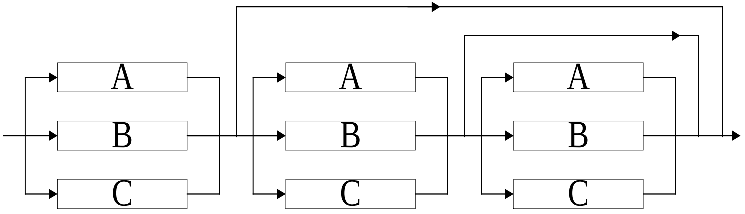


Figure 1.14 : Topologie des algorithmes *Une Passe* et *Niveaux Successifs*

### 1.5.3. Autres SRAP

Nous présentons maintenant un tour d'horizon de quelques systèmes SRAP afin de donner une idée de leur stratégie de décodage. Nous nous intéressons particulièrement à leur façon d'intégrer l'ensemble des connaissances (systèmes asynchrone et synchrone, multi-passes, modèles acoustiques, modèles de langage, etc.).

#### 1.5.3.1. Le système BYBLOS (BBN)

BYBLOS [Schwartz, 92] est un système asynchrone fondé sur le paradigme *N-Best* originalement développé afin de faciliter l'intégration de connaissances linguistiques pour la RAP. Le principe de ce système est d'intégrer les différentes sources de connaissances les unes à la suite des autres. Dans un premier temps, un modèle de Markov caché phonétique et un modèle de langage probabiliste de type bi-gramme sont fusionnés. Puis, le même principe est appliqué à de nouvelles sources de connaissances. En fait, la fusion s'obtient par le produit pondéré des scores des différentes sources de connaissances. L'optimisation des paramètres du système étant assurée par une suite d'expériences de reconnaissance qui visent à réduire le taux d'erreur.

#### 1.5.3.2. Le système du LIMSI

Dans le SRAP du LIMSI [Gauvain, 93; Lamel, 95 ; Adda, 97a], la modélisation acoustique est assurée par des unités de base constituées de phones en contexte qui permettent de former le lexique par concaténation. Chaque phone en contexte est modélisé par une chaîne de Markov cachée à 3 états, avec des densités d'observation multi-gaussiennes. Les matrices de covariance sont supposées diagonales. Le décodage utilise l'algorithme de Viterbi en parcourant l'espace de recherche de manière synchrone dans une stratégie de type *Une Passe*. Mais cette approche ne pouvant fonctionner sur de grands vocabulaires (l'espace de recherche devient trop important), une technique de décodage progressif est alors utilisée pour réduire le nombre d'hypothèses à évaluer au fur et à mesure des connaissances employées. Celles-ci deviennent plus précises et requièrent des modèles plus complexes. Ainsi, la première passe du décodage n'utilise que les bi-grammes et des modèles acoustiques de taille réduite pour donner, en résultat, un treillis de mots issus des solutions partielles de la meilleure solution. À partir de ce treillis, on peut obtenir le nouvel espace de recherche qui servira à la seconde passe du décodage avec des modèles (acoustiques et linguistiques) plus coûteux et plus puissants.

#### 1.5.3.3. Le système PARSYFAL (IBM)

Dans PARSYFAL [Mérialdo, 87 ; Mérialdo, 88 ; El-Bèze, 90 ; Marcadet, 91], un mot est décrit dans le dictionnaire comme une suite de syllabes représentées par des unités phonétiques. Chaque unité phonétique de base est modélisée par une source de Markov. Les modèles de Markov utilisés ici sont des modèles de Markov discrets (quantification vectorielle des vecteurs acoustiques). Le décodage est un Décodage Multi-Niveaux

(Multi-Level Decoding, MLD) qui permet d'utiliser de très grands vocabulaires. Il est fondé sur une reconnaissance acoustique d'unités sub-lexicales (pour PARSYFAL des syllabes), une composition d'unités successives pour former des mots à partir d'un dictionnaire et une composition des mots successifs pour former des phrases partielles en utilisant les contraintes d'un modèle linguistique. L'une des originalités du Décodage Multi-Niveaux est, sans aucun doute, la détection des frontières des unités sub-lexicales. Contrairement aux stratégies de type synchrone où le calcul des frontières est implicite, la solution adoptée ici consiste à retenir comme fin de syllabe la tranche de temps *t* qui correspond à la probabilité maximale que cette syllabe se termine en *t*.

#### 1.5.3.4. Le système TANGORA (IBM)

Le système TANGORA [Cerf-Danon, 91, Marcadet, 91] a été créé initialement pour l'anglais par l'équipe de Jelinek. Par la suite, d'autres équipes d'IBM l'ont adapté à différentes langues (allemand, espagnol, français, italien). Son architecture peut être décomposée en quatre modules : un pré-filtre acoustique (Fast-Match) qui utilise des modèles phonétiques et permet de créer une première liste "grossière" de mots candidats ; un modèle de langage (tri-gramme) qui réorganise la liste de mots ; un module de décodage qui ré-estime et pondère (avec les résultats des modules précédents) les probabilités de la liste ; enfin, un module qui coordonne les opérations, fixe le choix final et gère la communication avec l'utilisateur. Ce dernier module est, également, celui qui dirige la stratégie de décodage. L'algorithme fonctionne de gauche à droite en combinant les divers scores obtenus aux niveaux précédents. À chaque instant, la meilleure hypothèse présente non étendue est développée. C'est l'algorithme dit du décodage à pile (stack decoding).

#### 1.5.3.5. Les systèmes SPHINX I et II (CMU)

SPHINX [Lee, 90a] est un système qui utilise les techniques de RAP classiques fondées sur les chaînes de Markov cachées (stratégie de type synchrone, HMM discret, etc.). Afin d'améliorer les résultats, le système de base est enrichi de nouvelles connaissances (paramétrisation, modèle de durée, modèle de langage), de modèles contextuels (triphones avec groupement automatique) et de mots fonctionnels (ce sont des mots "difficiles" et suffisamment fréquents pour être modélisés par une chaîne de Markov spécifique).

Le système SPHINX II [Hwang, 93a] propose des améliorations et solutions à certains problèmes, notamment ceux posés par les triphones non rencontrés dans les bases de données d'apprentissage. Pour pallier cette difficulté, l'équipe de recherche de SPHINX II propose les notions de *senone* et d'Arbre Senonique de Décision. Par ailleurs, dans le cadre des SRAP et pour de grands vocabulaires, quand l'espace de recherche devient trop important, le décodage est alors multi-passes en trois étapes : la première passe est un algorithme gauche-droite de Viterbi standard couplé à un bi-gramme qui produit des fins de mots et des scores ; la seconde passe, guidée par les résultats de la première, est un algorithme droite-gauche de recherche qui produit les débuts de mots et de nouveaux scores ; la troisième et dernière passe est un algorithme de recherche A* qui combine les résultats des deux passes précédentes avec un modèle de langage. Enfin, l'*Unified Stochastic Engine* est proposé pour traiter le problème de l'optimisation en combinant les modèles acoustiques et les modèles de langage lors de la phase d'apprentissage.

## 1.6. Conclusion

Dans cette première partie, nous avons défini les bases théoriques utilisées dans les SRAP en différenciant la méthodologie de reconnaissance de la stratégie de décodage.

En ce qui concerne la méthodologie de reconnaissance, un accent particulier a été mis sur l'approche globale probabiliste employée par la suite. Il résulte de cette étude que les mécanismes de base utilisés pour la modélisation Markovienne des probabilités acoustiques et linguistiques sont abondamment décrits dans la littérature ; il existe d'ailleurs des ensembles logiciels les implémentant. C'est donc essentiellement de la qualité des données utilisées, des choix effectués *a priori* et de l'ensemble des ressources disponibles dont dépend la performance d'un SRAP.

En ce qui concerne les stratégies de décodage, l'étude des systèmes de reconnaissance modernes a montré que les stratégies de décodage actuelles se sont grandement complexifiées afin de répondre aux nouvelles exigences (grand vocabulaire, multi-locuteur, résultat de type N meilleures solutions, etc.) et/ou pour faciliter l'intégration d'autres types d'informations (linguistiques, visuelles, etc.). Si les principes fondateurs permettant de traiter la parole continue restent les mêmes, c'est en fait, l'ajout de nouvelles connaissances qui permet d'améliorer les systèmes.

Mais, lors de la phase de décodage, les ressources informatiques (malgré les progrès technologiques constants) ne permettent plus de fusionner dans un traitement simultané et synchrone l'ensemble des connaissances acoustiques et linguistiques disponibles. Pour résoudre ce problème, la solution adoptée généralement consiste à utiliser des stratégies asynchrones et/ou des stratégies multi-passes. Par ailleurs, nous notons que les connaissances linguistiques (lexique, contraintes syntaxiques, modèles de langage probabilistes, etc.) jouent un rôle essentiel ; c'est particulièrement le cas des modèles de langage probabilistes dans les SRAP de type machine à dictée.

Enfin, dans leur grande majorité, les SRAP actuels utilisent l'approche globale probabiliste et une modélisation markovienne des probabilités acoustiques et linguistiques.

## Deuxième Partie

# Réalisation d'un système de reconnaissance : Décodage Acoustico-Phonétique

Nous commençons cette deuxième partie en précisant le cadre général de notre travail.

Puis, nous présentons la réalisation d'un système de Décodage Acoustico-Phonétique (DAP) fondé sur la modélisation Markovienne. Cette présentation sert de fil conducteur pour une description des différents éléments réalisés.

Ces éléments sont :

- des modules logiciels permettant la construction d'un SRAP (cf. *2.2. Outils de développement*),
- des corpus de données (cf. *2.3. Bases de données*),
- un module de phonétisation (cf. *2.4. Module de phonétisation*),
- une paramétrisation (cf. *2.5. Paramétrisation des données acoustiques*),
- un alignement phonétique de référence (cf. *2.6. Alignement des données acoustiques et phonétiques*),
- des modèles acoustiques (cf. *2.7. Modèles de Markov*),
- une procédure d'apprentissage (cf. *2.8. Apprentissage*).

Les résultats de reconnaissance du DAP permettent de valider l'ensemble de ces éléments (cf. *2.9. Reconnaissances et résultats*).

## 2.1. Cadre de l'étude

La réalisation d'un SRAP présuppose une définition précise de ses objectifs et caractéristiques. Dans ce chapitre, nous précisons le cadre général de notre travail, ses ambitions, mais aussi ses limites.

### 2.1.1. Le cadre

L'un de nos objectifs est de réaliser un SRAPC du français, indépendant du locuteur et sur un grand vocabulaire[1]. La réalisation d'un tel système est un problème difficile car les meilleurs SRAPC sont d'une grande complexité. Ils intègrent des technologies, des connaissances et des "savoir-faire" pluridisciplinaires développés par des équipes de recherche pendant de nombreuses années. De plus, ils nécessitent de disposer de ressources substantielles en matériel informatique et en corpus de données audionumériques (signal de parole) et linguistiques (lexique, modèle de langage, etc.). Pratiquement, la réalisation d'un SRAP implique le développement de nombreux modules (phonétisation, apprentissage, décodage, lexique, modèle de langage, etc.).

Par ailleurs, une validation et une évaluation objective des modules et du SRAP, dans son ensemble, sont nécessaires. Sans entrer dans l'étude des paradigmes de l'évaluation [Néel, 95], nous pouvons dégager deux approches [Barras, 96]. La première consiste à faire cette évaluation avec une base de données de référence admise par la communauté, sur laquelle existent des résultats équivalents. De cette façon, il est possible de comparer directement les performances du système. La seconde approche nécessite le développement préalable d'un système de référence. Le principe consiste alors à évaluer, dans les mêmes conditions, les performances du système développé au regard de celles du système de référence. Dans ce cas, ce sont les modifications apportées au système de référence qui sont évaluées. En 1982, G. Chollet et C. Gagnoulet définissaient[2] un SRAP de référence comme un système représentatif de l'état de l'art et facile à implémenter [Chollet, 82].

Cependant, en 1998, les techniques utilisées rendent la réalisation et l'évaluation d'un SRAP *état de l'art* plus difficile ; cette complexité croissante s'accompagne d'une augmentation significative de l'effort, tant humain que matériel, pour développer, améliorer, maintenir et évaluer[3] ce système. Ceci n'est possible que dans les équipes de recherche dépassant une certaine "taille critique". De ce fait, malgré un intérêt manifeste de la communauté parole pour un système de référence[1], il n'en existe pas actuellement.

---

[1] Dans la suite de ce document, le terme SRAPC désignera un Système de Reconnaissance Automatique de la Parole Continue, indépendant du locuteur et sur un grand vocabulaire.

[2] G. Chollet et C. Gagnoulet diffusaient également un système de référence. Ce système, fondé sur une DTW dont le code est en langage FORTRAN, permet la reconnaissance de mots isolés.

[3] En ce qui concerne l'évaluation des SRAP, se référer aux campagnes ARPA [Gauvain, 97] et à la première campagne AUPELF pour l'évaluation de systèmes de dictée vocale pour le français [Dolmazon, 97].

## 2.1.2. L'ambition

Pour notre part, nous avons choisi de développer et d'intégrer un certain nombre d'éléments nécessaires à la réalisation d'un SRAP. Pour des raisons de coût, le principe adopté d'une manière générale est une automatisation des processus de construction d’un SRAP.

Dans les chapitres de cette deuxième partie, nous décrivons les différents éléments permettant la réalisation du module de base de tout SRAP : le Décodage Acoustico-phonétique (DAP). À l'instar de [Lamel, 93a], nous pensons que l'évaluation d'un SRAP au niveau phonétique (c'est-à-dire au niveau acoustique) est nécessaire[2]. En effet, plus les modèles phonétiques sont performants, plus le système de reconnaissance correspondant est performant. Par ailleurs, cette évaluation permet de mettre en évidence certains problèmes acoustiques qui auraient pu être masqués par une analyse des résultats uniquement au niveau des mots. Ce point à une importance particulière dans le cas des SRAPC de type machine à dictée où des contraintes linguistiques de type n-grammes sont utilisées.

## 2.1.3. Les limites

Si le résultat final de notre travail est bien un SRAP opérationnel, ce système n'intègre pas toutes les subtilités du moment et ses performances ne sont pas directement comparables à un système dit *état de l'art*. En fait, il nous permet de valider les modules que nous avons développés pour la RAP. D'une certaine manière, il constitue le système de référence.

Notons également que l'évaluation des différents modules réalisés posent de nombreux problèmes. À l'exception de la campagne d'évaluation de l'AUPELF-UREF, il n'existe pas de référence qui permettrait de comparer objectivement des SRAP sur le français. Par ailleurs, le LIA n'a pas participé aux tests de RAP de cette campagne d'évaluation ; nous ne disposons donc pas de tous les corpus utilisés (il nous manque les corpus de test). De fait, les comparaisons des résultats obtenues avec d'autres SRAP comportent certains biais.

---

[1] Notons cependant que cinq équipes françaises de recherche (CRIN, ENST, IRISA, IRIT, LIA) étudient actuellement la faisabilité d'un système de référence : c'est le projet SIROCCO. Ce projet vise à développer, en partenariat, un système modulaire de reconnaissance automatique de la parole grand vocabulaire et indépendant du locuteur. Ce système est destiné à devenir une ressource logicielle commune permettant de reproduire des expériences calibrées dans des sites différents. De cette façon, il serait possible de tester certaines innovations sans avoir à développer et à maintenir un SRAP *état de l'art* dans sa globalité [Bimbot, 97a ; Bimbot, 97b].

[2] Et cela, même si depuis quelques années, l'évaluation des SRAP s'est focalisée sur les taux d'erreurs au niveau mot.

## 2.2. Outils de développement

Les outils logiciels utilisés pour le développement des SRAP présentés dans ce travail proviennent du kit HTK (Hidden markov model ToolKit). HTK est une boîte à outils logicielle destinée à la construction de SRAP fondés sur les modèles de Markov cachés [Young, 94a]. Il a été développé à l'Université de Cambrige par S. J. Young et son équipe. Il est distribué par *Entropic Research Laboratories Inc* [Young, 96]. HTK est à la base des SRAP du *Cambridge University Engineering Departement* (CUED) [Woodland, 92 ; Woodland, 95 ; Pye, 95]. Par ailleurs, ce kit de développement est relativement bien répandu dans le milieu de la recherche où il est considéré par la communauté parole comme un bon choix pour la construction de SRAP de référence [Chollet, 95]. Aussi, pour limiter la phase de développement, nous avons décidé d'adopter HTK comme base de départ pour la réalisation d'un SRAP.

| | |
|---|---|
| Hbuild | Conversion de modèles de langage dans différents types de format. |
| HcompV | Calcul de la moyenne et de la variance sur un ensemble de données d'apprentissage. |
| Hcopy | Calcul des paramètres de fichiers signaux. |
| HDMan | Édition des dictionnaires. |
| HERest | Phase d'apprentissage - Ré-estimation des HMM en continu (Baum-Welch). |
| HHEd | Édition des HMM. |
| Hinit | Phase d'apprentissage - Initialisation d'un HMM. |
| Hled | Édition des fichiers d'étiquettes. |
| Hlist | Visualisation en format texte des fichiers de données. |
| HLStats | Calcul de statistiques sur les étiquettes. |
| Hparse | Génération du graphe de décodage. |
| Hquant | Quantification vectorielle pour HMM discret. |
| Hrest | Phase d'apprentissage - Ré-estimation d'un HMM (Baum-Welch). |
| Hresults | Résultats du décodage (alignement dynamique entre les fichiers de résultats et de références). |
| HSGen | Générateur automatique de phrase en fonction d'une grammaire. |
| HSLab | Affichage du signal et des étiquettes. |
| Hsmooth | Lissage des paramètres des HMM. |
| Hvite | Décodage parole continue (Viterbi). |

Tableau 2.1 : Outils logiciels de base de HTK (Version 2.0)

HTK est constitué d'un ensemble de bibliothèques et d'outils logiciels écrit en langage C (cf. tableau 2.1). Sa modularité permet de développer ses propres logiciels et/ou d'utiliser directement les outils disponibles. Le kit de développement comprend des fonctions de traitement du signal, des outils d'apprentissage et de décodage fondés sur les HMM, la gestion des modèles de langage de type bi-grammes, la gestion des lexiques et un programme d'évaluation.

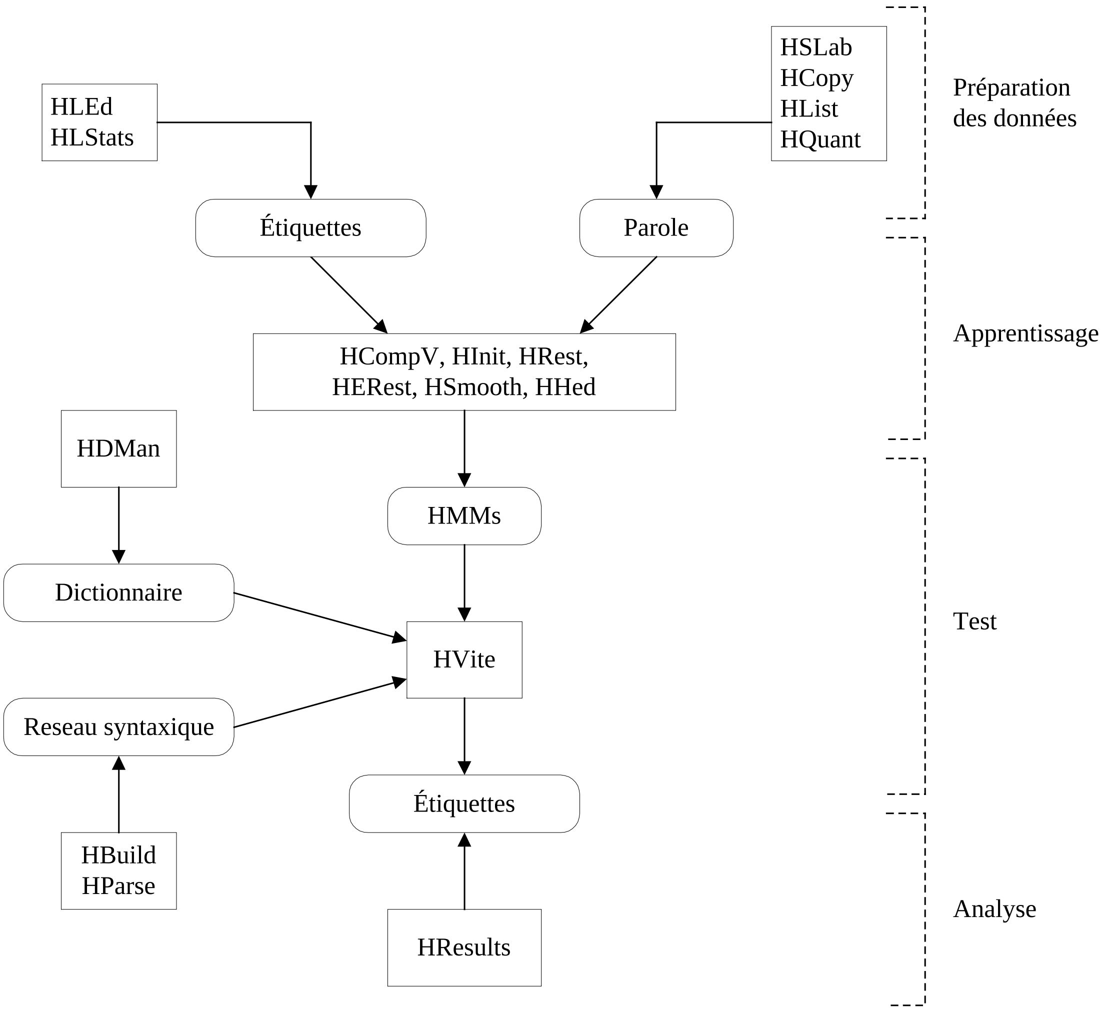

HLEd
HLStats
HSLab
HCopy
HList
HQuant
Préparation des données
Étiquettes
Parole
HCompV, HInit, HRest, HERest, HSmooth, HHed
Apprentissage
HDMan
HMMs
Dictionnaire
HVite
Test
Reseau syntaxique
Étiquettes
HBuild
HParse
HResults
Analyse

Figure 2.1 : Structure d'un SRAP avec HTK
(d'après The HTK Book, p. 20)

Les communications entre les outils logiciels de HTK se font par l'intermédiaire de fichiers de différents types (signaux, paramètres, étiquettes, description des HMM, définition des réseaux syntaxiques, modèles de langage) et les options d'utilisation sont transmises en argument de la ligne de commande. Cette approche facilite l'automatisation des liens entre modules grâce à la programmation de suites de commandes (dans notre cas, en C-Shell sous Hp-ux). Notons que nous avons conservé cette manière de faire pour les modules développés en interne au LIA. Les principaux outils d'HTK s'enchaînent naturellement (cf. figure 2.1) afin de réaliser les différentes étapes d'un SRAP. Ils permettent une grande liberté de choix sur l'ensemble des éléments constitutifs du système : les modèles acoustiques peuvent représenter tout type d'unité sub-lexicale (mot, phonème, etc.) ; ils permettent le partage d'états [Young, 94b] ; ils peuvent être contextuels et leur topologie totalement libre ; les distributions de probabilités associées aux états des HMM peuvent être discrètes ou continues et décrites par des multi-gaussiennes ; l'apprentissage des HMM est initialisé par l'algorithme de Viterbi, puis ré-estimé par l'algorithme de Baum-Welch ; le décodage est réalisé par l'algorithme de Viterbi sous la contrainte d'un réseau syntaxique et éventuellement d'un modèle de langage de type bi-gramme ; enfin, les résultats du décodage sont évalués par alignement dynamique avec les données de référence.

Pour une description détaillée de l'algorithme de décodage utilisé dans HTK, nous renvoyons au chapitre *1.5.2. Le système CUED-HTK et le Modèle de Propagation de Jeton.*

## 2.3. Bases de données

Les bases de données servant à l'apprentissage des modèles stochastiques (HMM, modèles de langage ou tout autre type de modèles probabilistes) sont des ressources essentielles. Pendant les 1$^{\text{ères}}$ Journées Scientifiques et Techniques (JST), lors d'une communication à propos des progrès réalisés en RAP, J. L. Gauvain déclarait : "une des principales sources de progrès est bien entendu la disponibilité de très grands corpus de parole à partir desquels il nous est possible d'estimer les paramètres de modèles acoustiques et de modèles de langage de plus en plus complexes (typiquement quelques millions de paramètres)" [Gauvain, 97]. En effet, en prenant pour exemple un SRAP à base de HMM, la précision de la modélisation acoustique est fonction du nombre de contextes différents que l'on peut correctement modéliser [Adda, 97a].

### 2.3.1. Les corpus BREF et BREF-80

À l'instar des grandes bases de données nord-américaines [Fisher, 86 ; Lamel, 86], le corpus BREF a été conçu pour fournir des données acoustiques de parole pour le développement et l'évaluation des SRAP de type machine à dictée [Gauvain, 90 ; Lamel, 91][1]. L'ensemble des textes enregistrés dans BREF ont été extraits du journal *Le Monde*. Ces textes ont été sélectionnés pour maximiser le nombre de contextes phonétiques et la taille du vocabulaire. Le corpus complet contient plus de 100 heures de parole lues sans préciser la ponctuation, soit 167 359 phrases provenant de 120 locuteurs.

---

[1] BREF à également été conçu pour fournir un large corpus de parole continue pour l'acquisition de connaissances acoustico-phonétiques en français.

Issu de BREF, le sous-corpus BREF-80 comprend les enregistrements de 80 locuteurs. Il contient 5 330 phrases, soit environ 67 phrases par locuteur. Bien que ces données ne représentent qu'une petite partie du corpus BREF, les phrases ont été sélectionnées pour couvrir la plupart des textes d'apprentissage de BREF afin de conserver le plus grand nombre de contextes phonétiques avec le minimum de données. Le sous-corpus BREF-80 a été spécialement étudié pour l'apprentissage des SRAP indépendants du locuteur et du vocabulaire.

BREF a été réalisé par le LIMSI et a bénéficié du soutien financier du GDR-PRC CHM, de l'ACCT (OFIL) et de la CEE (projet ESPRIT Polyglot).

### 2.3.2. Les corpus des Actions de Recherches Concertées "Linguistique, Informatique et Corpus Oraux"

Précédant les JST 97, deux journées ateliers (les 13 et 14 avril 1997) ont été consacrées au premier bilan des Actions de Recherches Concertées (ARCs), lancées en 1994 par l'AUPELF-UREF. L'ARC "Linguistique, Informatique et corpus ORaux" (ILOR) [Adda, 97b] s'était fixé les objectifs suivants [Dolmazon, 97] :

- la mise au point et l'utilisation d'une méthodologie commune pour l'évaluation comparative et contrastive de systèmes de reconnaissance automatique de parole sur une, ou plusieurs tâches de référence, afin d'accompagner et de faire progresser la recherche fondamentale et appliquée dans le domaine [Dolmazon, 97].
- le regroupement et la distribution des ressources nécessaires aux campagnes d'évaluation.

Les participants à la première campagne de test de systèmes de reconnaissance complets (campagne B1-DV1) de cet ARC (ARC-B1) ont donc pu disposer d'un ensemble de données pour l'apprentissage (BREF-80, BREF, textes du journal *Le Monde* de 1987 à 1988, listes de mots 64k/87-88 et 20k/87-88 des 64 000 et 20 000 mots les plus fréquents, phonétisation de la liste 64k à partir de BDLEX, modèles de langage bi-gramme et tri-gramme sur les listes 20k et 64k), le développement (corpus de textes lus par 20 locuteurs prononçant chacun des paragraphes issus du journal *Le Monde* sur la période du 13 au 25 mai 1996)[1] et l'évaluation (corpus de paroles lues par 20 locuteurs, différents des locuteurs de développement, prononçant chacun des paragraphes issus du journal *Le Monde* sur la période du 4 au 16 novembre 1996) de leur SRAP.

Le style d'élocution retenu pour l'ensemble de ces données acoustiques est la "parole lue" sans précision de la ponctuation. Les conditions d'enregistrement sont constantes

---

[1] Ce corpus de développement utilisé pour les tests du DAP (cf. *2.9. Reconnaissances et résultats*) est composé de 300 phrases lues par 8 femmes et 12 hommes.

(microphone SHURE SM10, locaux d'enregistrement et environnement de bureau semblables).

### 2.3.3. Bases de données utilisées

Les corpus de données disponibles au LIA pour la mise au point des modules acoustiques des SRAP sont un sous-ensemble de la campagne d'évaluation B1-DV1[1] : le corpus BREF-80 et le corpus de développement.

Nous avons choisi de partager ces ressources acoustiques de la façon suivante :

- Corpus d'apprentissage : BREF-80.
- Corpus de développement et de test : corpus de développement de B1-DV1.

De notre point de vue, la faiblesse principale de ces données est qu'il n'existe pas de phonétisation et d'alignement phonétique et acoustique de référence[2]. À l'instar de [Fohr, 96], une partie de notre travail consiste donc à régler ce problème.

## 2.4. Module de phonétisation

Après les corpus de parole, une autre ressource nécessaire à la réalisation d'un SRAP est une transcription phonétique des corpus de données. En effet, l'unité de reconnaissance de base des SRAP est, la plupart du temps, dérivée du phonème. Le premier travail consiste donc à phonétiser l'ensemble des corpus nécessaires (corpus d'apprentissage, corpus de test mais aussi les différents lexiques pour les phases de décodage) : c'est le rôle du module de phonétisation ou transcription graphèmes-phonèmes.

Dans le domaine de la transcription graphèmes-phonèmes, deux approches sont principalement utilisées : la première propose de résoudre le problème au moyen d'un lexique phonétique de référence (exemple du projet BDLEX) ; la seconde repose sur une base de règles de phonétisation utilisées dans un processus de transcription graphèmes-phonèmes automatique. Ces deux approches ne sont pas exclusives l'une de l'autre. En effet, un lexique de très grande taille a été, peu ou prou, constitué par des règles ; de même, dans les approches de transcription graphèmes-phonèmes automatiques, certaines règles nécessitent des exceptions qui donnent lieu à l'emploi de listes dont l'usage s'apparente à celui d'un lexique phonétique.

Pour notre part, c'est la seconde approche qui a été retenue.

---

[1] Malgré une part de financement public, le fameux corpus BREF n'est toujours pas disponible pour la communauté parole française.

[2] Notons que ce n'est pas le cas pour l'anglais où il existe des corpus pré-segmentés. Nous pensons notamment à TIMIT.

### 2.4.1. Ensemble des phonèmes

Avant de commencer la description du module de phonétisation, ce chapitre présente l'ensemble des phonèmes adoptés :

```
ii, ou, uu, ei, ai, au, oo, eu, oe, aa, an, on, in, un,
yy, ww, uy, pp, tt, kk, bb, dd, gg, ff, ss, ch, vv, zz,
jj, mm, nn, ll, rr.
```

Soit un total de 33 phonèmes.

Ce jeu de phonèmes est utilisé pour la transcription graphèmes-phonèmes et dans les différents éléments du SRAP.

L'annexe B : *Alphabet Phonétique International* présente les équivalences entre ces codes phonétiques et l'Alphabet Phonétique International (API).

### 2.4.2. Phonétisation pour la synthèse de la parole

La prononciation d'un texte implique un certain nombre de choix de la part du locuteur : d'une part, traiter les ambiguïtés intrinsèques à la transcription graphème-phonème d'une langue donnée (ambiguïtés graphiques, syntaxiques et sémantiques) ; d'autre part, générer une prononciation en accord avec le locuteur et le contexte du discours (gestion des variantes dialectales, des accents régionaux, du débit de parole, etc.). En effet, même s'il arrive qu'à une graphie ne soit associée qu'une seule forme phonétique, il y a souvent ambiguïté quant à la transcription d'une lettre ou d'un groupe de lettres. Ces ambiguïtés peuvent être levées sur plusieurs niveaux : graphique (le "g" dans "**g**arçon" et "**g**eôle"), syntaxique (la séquence "ent" dans "jugem**ent**" et "mang**ent**") ou sémantique (le mot "fils" dans la phrase "La mère tisse des **fils** de laine pour ses **fils**"). Par ailleurs, le traitement des mots non-usuels (noms propres, sigles, mots nouveaux, mots d'origine étrangère, etc.) nécessite la mise en place de traitements particuliers pour répondre aux nouvelles ambiguïtés introduites.

Les premiers systèmes de phonétisation automatique ont été développés selon une approche à base de règles [Prouts, 80]. Cette approche permet de prendre en compte les différents niveaux de connaissance impliqués dans la transcription graphèmes-phonèmes en codant les régularités de transcription phonétique des mots usuels du langage. Les inévitables exceptions à ces règles sont alors codées, soit sous forme de liste, soit selon le formalisme même des règles.

Nous disposons au LIA d'un système de conversion automatique graphèmes-phonèmes, appelé GRIPHON, utilisé dans le cadre d'une application de synthèse automatique de la parole [Béchet, 95]. Ce système s'est donné comme objectif le traitement de textes quelconques. Il utilise une base de règles contextuelles pouvant être contraintes. L'intégration des différents niveaux linguistiques se fait en amont de la phase de phonétisation par un étiquetage syntaxique des mots composant la phrase traitée. C'est en fonction des étiquettes posées que telle ou telle base de règles sera employée. Notons que GRIPHON prend en compte le traitement des mots non usuels et, d'une manière plus générale, de tous les mots inconnus [Spriet, 96 ; Béchet, 97a].

### 2.4.3. Phonétisation pour la RAP

À l'instar de GRIPHON, la plupart des systèmes de phonétisation se contentent le plus souvent de régler les ambiguïtés de transcription graphèmes-phonèmes en proposant une prononciation "standard" unique (par rapport à une référence). Cette solution s'avère satisfaisante dans les applications de synthèse automatique de la parole. Cependant, si le processus de phonétisation doit être utilisé en RAP, il devient nécessaire de générer plusieurs variantes afin d'adapter le système aux différentes prononciations possibles. Ces variantes sont bien entendues indispensables lors de la phase de décodage où, dans la plupart des cas, elles s'intègrent au lexique de l'application [Lamel, 96]. Par ailleurs, la mise en œuvre de variantes phonétiques est également nécessaire dans le cadre de l'apprentissage des modèles acoustiques ou, plutôt, pour la détermination de l'alignement optimal entre modèles et signaux acoustiques des corpus de données destinés à l'apprentissage.

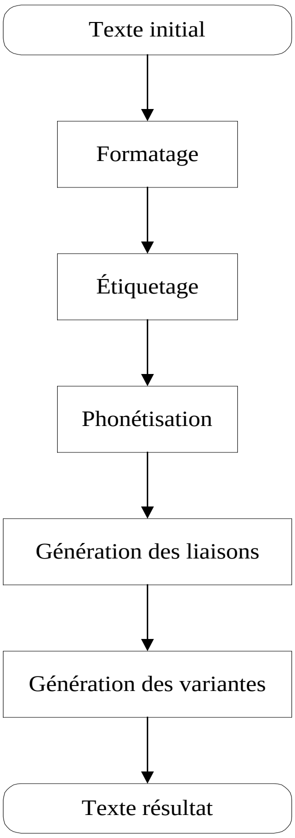


Figure 2.2 : Synoptique du système de phonétisation pour la RAP

Le module de phonétisation pour la RAP proposé est fondé sur le système de conversion automatique graphèmes-phonèmes GRIPHON. Il peut se décomposer en plusieurs étapes successives posant chacune ses propres problèmes (cf. figure 2.2).

#### 2.4.3.1. Formatage de texte

Cette étape permet de transformer un texte afin de le rendre compatible avec l'étape de phonétisation. L'objectif est de produire un texte sous forme d'une suite de mots, composée uniquement de caractères alphabétiques et de signes de ponctuation. C'est à ce niveau que l'on va traiter les problèmes liés aux nombres (passage de leur écriture en chiffres arabes ou romains à une transcription orthographique) ainsi que les différentes interprétations possibles des abréviations (si "Mlle" se traduira toujours par "mademoiselle", la lettre "m" pourra, selon le contexte, se réécrire, soit en "mètre", soit en "minute", soit encore rester telle quelle).

#### 2.4.3.2. Étiquetage syntaxique

Lors du processus de conversion graphèmes-phonèmes, il peut être nécessaire de disposer d'un étiquetage syntaxique afin de lever certaines ambiguïtés de phonétisation (comme, par exemple, pour le mot "président" qui se prononce /prezidɑ̃/ s'il est employé comme un nom, et /prezid/ s'il est employé comme verbe conjugué au présent, à la troisième personne du pluriel). Cette étape d'étiquetage consiste à affecter une catégorie syntaxique à chaque mot de la phrase à phonétiser. Pour cela, nous utilisons le module d'étiquetage syntaxique développé au LIA [El-Bèze, 95 ; Spriet, 95].

**Remarque**

La liste des étiquettes syntaxiques utilisées est donnée dans l'annexe C : *Classes syntaxiques*.

#### 2.4.3.3. Phonétisation

Suite aux traitements précédents, l'entrée de l'étape de phonétisation se présente sous la forme d'un texte "nettoyé" ne comportant ni abréviations, ni caractères non alphabétiques ; de plus, à chaque mot du texte est associée une étiquette. Le cœur du système se compose d'un ensemble de règles contextuelles de phonétisation et d'un programme qui applique ces règles pour transformer une forme graphique en une forme phonétique ; l'application des règles de phonétisation est contrainte par les étiquettes.

En ce qui concerne la phonétisation, notre étude a essentiellement porté sur le traitement des sigles et des noms propres. En effet, une simple observation des corpus journalistiques utilisés dans ce travail montre que l'univers des sigles et des noms propres, loin d'être clos, est en perpétuelle expansion.

### Traitement des sigles

Les sigles envahissent notre vocabulaire quotidien. Leur prolifération est telle qu'on ne peut plus les ignorer dans les systèmes de Traitement Automatique du Langage Écrit et Parlée (TALEO). La gestion d'un dictionnaire exhaustif de tous les sigles répertoriés avec leur prononciation attestée se révèle insuffisante.

Face à un sigle, on peut décider de le lire comme s'il s'agissait d'un mot ou de l'épeler lettre par lettre. Il ne s'agit pas d'un "ou" exclusif car quelques sigles sont mi-lus, mi-épelés (V-DAT, CDROM) et d'autres admettent les deux modes d'oralisation (ONU, CES).

Nous proposons de générer deux transcriptions phonétiques différentes en fonction du mode d'oralisation. Dans le cas des sigles lus, nous appliquons un ensemble de règles de phonétisation spécifiques. En effet, si les règles concernant les mots usuels suffisent dans la plupart des cas pour oraliser des sigles simples ("ONU", "ARC", etc.), elles sont incomplètes pour d'autres sigles ("SGEN" qui se prononce */zgɛn/*, par exemple). Il a donc fallu mettre en évidence des règles de prononciation propres à l'oralisation des sigles. Ces règles ont été établies à partir de l'examen des 1 000 sigles les plus courants extraits d'un corpus du journal *Le Monde*.

### Traitement des noms propres

L'oralisation des noms propres justifie à elle seule un traitement particulier. En effet, la phonétisation des noms propres diffère de celle des mots usuels de manière significative. Des phénomènes relevant d'archaïsmes ("**Isle**-sur-Sorgue" prononcé /il/ et non pas /isl/), d'agglutination ("**Montpellier**" prononcé /mɔ̃tpølje/), de régionalisme ("S$^t$ **Gaudens**" prononcé /godɛ̃s/ dans le sud-ouest) [Yvon, 93] perturbent l'application des règles d'oralisation classiques. Par ailleurs, la phonétisation des mots d'origine étrangère est grandement conditionnée par l'origine linguistique du mot traité [Belrhali, 95a]. Cette origine conditionne une façon particulière de prononcer ces mots dans la langue d'emprunt.

À partir d'un corpus du journal *Le Monde*, 10 000 noms et prénoms ont été classés en fonction de certains traits communs, caractéristiques de leurs prononciations. Cette classification a abouti à des ensembles disjoints de par leur consonance (francophone, slave, asiatique, germanique, etc.) mais pas forcément par leur graphie. Huit groupes linguistiques ont ainsi été définis de façon empirique[1]. Des ensembles de règles *ad hoc* modélisent les particularités[2] de ces différents groupes.

---

[1] À ce propos, il serait intéressant de questionner des linguistes sur la pertinence de ces groupes linguistiques.

[2] Cet ensemble de règles ne représente pas la phonétisation complète des langues ainsi modélisées mais plutôt la manière dont un locuteur francophone peut les prononcer.

Le traitement d'un nom propre entraîne sa phonétisation par l'ensemble des huit groupes de règles. Lorsque deux phonétisations sont identiques, nous n'en gardons qu'une seule.

#### 2.4.3.4. Génération des liaisons

La liaison est la survivance de quelques enchaînements de consonnes finales en ancien français : autrefois, toutes les consonnes finales étaient prononcées. En français moderne, un certain nombre de ces consonnes sont muettes dans les mots pris isolément. Cependant, on les prononce lorsque le lien entre un mot (à finale consonantique) et le mot suivant (à initiale vocalique) est assez fort pour que se conserve l'enchaînement ancien [Delattre, 66]. Toute la difficulté du traitement des liaisons résulte précisément dans l'appréciation de la force de ce lien.

On distingue généralement trois types de liaisons : les liaisons obligatoires, interdites et facultatives. Malgré quelques principes généraux, les règles qui régissent ces liaisons sont souvent floues et font appel à des considérations syntaxiques, parfois sémantiques.

Dans une application de synthèse automatique de la parole, on se limite généralement à la mise en place de règles extraites de la littérature [El-Bèze, 90 ; Béchade, 92 ; Grévisse, 93]. Les critères permettant de trancher pour déterminer la présence ou l'absence de liaison entre deux mots sont la graphie et la catégorie syntaxique ; non seulement des mots examinés, mais aussi des mots qui les précèdent ou les suivent. À l'inverse des applications de synthèse qui peuvent se contenter de gérer uniquement les liaisons interdites et obligatoires, en RAP, l'étude des corpus utilisés montre une extrême variabilité lors de la réalisation des liaisons [Belrhali, 95b]. Les contraintes d'application doivent donc être plus souples.

#### 2.4.3.5. Variantes et phonétisation

En RAP, la mise en place de variantes de prononciation est indispensable. Mais, l'application systèmatique des règles extraites dans la littérature (ə muet, assimilation, dénasalisation, etc.) entraîne une surgénération d'hypothèses. De plus, certaines de ces hypothèses peuvent se révéler incohérentes. C'est pourquoi, il s'avère nécessaire de contrôler le processus de génération des variantes.

Plusieurs méthodes sont possibles :
- l'utilisation d'un dictionnaire de variantes obtenues manuellement [Adda, 97a].
- une étude de corpus de parole transcrit phonétiquement par un expert [Belrhali, 95b].
- la génération d'hypothèses avec contrôle automatique [Pérennou, 96]

Pour notre part, en fonction du type de donnée à phonétiser, nous utilisons deux approches différentes. La première, dite approche hors contexte, est utilisée pour la phonétisation de phrases. Elle consiste à appliquer systématiquement un ensemble de règles de génération de variantes. Dans ce cas, le problème de la surgénération n'est pas véritablement contraignant. En effet, ce type de donnée est destiné à un processus d'alignement automatique et la prise en compte d'hypothèses supplémentaires se limite à la phrase traitée. Mais dans le cadre de la phonétisation d'un lexique, le phénomène de surgénération peut impliquer une augmentation significative de l'espace de recherche à traiter par le module de reconnaissance. Dans la seconde approche, dite approche en contexte, nous proposons de constituer le lexique de prononciation à partir d'un corpus phonétisé par un système de transcription graphèmes-phonèmes pour la synthèse (GRIPHON). Les variantes du lexique correspondent alors aux différentes phonétisations observées dans le corpus de données. Ces variantes résultent du choix du groupe de règles de phonétisation des noms propres [Béchet, 97b], les différentes liaisons générées en fonction du contexte d'occurrence des mots du lexique et les règles de cohérences sur la phrase concernant la chute du ə muet. Par ailleurs, les mots du lexique doivent appartenir au corpus phonétisé.

Les règles de génération de variantes mises en place dans notre module de phonétisation pour la RAP s'appliquent :

- au ə muet ou *schwa* qui est soit éludé, soit réalisé sous la forme de /ø/.
- aux phénomènes de liaisons où les consonnes finales et génératrices sont soit éludées, soit réalisées.
- aux sigles et aux noms propres qui sont phonétisés en utilisant toutes leurs étiquettes possibles.

Le tableau 2.2 présente un exemple de phrase dans le cadre de la phonétisation hors contexte d'une phrase. Nous renvoyons à la troisième partie de ce document pour des exemples de phonétisation de lexique.

**Phrase originale (IOME0185) :**

Il s'en est entretenu avec le secrétaire d'Etat, et a laissé entendre que le président Bush avait accueilli favorablement son initiative.

**Phrase avec étiquettes syntaxiques et variantes phonétiques :**

| | | |
|---|---|---|
| il | PPER3MS | iill |
| s' | PREFMS | ss |
| en | PPOBJMS | an |
| | PPOBJMS | annn |
| | PPOBJMS | aann |
| est | NMS | aisstt |
| | VE3S | ai |
| | VE3S | aitt |
| entretenu | VPPMS | anttrreutteunnuu |
| | VPPMS | anttrreuttnnuu |
| | VPPMS | anttrrtteunnuu |
| | VPPMS | anttrrttnnuu |
| avec | PREP | aavvaikk |
| le | DETMS | lleu |
| | DETMS | ll |
| secrétaire | NMS | sseukkrreittairr |
| | NMS | sskkrreittairr |
| d' | PREPADE | dd |
| Etat | NMS | aittaa |
| | NMS | aittaatt |
| , | YPFAI | |
| et | COCO | ei |
| | COCO | eitt |
| a | VA3S | aa |
| laissé | VPPMS | llaissei |
| entendre | VINF | anttanddrreu |
| | VINF | anttanddrr |
| que | COSUB | kkeu |
| | COSUB | kk |
| le | DETMS | lleu |
| | DETMS | ll |
| président | NMS | pprreizziiddan |
| | NMS | pprreizziiddantt |
| | V3P | pprreizziidd |
| | V3P | pprreizziiddtt |
| Bush | XFAMIL->Anglais | bbouch |
| | XFAMIL->Français | bbuuch |
| avait | VA3S | aavvai |
| | VA3S | aavvaitt |
| accueilli | VPPMS | aakkoeyyii |
| favorablement | ADV | ffaavvoorraabblleumman |
| | ADV | ffaavvoorraabblleummantt |
| | ADV | ffaavvoorraabbllmman |
| | ADV | ffaavvoorraabbllmmantt |
| son | DETFS | sson |
| | DETFS | ssonnn |
| | DETFS | ssoonn |
| initiative | NFS | iinniissyyaattiivv |
| . | YPFOR | |

Tableau 2.2 : Exemple de phonétisation avec variantes phonétiques

Cet exemple appel quelques commentaires :

- la génération de variante liée au ə muet produit des incohérences, comme pour le mot "entretenu" où nous trouvons des suites de plus de trois consonnes.
- le mot "président" est étiqueté NMS (nom masculin singulier) et V3P (verbe 3$^{\text{ème}}$ personne du pluriel), ce qui permet de produire les variantes de prononciation associées.
- le mot "Bush" reçoit l'étiquette `XFAMIL` qui correspond à un nom propre de type nom de famille. Puis, le module de traitement des noms propres applique les différents groupes de règles. Notons que sur cette exemple, uniquement deux phonétisations différentes ont été produites.

## 2.4.4. Transcription phonèmes-graphèmes

Les travaux menés dans le cadre de la transcription graphèmes-phonèmes nous ont poussé à nous intéresser aux processus inverse. Inspiré du module de phonétisation utilisé pour la RAP, nous présentons dans ce chapitre un système de transcription automatique phonèmes-graphèmes [Béchet, 97c].

Le processus de transcription phonèmes-graphèmes peut s'intégrer dans différents types d'application de TALEO. Notamment, il pourrait être utilisé par des SRAP, pour le traitement des Mots Hors Vocabulaire (MHV) ou dans le cadre du Traitement Automatique du Langage Naturel (TALN) pour la correction orthographique en traitant le cas particulier des erreurs phonographiques [Véronis, 87 ; Laporte, 89 ; Ghneim, 95]. L'approche développée lors de cette étude ressemble à celle du module de phonétisation pour la RAP présentée précédemment : à partir d'une chaîne phonétique pré-segmentée en unités lexicales, un moteur à base de règles de transcription phonème-graphème propose une liste de graphies candidates. À l'issue de ce traitement, un module de tri fondé sur un critère de vraisemblance statistique va permettre de classer les graphies les unes par rapport aux autres. Notons que si la transcription orthographique est le processus inverse de la phonétisation, la relation qui les lie n'est malheureusement pas inversible. En effet, les contraintes graphiques posées dans les règles de phonétisation n'impliquent pas directement des contraintes phonétiques. Par ailleurs, s'il est souvent possible d'attribuer à une chaîne graphique une seule prononciation "standard", il est en revanche difficile, voire impossible, d'affecter une seule graphie à une chaîne phonétique donnée. Les phénomènes d'homophonie sont tels que l'on rencontre de nombreux cas d'homophones hétérographes en français [El-Bèze, 95], que ce soit sur le plan syntaxique ou morphologique.

### 2.4.5. Phonétisation de BREF-80

Concrètement, la phonétisation des phrases du corpus BREF-80 se traduit par la production, dans un processus intégralement automatique, de deux fichiers pour chaque phrase.

Ces fichiers sont :
- Le fichier des mots de la phrase,
- Le fichier des phonétiques (avec variantes) correspondantes.

Ces fichiers qui associent les formes orthographiques des mots aux différentes transcriptions phonétiques sont utilisés pour l'alignement des données acoustiques et phonétiques.

## 2.5. Paramétrisation des données acoustiques

Dans la plupart des applications de TAP, les propriétés du signal (redondance, plage de fréquence, etc.) rendent indispensables une paramétrisation préalable. À partir d'une représentation du signal dans le domaine temporel (suite à une étape classique d'échantillonnage), une représentation plus compacte du signal doit être utilisée[1].

Nous avons choisi, *a priori*, d'utiliser 12 paramètres MFCC (Mel Frequency Cepstral Coefficients). Ces paramètres sont des coefficients cepstraux obtenus à partir des énergies d'un banc de filtres en échelle de fréquence Mel [Davis, 80]. De plus, afin de prendre en compte la dynamique du signal, nous ajoutons aux paramètres MFCC l'énergie et les coefficients différentiels (ou coefficients delta) du premier et du second ordre. Un vecteur acoustique est donc représenté par 39 paramètres (12 MFCC + E + 13 $\Delta$ + 13 $\Delta\Delta$).

Du point de vue de l'étude bibliographique, à l'heure actuelle, le choix de ce type de paramétrisation semble être satisfaisant pour représenter le signal de parole dans le cadre de la RAP [Gauvain, 94 ; Jouvet, 94 ; Woodland, 95].

## 2.6. Alignement des données acoustiques et phonétiques

L'objectif de l'alignement des données acoustiques et phonétiques est de déterminer la suite de phonèmes optimum parmi les différentes variantes proposées par le module de phonétisation et de mettre en correspondance ces phonèmes avec les vecteurs acoustiques représentant le signal. Le module d'alignement présenté est destiné à l'apprentissage de modèles acoustiques phonétiques d'un SRAP, dont la phase d'initialisation nécessite une association entre les phonèmes et les vecteurs acoustiques du corpus d'apprentissage.

---

[1] Pour une étude comparative des différents types de paramétrisation utilisées en RAP, nous proposons de consulter [Barras, 96].

Comme nous l'avons vu précédemment, le problème de la transcription graphèmes-phonèmes dans les SRAP nécessite la mise en place de variantes de prononciations. En effet, si pour une application de synthèse de la parole, il suffit de proposer UNE prononciation satisfaisante, il faut aligner LA prononciation que le locuteur a utilisée dans le cadre de l'alignement de données acoustiques et phonétiques. Il est donc necessaire de générer toutes les prononciations possibles d'une phrase et de déterminer la prononciation optimale.

### 2.6.1. Principe de l'alignement automatique

L'alignement des données acoustiques et phonétiques est une étape qui peut s'avérer extrêmement coûteuse, voire même impossible à réaliser pour de grands corpus, si la procédure est manuelle. Il est donc indispensable de disposer d'une procédure d'alignement automatique.

La procédure d'alignement automatique est fondée sur une implémentation particulière de l'algorithme de Viterbi. L'alignement consiste à "contraindre" le processus de décodage sur la suite des labels représentant la phrase ; l'algorithme de Viterbi déterminant alors l'alignement optimal des paramètres acoustiques sur ces labels.

Le problème est le suivant : pour pouvoir réaliser cet alignement, il faut disposer d'un ensemble de modèles de Markov initiaux. Afin de calculer ces HMM initiaux, plusieurs options sont envisageables. Par exemple, un calcul global des paramètres (moyennes et variances) sur un sous-ensemble du corpus d'apprentissage ; l'ensemble des HMM initiaux contient alors les mêmes paramètres [Young, 96]. Une autre option est de réaliser un étiquetage manuel sur un petit ensemble de données ; les paramètres appris sur cet ensemble serviront à l'alignement sur la totalité du corpus d'apprentissage. Cette technique est appelée *boot-strap*. Enfin, il est également possible d'utiliser des HMM provenant d'un autre corpus.

### 2.6.2. Procédure d'alignement automatique

Les ressources (corpus et HMM initiaux) utilisées pour l'alignement automatique de BREF-80 ont été construites à l'occasion de divers travaux [Linares, 96 ; Igounet, 96]. C'est à partir des HMM calculés sur un corpus d'un millier de phrases mono-locuteur, enregistrées au LIA, que nous avons réalisé un premier alignement. Par la suite, c'est une succession de deux phases d'alignement et d'apprentissage qui permet la construction des modèles acoustiques utilisés pour le DAP.

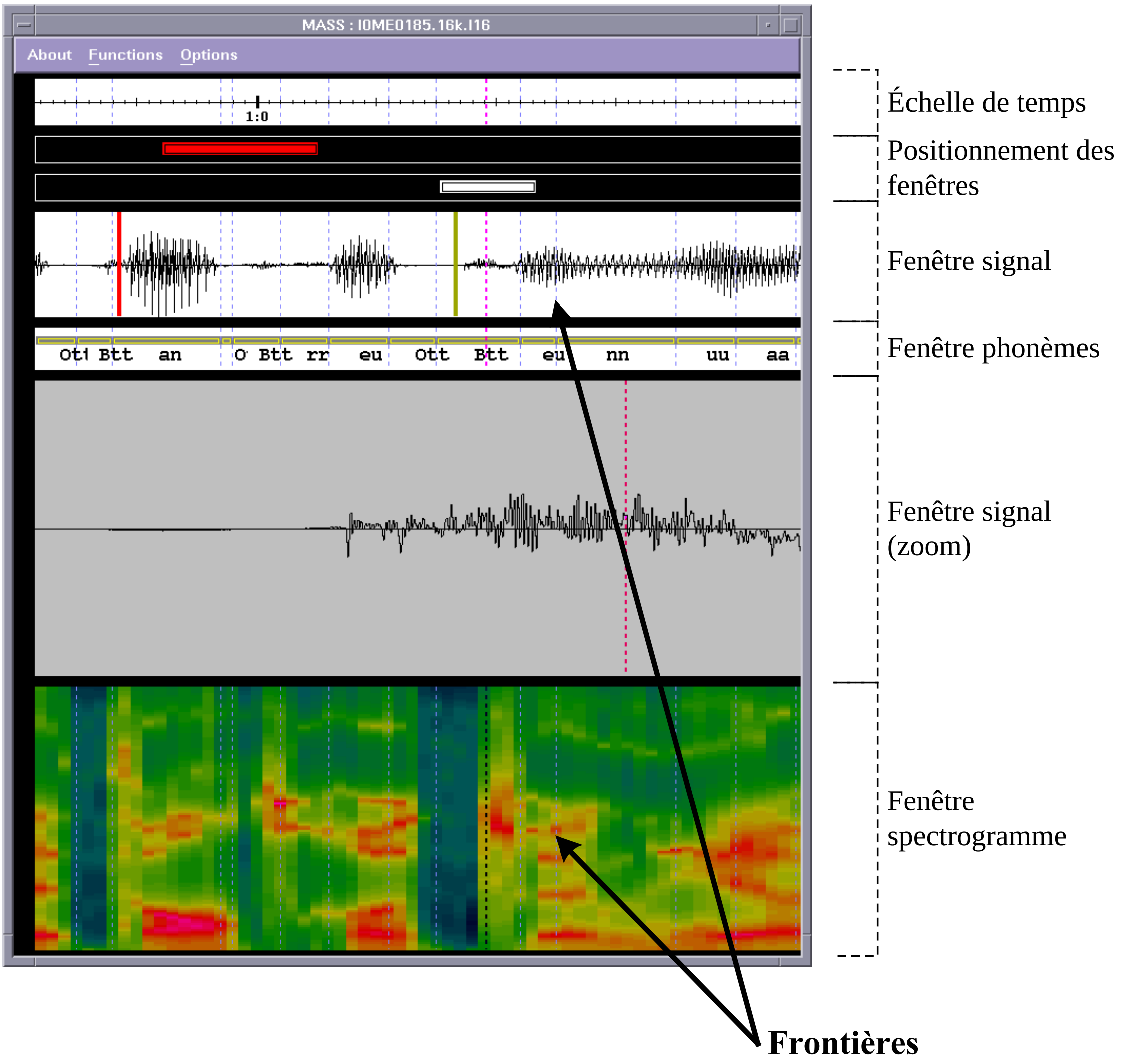

MASS : I0ME0185.16k.I16
About
Functions
Options
1:0
Ot! Btt an O Btt rr eu Ott Btt eu nn uu aa
Échelle de temps
Positionnement des fenêtres
Fenêtre signal
Fenêtre phonèmes
Fenêtre signal (zoom)
Fenêtre spectrogramme
Frontières

Figure 2.3 : Exemple d'alignement pour le mot *entretenu* (phrase I0ME0185)

La figure 2.3 présente un exemple d'alignement automatique sur la phrase I0ME0185 en fonction des variantes phonétiques proposées par le phonétiseur. Sur la figure, nous trouvons la suite des phonèmes après alignement dans la fenêtre phonèmes ; les frontières apparaissent sur les fenêtres signal, phonèmes et spectrogramme.

Après phonétisation, cette procédure d'alignement automatique est utilisée pour les corpus d'apprentissage et de test.

## 2.7. Modèles de Markov

À chaque unité acoustico-phonétique du corpus d'apprentissage est associé un prototype de modèle de Markov. Ce prototype contient la topologie choisie pour cette unité, c'est-à-dire le nombre d'états du modèle, les probabilités initiales de transitions entre états et les paramètres de la loi de probabilité associée à chaque état.

### 2.7.1. Topologies

Toutes les unités acoustico-phonétiques sont modélisées par une topologie de type Bakis avec des densités de probabilités gaussiennes sur les états. À partir des phonèmes définis dans le chapitre *2.4.1. Ensemble des phonèmes*, deux topologies ont été retenues :

- trois états sans saut (cf. figure 2.4) pour les modèles `ii, ou, uu, ei, ai, au, oo, eu, oe, aa, an, on, in, un, yy, ww, uy, Bpp, Btt, Bkk, bb, dd, gg, ff, ss, ch, vv, zz, jj, mm, nn, ll, rr`. La topologie trois états de type Bakis est un choix classique en RAP. Elle permet d'interpréter l'état initial et l'état final comme des états transitoires qui encadrent un état central plus stationnaire. Nous avons choisi de modéliser les phonèmes `pp`, `tt` et `kk` en deux parties : une partie représentant l'occlusion (noté `O` pour Occlusion) et une partie représentant l'explosion (noté `B` pour Burst) [Marcadet, 91].

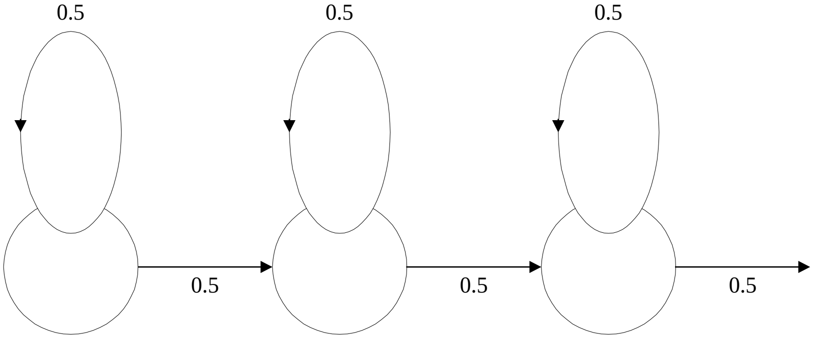


Figure 2.4 : Topologie trois états sans saut de type Bakis

- deux états avec saut (cf. figure 2.5) pour les modèles `Opp, Ott, Okk`. Cette topologie modélise la partie occlusive des phonèmes `pp, tt` et `kk`. La transition directe entre l'état initial et l'état final permet de modéliser une réalisation acoustique très courte de l'occlusion.

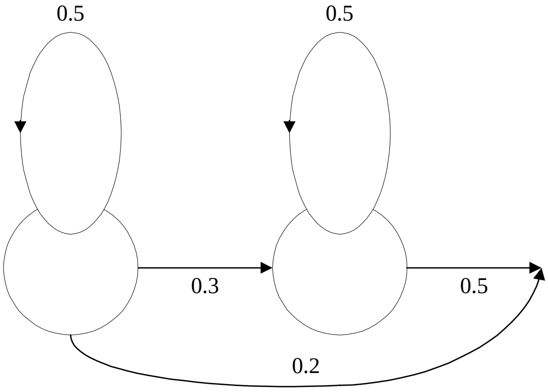


Figure 2.5 : Topologie deux états avec saut de type Bakis

### 2.7.2. Perfectionnements

En général, les modèles de Markov utilisés dans la plupart des SRAP intègrent d'autres perfectionnements. En effet, diverses techniques visant à augmenter le nombre de paramètres des modèles permettent une meilleure modélisation du signal de parole [Jouvet, 95]. Pour des modèles de Markov à densités de probabilités gaussiennes, cette augmentation du nombre de paramètres peut s'effectuer dans trois directions.

Tout d'abord, par l'augmentation du nombre de modèles représentant chaque unité : c'est la technique dite "des multi-modèles" [Jouvet, 94 ]. Pour une unité donnée, le corpus d'apprentissage est classifié en sous-ensembles destinés à chaque type de modèle. Lors du décodage, le réseau représentant l'ensemble du vocabulaire est répété autant de fois que nécessaire. Le multi-modèle comporte donc plusieurs branches, chacune disposant de son propre jeu de paramètres. Dans la littérature, l'exemple de multi-modèles le plus couramment utilisé est fondé sur une classification *a priori* des locuteurs entre hommes et femmes.

Une deuxième approche, qui vise également à augmenter le nombre de modèles, consiste à étendre la portée des modèles de Markov sur les unités adjacentes gauche et/ou droite. Cela permet de prendre en compte les influences contextuelles [Schwartz, 95]. Quand l'unité de base est le phonème, on parle alors de phonèmes en contexte. Mais une telle approche conduit à un grand nombre d'unités. Dans la pratique, il est impossible de les apprendre dans leur totalités, et l'estimation des paramètres de certaines de ces unités n'est pas fiable. Par conséquent, il faut réduire ce nombre d'unités : à partir de connaissances phonétiques, du choix du modèle à employer par arbres de décision [Bahl, 89], de regroupements automatiques des modèles [Lee, 90b], ou encore de regroupements des densités de probabilités [Hwang, 93b] ou des états [Young, 94].

Enfin, l'utilisation de densités de probabilités d'émission multi-gaussiennes permet d'éviter certaines hypothèses grossières sur la forme de la densité, pourvu que le nombre de gaussiennes soit suffisamment élevé [Jouvet, 95]. La procédure de mise en œuvre des multi-gaussiennes peut notamment fonctionner de manière incrémentale par duplication des gaussiennes, perturbation et apprentissage.

Dans le cadre de la réalisation du système de DAP, nous nous sommes limités à l'emploi de multi-gaussiennes.

## 2.8. Apprentissage

L'apprentissage des paramètres acoustiques optimaux des modèles de Markov est une étape essentielle dans la construction d'un SRAP. C'est la qualité de cette modélisation qui conditionne en grande partie les résultats du DAP.

De notre point de vue, plutôt que l'algorithme d'apprentissage utilisé, ce sont les données d'apprentissage qui sont, en premier lieu, primordiales pour obtenir de bonnes performances dans un SRAP. Définissons l'ensemble des ressources d'apprentissage idéales ; c'est-à-dire un corpus acoustique "infini" contenant suffisamment de représentants de chaque modèle de Markov et un alignement acoustico-phonétique conforme. Mais la réalisation d'un tel ensemble est irréaliste. C'est pourquoi certaines approximations sont couramment utilisées comme, par exemple, le partage des états pour remédier au déficit de représentation des modèles dans le corpus [Young, 94b] ou des techniques de phonétisation et d'alignement automatiques.

Rappelons les ressources nécessaires pour la phase d'apprentissage :

- Ensemble logiciels (cf. *2.2. Outils de développements*).
- Corpus de données acoustiques (cf. *2.3. Bases de données*).
- Représentations phonétiques des données (cf. *2.4. Module de Phonétisation*).
- Paramètres acoustiques et phonétiques (cf. *2.5. Paramétrisation des données acoustiques* et *2.6. Alignement des données acoustiques et phonétiques*).
- Chaînes de Markov (cf. *2.7. Modèles de Markov*).

Dans ce chapitre, nous décrivons le processus d'apprentissage mis en œuvre, à partir de ces ressources, dans le cadre de la réalisation d'un module de DAP.

### 2.8.1. Initialisation de l'apprentissage

La phase d'initialisation permet une estimation rudimentaire des paramètres pour chaque modèle de Markov modélisant une unité acoustique. Cette phase utilise une segmentation phonétique du corpus d'apprentissage. Sa particularité est de traiter chaque modèle de Markov indépendamment par regroupement de l'ensemble des segments de parole associés à chaque modèle.

Nous proposons d'initialiser les modèles de Markov de deux façons différentes.

#### 2.8.1.1. Initialisation à partir de l'alignement acoustique et phonétique

Pour cette phase d'initialisation des modèles de Markov, nous utilisons la transcription phonétique de la base de données et un alignement acoustique correspondant.

Elle est composée de deux étapes :

- La première étape consiste, pour chaque HMM, à l'initialisation des paramètres de ses états au moyen d'une procédure fondée sur le *K-means* et sur l'algorithme de Viterbi (cf. figure 2.6).

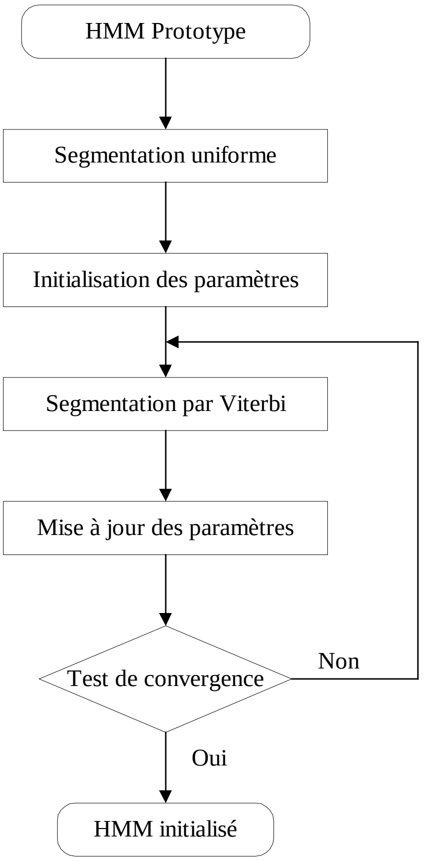

HMM Prototype
Segmentation uniforme
Initialisation des paramètres
Segmentation par Viterbi
Mise à jour des paramètres
Test de convergence
Non
Oui
HMM initialisé

Figure 2.6 : Première étape de la phase d'initialisation

Chaque portion de signal étiquetée, à partir de l'étape d'alignement automatique définie précédemment, est tout d'abord découpée en un nombre de segments égal au nombre d'états du HMM associé (cf. figure 2.6 - Segmentation uniforme). Les paramètres des états du modèle (variance et moyenne) peuvent alors être initialisés sur l'ensemble des observations acoustiques correspondantes (cf. figure 2.6 - Initialisation des paramètres). Dans le cas particulier où des mixtures de gaussiennes sont utilisées, une version modifiée de l'algorithme du *k-means* permet de classer les vecteurs acoustiques par gaussiennes. Puis, une procédure itérative fondée sur l'algorithme de Viterbi est utilisée pour produire une segmentation optimale des données acoustiques sur les états du HMM (cf. figure 2.6 - Segmentation par Viterbi), ce qui permet de mettre à jour les paramètres du HMM (cf. figure 2.6 - Mise à jour des paramètres). Ce processus itératif s'arrête quand les probabilités ne s'améliorent plus significativement ou quand un nombre d'itérations, fixé *a priori*, est atteint (cf. figure 2.6 - Test de convergence). Quant aux probabilités de transition, elles sont simplement estimées par le compte du nombre de fois où chaque état du modèle est utilisé pendant le processus d'alignement.

- Le principe de fonctionnement de la seconde étape est très proche de la première (cf. figure 2.7).

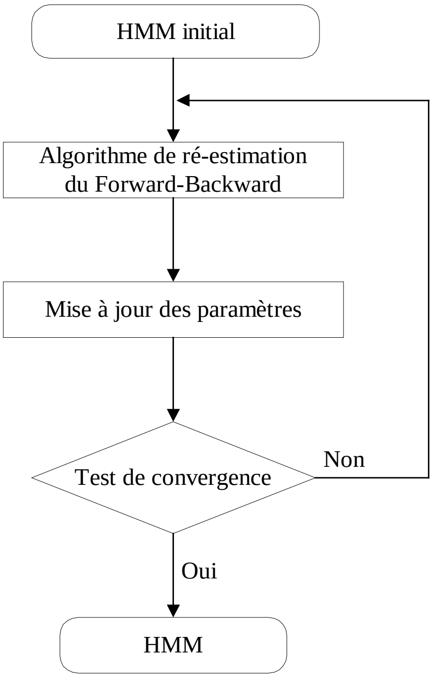

HMM initial
Algorithme de ré-estimation du Forward-Backward
Mise à jour des paramètres
Test de convergence
Non
Oui
HMM

Figure 2.7 : Seconde étape de la phase d'initialisation

Trois différences sont à relever : une utilisation de l'algorithme du *Forward-Backward* à la place de celui de Viterbi, pas de segmentation selon une répartition proportionnelle au nombre d'états et une initialisation préalable des HMM. Cette étape permet une ré-estimation des paramètres des modèles de Markov dans un processus itératif.

### 2.8.1.2. Initialisation globale moyenne

L'une des difficultés essentielles d'une initialisation fondée sur un alignement des données acoustiques et phonétiques est le degré de validité de cet alignement. C'est pourquoi la qualité de l'alignement est, en général, contrôlée et corrigée dans des processus semi-automatiques [Lamel, 93 ; Fohr, 96]. Nous proposons de mesurer l'impact de l'initialisation des modèles probabilistes en comparant les performances obtenues entre une procédure fondée sur un alignement automatique et une initialisation globale moyenne.

L'initialisation globale moyenne consiste à calculer globalement les moyennes et variances des modèles de Markov sur l'ensemble des données d'apprentissage. De cette façon, les paramètres (moyennes et variances) de tous les modèles de Markov sont initialisés à l'identique.

**Remarque**

> Si dans le cas d'une initialisation globale moyenne, un alignement des données acoustiques et phonétiques n'est plus utile, la suite des phonèmes prononcée est, quant à elle, toujours nécessaire pour la seconde phase (ou phase de ré-estimation) de l'apprentissage des modèles de Markov. La phase d'alignement des données acoustiques et phonétiques reste donc indispensable afin de déterminer la suite de phonèmes optimum en fonction des variantes phonétiques proposées par le module de transcription graphèmes-phonèmes.

## 2.8.2. Phases de ré-estimation

L'essentiel de l'apprentissage est constitué par une succession de phases de ré-estimation avec augmentation du nombre de gaussiennes sur les états des modèles de Markov. Le choix du nombre de gaussiennes est déterminé empiriquement par les résultats de décodage.

### 2.8.2.1. Principe de la ré-estimation

À partir de la concaténation des transcriptions phonétiques de la phrase, un modèle de Markov composite est généré. Puis, l'algorithme du *Forward-Backward* est appliqué sur chacune des phrases (HMM composite et vecteurs acoustiques) du corpus d'apprentissage. Après passage de l'algorithme sur l'ensemble du corpus, les nouveaux paramètres des HMM peuvent alors être estimés.

Dans l'apprentissage des HMM utilisés dans le DAP, nous avons choisi de faire dix itérations de l'algorithme du *Forward-Backward* par phase de ré-estimation.

**Remarque**

> L'algorithme du *Forward-Backward* est utilisé dans la seconde étape de l'initialisation des HMM et dans la phase de ré-estimation. La principale différence dans ces deux processus provient de la prise en compte ou non des informations de segmentation. En fait, dans la mesure où les informations de segmentation ne sont pas prises en compte dans la phase de ré-estimation, cette phase permet de remettre en cause les frontières issues de l'alignement, qu'il soit automatique ou manuel. C'est pourquoi, cette phase de ré-estimation est particulièrement bien adaptée à l'apprentissage de SRAPC où les frontières entre unités acoustiques sont particulièrement difficiles à définir.

#### 2.8.2.2. Ajout de gaussiennes

L'augmentation du nombre de gaussiennes permet une amélioration de la modélisation des phénomènes acoustiques.

Pour des raisons de stabilité des procédures, il est préférable d'apprendre un modèle avec peu de paramètres avant de raffiner la modélisation. C'est pourquoi, l'apprentissage avec multi-gaussiennes repose sur une augmentation progressive du nombre de gaussiennes. Son principe est le suivant : tout d'abord, des modèles à une gaussienne par état sont appris ; puis, le nombre de gaussiennes par état est augmenté progressivement, par clonage et perturbation[1] ; enfin, les modèles sont ré-estimés.

## 2.9. Reconnaissances et résultats

Le module de reconnaissance utilise une implémentation particulière de l'algorithme de Viterbi : le Modèle de Propagation de Jeton (cf. *1.5.2. Le système CUED-HTK et le Modèle de Propagation de Jeton*).

Dans le cadre du DAP, le réseau de décodage est un modèle composite en boucle sur l'ensemble des phonèmes avec un modèle de langage bi-gramme phonétique. Les modèles de Markov sont constitués de topologie de type Bakis et utilisent des densités de probabilité continues multi-gaussiennes.

### 2.9.1. Protocole expérimental

Ce chapitre présente les expériences de reconnaissance réalisées et définit les critères utilisés pour l'évaluation.

#### 2.9.1.1. Expériences de reconnaissance

Les expériences de reconnaissance ont consisté à faire varier les paramètres suivants :

- le type d'initialisation de la phase d'apprentissage.
- le nombre de gaussiennes avec 1, 2, 4, 8, 12, 16, 20 et 24 gaussiennes.
- le poids du modèle de langage avec les facteurs 1, 2, 4 et 6.
- le nombre d'itérations de la phase de ré-estimation.

Nous rappelons que le corpus de test utilisé est issu des données de développement des ARC pour la campagne B1-DV1 et comprend un total de 300 phrases prononcées par 20 locuteurs femmes et hommes. La phonétisation de ce corpus de test est obtenue de la même façon que pour le corpus d'apprentissage. Il est constitué de 30 259 phonèmes. Quant au corpus d'apprentissage, il est issu de BREF-80 et comprend 5 330 phrases prononcées par 80 locuteurs femmes et hommes. Après phonétisation et alignement, ce corpus d'apprentissage est constitué de 353 208 phonèmes.

---

[1] Sur un état donné, la gaussienne de poids maximum est divisée en deux avec une légère perturbation de la moyenne.

#### 2.9.1.2. Critères de l'évaluation

Les résultats de la reconnaissance sont comparés aux étiquettes de référence par une procédure d'alignement dynamique standard [Young, 96]. Le silence n'est pas pris en compte dans l'évaluation.

Habituellement, les performances des SRAP sont évaluées par le pourcentage d'identification (*percent correct* en anglais) et le pourcentage de reconnaissance (*percent accuracy* en anglais).

Le pourcentage d'identification correspond à l'équation suivante :

$$\%\ \text{Ident.} = \frac{N - O - S}{N} \times 100 \qquad \text{(Eq. 2.1)}$$

Le pourcentage de reconnaissance correspond à l'équation suivante :

$$\%\ \text{Reco.} = \frac{N - O - S - I}{N} \times 100 \qquad \text{(Eq. 2.2)}$$

Avec :

N : le nombre total de labels.
O : le nombre d'omissions.
S : le nombre de substitutions.
I : le nombre d'insertions.

**Remarque**

> Comme le montre l'équation 2.1, le pourcentage d'identification ne prend pas en compte le nombre d'insertions. C'est le pourcentage de reconnaissance qui prend en compte l'ensemble des variables (cf. Eq. 2.2). Nous considérons le pourcentage de reconnaissance comme le paramètre le plus indicatif pour évaluer les performances d'un SRAP.

### 2.9.2. Modèle de langage bi-gramme phonétique pour le DAP

Le modèle de langage utilisé est un bi-gramme phonétique de type *back-off* avec discount égal à 0.5. Il est calculé sur le corpus d'apprentissage.

Notons que le choix du corpus d'apprentissage acoustique pour l'estimation du modèle bi-gramme phonétique est discutable. En effet, nous rappelons que dans ce corpus, les phrases ont été sélectionnées pour donner le plus grand nombre de contextes phonétiques avec le minimum de données. Le bi-gramme phonétique calculé comporte donc un biais ; il ne reflète pas véritablement l'usage.

Pour information, le tableau 2.3 présente quelques statistiques sur le nombre de représentants et les durées des unités phonétiques après les processus de phonétisation et d'alignement automatique sur le corpus d'apprentissage.

| Étiquettes | Nombre | Durée moyenne (ms) | Durée minimum (ms) | Durée maximum (ms) |
|---|---|---|---|---|
| aa | 26102 | 69,5 | 30 | 490 |
| ai | 14658 | 75,6 | 30 | 480 |
| an | 12124 | 107,7 | 30 | 400 |
| au | 4753 | 91,2 | 30 | 400 |
| bb | 4374 | 84,1 | 30 | 460 |
| ch | 2413 | 148 | 30 | 350 |
| dd | 17456 | 68,8 | 30 | 690 |
| ei | 21114 | 68,7 | 30 | 400 |
| eu | 14738 | 51,2 | 30 | 590 |
| ff | 5452 | 118,9 | 30 | 540 |
| gg | 2926 | 87 | 30 | 420 |
| ii | 20301 | 73,5 | 30 | 770 |
| in | 4190 | 101,5 | 30 | 500 |
| jj | 4278 | 114,3 | 30 | 300 |
| kk | 1866 | 105,9 | 40 | 490 |
| ll | 21666 | 58,4 | 30 | 580 |
| mm | 11085 | 91,6 | 30 | 450 |
| nn | 11658 | 86,4 | 30 | 360 |
| oe | 2540 | 93 | 30 | 370 |
| on | 6991 | 98,1 | 30 | 600 |
| oo | 9495 | 65,3 | 30 | 420 |
| ou | 5649 | 77,9 | 30 | 430 |
| pause | 17625 | 253,3 | 30 | 2010 |
| pp | 12259 | 98,3 | 40 | 450 |
| rr | 28715 | 63,3 | 30 | 750 |
| ss | 21416 | 113,3 | 30 | 490 |
| tt | 18811 | 95,5 | 40 | 590 |
| un | 1431 | 97,9 | 30 | 270 |
| uu | 8003 | 72,1 | 30 | 450 |
| uy | 1425 | 79,6 | 30 | 240 |
| vv | 6714 | 76,8 | 30 | 390 |
| ww | 2642 | 84,5 | 30 | 360 |
| yy | 8314 | 72,7 | 30 | 480 |
| zz | 5649 | 94,9 | 30 | 300 |

Tableau 2.3 : Statistiques sur le nombre de représentants
et les durées des unités phonétiques

### 2.9.3. Résultats du DAP

Ce chapitre expose les résultats de reconnaissance phonétique sur le corpus de test.

Pour des raisons de formulation, les résultats sont exposés en fonction du type d'initialisation de l'apprentissage dans les chapitres concernant le nombre de gaussiennes, le poids du modèle de langage et le nombre d'itérations de la phase de ré-estimation. L'interprétation de ces résultats est présentée dans le chapitre *2.9.3.4. Résultats en fonction du type d'initialisation.*

**Remarques**

- Aucun facteur de pénalité n'est utilisé sur les expériences de reconnaissance qui suivent.
- L'ensemble des résultats de reconnaissance du DAP est donné dans l'annexe D : *Résultats de reconnaissance du DAP*.

#### 2.9.3.1. Résultats en fonction du nombre de gaussiennes

Le tableau 2.4 et la figure 2.8 présentent les résultats de reconnaissance phonétique sur le corpus de test en fonction du type d'initialisation de l'apprentissage et du nombre de gaussiennes (sans modèle de langage) :

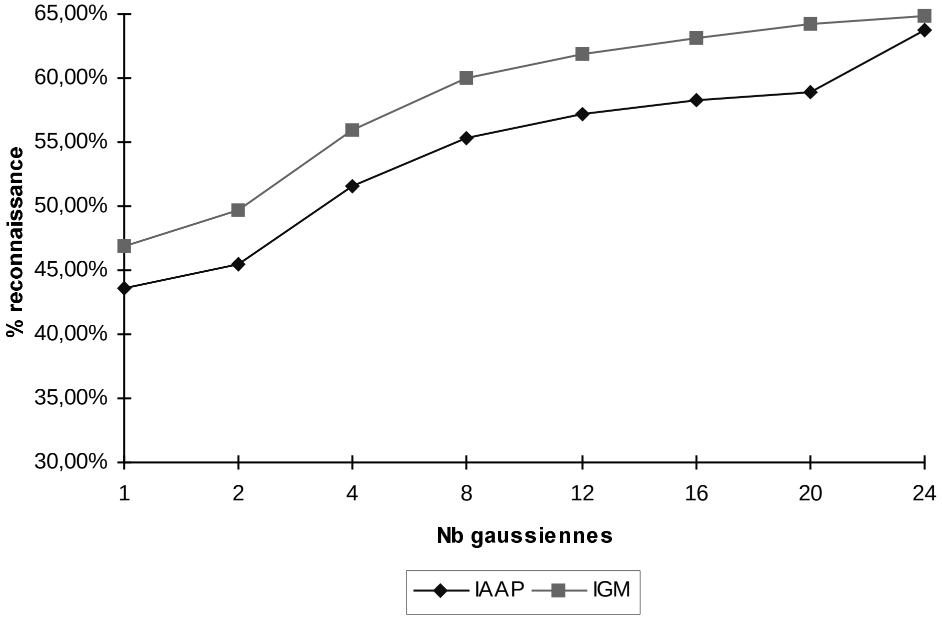


Figure 2.8 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre de gaussiennes

| Initialisation à partir de l'Alignement Acoustique et Phonétique (IAAP) | | | | | | |
|---|---|---|---|---|---|---|
| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
| 1 | 0 | 3 032 | 11 009 | 3 022 | 53,60 % | 43,61 % |
| 2 | 0 | 2 477 | 10 475 | 3 538 | 57,20 % | 45,50 % |
| 4 | 0 | 2 079 | 9 282 | 3 318 | 62,45 % | 51,49 % |
| 8 | 0 | 1 989 | 8 567 | 2 988 | 65,11 % | 55,24 % |
| 12 | 0 | 1 855 | 8 229 | 2 892 | 66,67 % | 57,12 % |
| 16 | 0 | 1 800 | 7 943 | 2 888 | 67,80 % | 58,26 % |
| 20 | 0 | 1 797 | 7 757 | 2 890 | 68,43 % | 58,88 % |
| 24 | 0 | 1 639 | 6 749 | 2 583 | 72,28 % | 63,74 % |

| Initialisation Globale Moyenne (IGM) | | | | | | |
|---|---|---|---|---|---|---|
| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
| 1 | 0 | 2 873 | 10 221 | 2 974 | 56,73 % | 46,90 % |
| 2 | 0 | 2 223 | 9 477 | 3 508 | 61,33 % | 49,74 % |
| 4 | 0 | 1 962 | 8 251 | 3 142 | 66,25 % | 55,86 % |
| 8 | 0 | 1 810 | 7 530 | 2 749 | 69,13 % | 60,05 % |
| 12 | 0 | 1 734 | 7 116 | 2 689 | 70,75 % | 61,87 % |
| 16 | 0 | 1 703 | 6 822 | 2 629 | 71,83 % | 63,14 % |
| 20 | 0 | 1 650 | 6 600 | 2 562 | 72,74 % | 64,27 % |
| 24 | 0 | 1 643 | 6 488 | 2 491 | 73,13 % | 64,90 % |

Tableau 2. 4 : Résultats de reconnaissance en fonction
du type d'initialisation de l'apprentissage et du nombre de gaussiennes

Comme nous pouvions l'envisager (cf. *2.7.2. Perfectionnements*), les résultats obtenus montrent que l'accroissement du nombre de gaussiennes permet une meilleur modélisation de l'espace acoustique. Mais, dans la mesure où les temps de calcul d'un système de DAP dépendent du nombre de paramètres (et donc du nombre de gaussiennes) des HMM utilisés, il est nécessaire de trouver un compromis entre le taux de reconnaissance et le nombre de paramètres.

Dans le cadre de ce système de DAP, nous choisissons d'utiliser des HMM avec 24 gaussiennes par états.

#### 2.9.3.2. Résultats en fonction des poids du modèle de langage

Dans la plupart des cas, l'ajout d'un modèle de langage dans un SRAP est indispensable pour obtenir de bonnes performances en reconnaissance. À ce propos, dans la littérature, les résultats de reconnaissance sont en général uniquement présentés avec l'apport d'un modèle de langage.

Le tableau 2.5 et la figure 2.9 présentent les résultats de reconnaissance phonétique, avec des modèles à 24 gaussiennes, sur le corpus de test en fonction du type d'initialisation de l'apprentissage et du poids d'un modèle de langage bi-gramme phonétique :

| Initialisation à partir de l'Alignement Acoustique et Phonétique (IAAP) | | | | | | |
|---|---|---|---|---|---|---|
| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
| 24 | 0 | 1 639 | 6 749 | 2 583 | 72,28 % | 63,74 % |
| 24 | 1 | 1 974 | 6 295 | 1 680 | 72,67 % | 67,12 % |
| 24 | 2 | 2 266 | 5 939 | 1 274 | 72,88 % | 68,67 % |
| 24 | 4 | 2 790 | 5 526 | 900 | 72,52 % | 69,54 % |
| 24 | 6 | 3 266 | 5 312 | 704 | 71,65 % | 69,32 % |

| Initialisation Globale Moyenne (IGM) | | | | | | |
|---|---|---|---|---|---|---|
| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
| 24 | 0 | 1 643 | 6 488 | 2 491 | 73,13 % | 64,90 % |
| 24 | 1 | 1 955 | 6 062 | 1 649 | 73,51 % | 68,06 % |
| 24 | 2 | 2 207 | 5 742 | 1 218 | 73,73 % | 69,70 % |
| 24 | 4 | 2 670 | 5 363 | 860 | 73,45 % | 70,61 % |
| 24 | 6 | 3 155 | 5 147 | 683 | 72,56 % | 70,31 % |

Tableau 2.5 : Résultats de reconnaissance en fonction
du type d'initialisation de l'apprentissage et du poids du modèle de langage

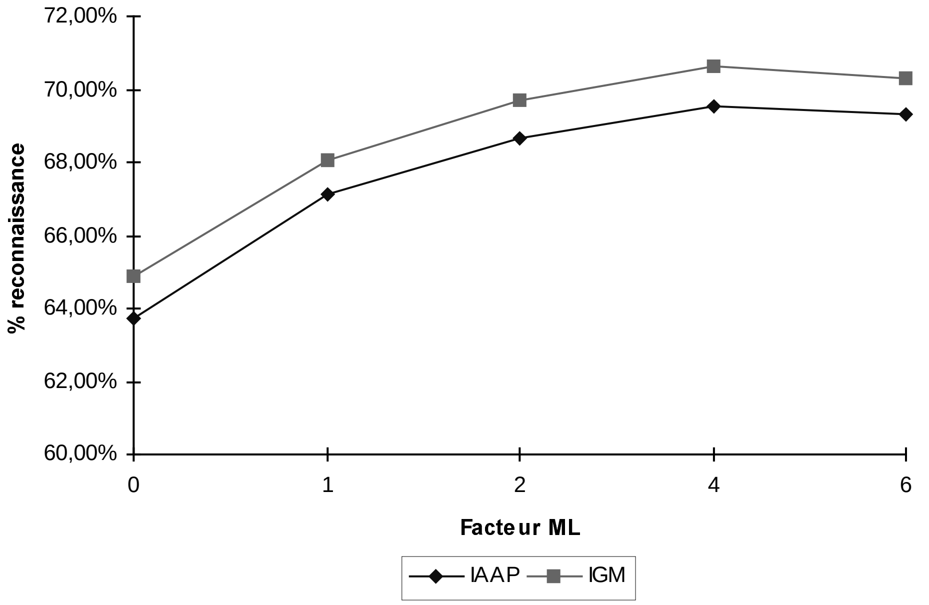


Figure 2.9 : Résultats de reconnaissance en fonction
du type d'initialisation de l'apprentissage et du poids du modèle de langage

Ces résultats montrent l'influence du modèle de langage sur les résultats de reconnaissance. Nous remarquons que plus le poids des probabilités linguistiques augmente par rapport aux probabilités acoustiques, plus le nombre de substitutions et d'insertions diminue, et le nombre d'omissions augmente. L'optimisation du poids du modèle consiste donc à "régler" les nombres d'omissions, de substitutions et d'insertions pour obtenir le meilleur taux de reconnaissance possible sur un corpus donné. Généralement, cette optimisation est réalisée empiriquement sur un corpus de développement représentatif de l'application ciblée.

Dans le cadre de ce système de DAP, l'optimum est un poids de valeur 4 sur le ML bi-gramme phonétique.

#### 2.9.3.3. Résultats en fonction du nombre d'itérations de la phase de ré-estimation

À partir des paramètres optimaux obtenus grâce aux expériences précédentes (soit les HMM à 24 gaussiennes et un poids de facteur 4 sur le ML), nous présentons dans le tableau 2.6 et la figure 2.10 les résultats de reconnaissance phonétique sur le corpus de test en fonction du nombre d'itérations de la phase de ré-estimation :

Initialisation à partir de l'Alignement Acoustique et Phonétique (IAAP)

| Nb itérations | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|
| 1 | 3 018 | 6 467 | 983 | 68,65 % | 65,41 % |
| 5 | 2 980 | 6 400 | 976 | 69,00 % | 65, 78 % |
| 10 | 2 790 | 5 526 | 900 | 72,52 % | 69, 54 % |
| 15 | 2 769 | 5 449 | 899 | 72,84 % | 69,87 % |
| 20 | 2 775 | 5 392 | 894 | 73,01 % | 70,06 % |

Initialisation Globale Moyenne (IGM)

| Nb itérations | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|
| 1 | 2 703 | 5 489 | 891 | 72,93 % | 69,98 % |
| 5 | 2 670 | 5 436 | 866 | 73,21 % | 70,35 % |
| 10 | 2 670 | 5 363 | 860 | 73,45 % | 70,61 % |
| 15 | 2 692 | 5 346 | 854 | 73,44 % | 70,61 % |
| 20 | 2 658 | 5 352 | 847 | 73,53 % | 70,73 % |

Tableau 2.6 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre d'itérations de la phase de ré-estimation

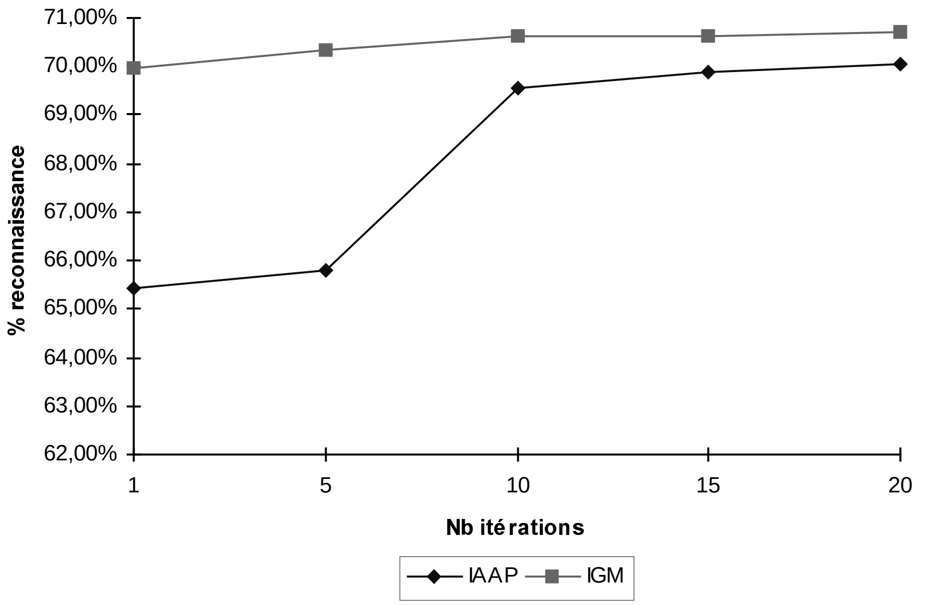


Figure 2.10 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre d'itérations de la phase de ré-estimation

Ces résultats montrent l'optimisation progressive des paramètres des HMM au cours de la phase d'apprentissage. Si le nombre de gaussiennes (le nombre de paramètres des HMM) est l'un des facteurs principaux pour les temps de calcul des phases d'apprentissage et de reconnaissance, le nombre d'itérations de la phase de ré-estimation n'influence, quant à lui, que la phase d'apprentissage. L'augmentation des performances de reconnaissance suivant une progression asymptotique (cf. figure 2.10), nous considérons le nombre d'itérations optimal pour cette expérience entre 15 et 20.

Par économie, nous n'avons poussé la phase de ré-estimation à son terme que sur les HMM à 24 gaussiennes.

#### 2.9.3.4. Résultats en fonction du type d'initialisation

Dans les expériences précédentes, nous présentons les résultats obtenus avec les deux types d'initialisation proposés pour les HMM. Quels que soient les critères évalués lors de ces expériences, l'initialisation globale moyenne donne de meilleurs taux de reconnaissance que l'initialisation à partir de l'alignement acoustique et phonétique. L'initialisation globale moyenne est plus efficace qu'une initialisation basée sur un alignement strictement automatique. L'expérience sur le nombre d'itérations de la phase de ré-estimation illustre notre propos ; en effet, dans les premières itérations, l'écart des taux de reconnaissance entre les deux types d'initialisation est particulièrement significatif.

Par ailleurs, cette expérience montre que la qualité de la phase d'initialisation de l'apprentissage des HMM est directement liée à la qualité des modèles finaux. Les erreurs de phonétisation (mauvaises correspondances entre phonétique et acoustique) et d'alignement (frontières mal placées) sont préjudiciables à la qualité de l'initialisation de l'apprentissage. C'est pourquoi, l'alignement des données acoustiques et phonétiques est généralement contrôlé et corrigé dans un processus semi-automatique. Il semble qu'un alignement strictement automatique ne puisse être utilisé directement.

### 2.9.4. Comparaison des résultats du DAP

Finalement, dans le cadre du DAP, nous obtenons un résultat de reconnaissance de l'ordre de 70,7 %[1], avec un intervalle de confiance à 95 % de [70,21 % ; 71,24 %].

Nous pouvons confronter ce résultat avec ceux obtenus par [Fohr, 96] et [Lamel, 93] qui annoncent respectivement 72,5 % et 81,7 % de reconnaissance. Toutefois, cette confrontation nécessite quelques précisions concernant les différentes conditions des expériences réalisées. Deux facteurs de différenciation sont en effet déterminants : le type de modèle de Markov et les bases de données utilisées.

---

[1] Avec la prise en compte des silences, nous obtenons 71,19 % de reconnaissance.

Nos résultats sont proches de ceux présentés dans [Fohr, 96] qui annonce 72,5 % de reconnaissance. Ceci s'explique par une relative similitude pour les paramètres adoptés dans les deux systèmes de reconnaissance (nombre de phonèmes, pas de phonèmes contextuels, multi-gaussiennes). Par ailleurs, une procédure d'étiquetage automatique est également mise en œuvre. Les différences essentielles de [Fohr, 96] avec notre module de DAP sont l'emploi de modèle de Markov du second ordre et l'utilisation d'une petite partie de BREF-80 (20 locuteurs extraits du corpus d'apprentissage) pour le corpus de test.

En ce qui concerne [Lamel, 93] qui annonce 81,7 % de reconnaissance, nous observons l'emploi de 428 modèles dépendants du contexte dont l'apprentissage a été rendu possible grâce à l'utilisation du corpus BREF.

## 2.10. Conclusion

Dans cette deuxième partie, nous avons réalisé un système de DAP fondé sur des modèles acoustiques Markovien indépendants du contexte.

Nous avons commencé par préciser le cadre de notre travail, ses ambitions mais aussi ses limites. D'une manière général, le principe adopté est une automatisation de l'ensemble des processus de construction d'un système de reconnaissance. La description de ces processus permet d'exposer les différentes problématiques abordées lors de ce travail.

Après une présentation succincte des modules logiciels destinés à la construction de SRAP et des bases de données utilisées, nous avons proposé un processus de transcription graphèmes-phonèmes automatique adapté à la construction d'un SRAP. Ce module est fondé sur une approche à base de règles et sur un module d'étiquetage. Il permet notamment de traiter les sigles, les noms propres, génère des liaisons et des variantes de prononciations. En fonction du type de donnée, nous avons utilisé deux approches différentes : l'approche hors contexte qui consiste à appliquer systématiquement les règles de génération de variantes (pour les corpus d'apprentissage et de test) ; l'approche en contexte qui utilise les différentes phonétisations observées dans un corpus de phrases phonétisées par un système de transcription graphèmes-phonèmes pour la synthèse (pour les lexiques de prononciation).

C'est à partir des différentes variantes proposées par ce phonétiseur qu'une procédure d'alignement automatique détermine une suite de phonèmes optimums et son alignement acoustique correspondant. Nous avons évalué l'impact de deux types d'initialisation des modèles acoustiques (machines de Bakis standards) pour la phase d'apprentissage. Le premier type d'initialisation utilise un alignement acoustique et phonétique ; le second une initialisation globale moyenne qui consiste à calculer globalement les moyennes et variances des HMM sur l'ensemble des données d'apprentissage.

Pour terminer cette deuxième partie, nous avons réalisé des expériences de décodage acoustico-phonétique et une comparaison des résultats obtenus. Le taux de reconnaissance phonétique final de 70,7 % valide les éléments réalisés et le principe d'automatisation de l'ensemble des processus de construction du système pour le DAP. Par ailleurs, ces expériences ont montré qu'une initialisation globale moyenne des HMM était plus efficace qu'une initialisation basée sur un alignement acoustico-phonétique automatique non contrôlé.

Cependant, ces résultats sont en deçà de l'état de l'art. Sans vouloir réécrire le chapitre *2.7.2. Perfectionnements*, nous rappelons ici les améliorations qui devraient nous permettre d'augmenter significativement les taux de reconnaissance. Tout d'abord, il est essentiel de passer à des modèles dépendants du contexte et à des multi-modèles hommes/femmes ; pour cela, une augmentation significative du corpus d'apprentissage est indispensable et l'utilisation de la base de donnée BREF dans sa totalité s'avère incontournable. Par ailleurs, une vérification semi-automatique de la phonétisation et de l'alignement acoustico-phonétique du corpus d'apprentissage devrait permettre d'améliorer les résultats. À la vue de la taille du corpus utilisé, ce travail "titanesque" nous semble difficilement envisageable. En général, ce type de vérification n'est réalisé que sur un sous-ensemble des données (notion de *bootstrap*). Mais, dans ce cas, une procédure d'alignement automatique doit être utilisée ; ce qui peut pénaliser l'apprentissage. À ce propos, il serait intéressant de quantifier le volume minimum de données à vérifier pour obtenir une meilleure initialisation des HMM par rapport à l'initialisation globale moyenne.

# Troisième Partie

# Réalisation d'un système de reconnaissance : Accès Lexical et Modélisation du Langage

La troisième partie est consacrée au passage du niveau phonétique au niveau mot. À partir des modèles acoustiques phonétiques issus du module de DAP, nous ajoutons l'accès lexical et la modélisation du langage pour constituer le SRAPC.

Tout d'abord, nous présentons le type d'accès lexical et de modélisation linguistique utilisé dans le cadre de la réalisation d'un SRAPC de référence. Notons que les principes définis dans les chapitres précédents sont conservés. L'objectif est de définir un SRAPC "minimum" fondé sur les mêmes techniques que le système de DAP décrit dans la deuxième partie de ce document.

Puis, nous proposons deux stratégies de reconnaissance alternatives. La première est une stratégie multi-passes fondée sur une simplification du lexique de décodage (lexique homophone) et un processus de filtrage acoustico-linguistique permettant de déterminer l'hypothèse optimale. La seconde stratégie vise à inclure des ressources linguistiques diverses (pour le moment un tri-gramme) directement dans un processus de décodage dérivé de l'algorithme de parcours de graphe A*.

Des expériences de reconnaissance sur ces SRAP permettent de comparer les différentes approches proposées.

## 3.1. Présentation d'un SRAP de référence

Que ce soit pour le DAP ou la Reconnaissance Automatique de Parole Continue (RAPC) de type machine à dictée, les principes des algorithmes utilisés restent les mêmes. Malgré cela, la transition d'un système de DAP à un système parole continue de type machine à dictée pose de nouveaux problèmes, notamment dus à une forte augmentation de complexité du processus de décodage et aux nouvelles ressources nécessaires.

La figure 3.1 rappelle la structure d'un module de décodage en RAP :

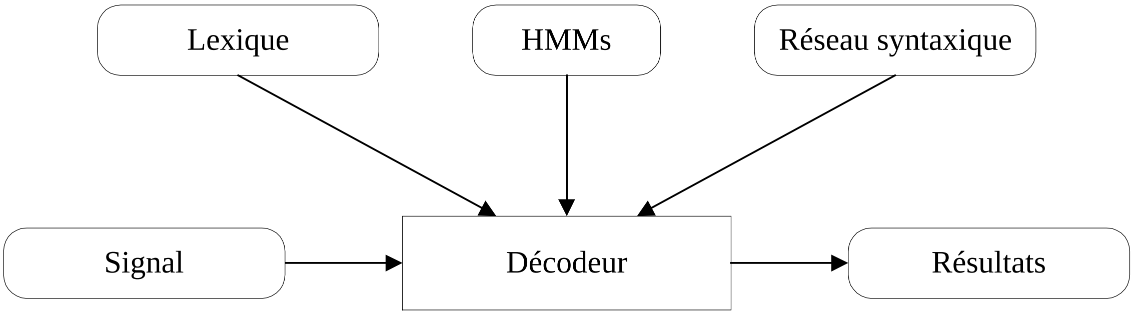


Figure 3.1 : Structure d'un module de décodage en RAP

Dans le cadre d'un SRAP de type machine à dictée, les différences par rapport à un module de DAP proviennent essentiellement du lexique de décodage (et de son réseau syntaxique associé) qui passe de quelques dizaines d'unités phonétiques (33 unités phonétiques dans le module de DAP) à des milliers de mots (éventuellement composés de ces 33 unités phonétiques).

Pour notre part, nous conservons les principes définis dans les chapitres précédents (cf. *2.1. Cadre de l'étude*) et présentons le type d'accès lexical et de modélisation linguistique utilisé pour la fabrication d'un SRAPC de référence dans un processus automatique. L'objectif est de définir un SRAPC "minimum" fondé sur les mêmes techniques que le système de DAP décrit dans la deuxième partie de ce document : à savoir, un graphe de décodage et son modèle de langage bi-gramme associé.

### 3.1.1. Lexique

Dans l'introduction de l'article "Les lexiques de prononciation dans les systèmes de reconnaissance de la parole" [Lamel, 96], L. F. Lamel présente la problématique de la modélisation lexicale dans les SRAP. Elle affirme que, lors du développement d'un système de reconnaissance, la création d'un lexique de prononciation est reconnue comme étant une phase décisive. Notamment grâce à l'apport explicite de connaissances linguistiques qui, choisies en vue de l'efficacité globale du système, amènes un gain réel.

La création d'un lexique comprend souvent une grande part de travail non automatique, et requiert des connaissances et une expertise qui sont difficiles à quantifier et à transmettre.

La construction d'un lexique de prononciation pour un SRAP se décompose en deux étapes : le choix des mots du vocabulaire utilisés et leur représentation phonétique. Notre travail a essentiellement consisté à étudier le problème de la phonétisation automatique pour la RAP (cf. deuxième partie de ce document).

### 3.1.2. Représentation du lexique et prise en compte d'un Modèle de Langage bi-gramme

Les chapitres *1.5.2.2. Reconnaissance de mots connectés* et *1.5.2.3. Prise en compte des contraintes syntaxiques dans le Modèle de Propagation de Jeton* présentent la démarche utilisée pour intégrer le lexique de prononciation et les contraintes syntaxiques dans un processus synchrone de RAP. Nous rappelons que cette démarche consiste à construire un graphe de recherche *ad hoc* modélisant les suites de mots possibles en intégrant un ML bi-gramme, et un lexique de prononciation comportant les différentes prononciations des mots du vocabulaire.

Le nombre de mots du lexique influence donc directement la taille du graphe de recherche [Ney, 84]. Par conséquent, dans le cadre d'application de RAP à grand vocabulaire, les graphes engendrés de cette façon peuvent devenir rapidement incompatibles avec des temps de réponses raisonnables[1].

## 3.2. Reconnaissance

Nous présentons dans ce chapitre un SRAPC de référence utilisant trois lexiques de 20 000 mots et leur ML bi-gramme associé.

### 3.2.1. Paramètres du SRAPC de référence

Nous rappelons que le corpus de test (8 183 mots), les lexiques et les ML utilisés dans ces expériences proviennent des ressources mises à la disposition des équipes de recherche participant à l'ARC-B1.

Pour ces tests de reconnaissance, nous avons utilisé trois lexiques de 20 000 mots. Le premier lexique (Lex 1) et son ML bi-gramme associé sont issus de la campagne B1-DV1 de test de systèmes de reconnaissance ; sa phonétisation est réalisée avec l'approche hors contexte. Ce lexique est utilisé pour comparer les résultats obtenus par le SRAPC de référence avec ceux présentés lors des JST 97. Les second et troisième lexiques (respectivement Lex 2 et Lex 3) correspondent aux 20 000 mots les plus fréquents extraits du journal *Le Monde* (années 1992 et 1993). La différence entre ces lexiques provient du mode de phonétisation utilisé. Lex 2 est phonétisé avec l'approche hors contexte et Lex 3 avec l'approche en contexte. Ces deux lexiques nous permettent de comparer, d'une part, les résultats obtenus par le SRAPC de référence en fonction des approches de phonétisation. D'autre part, Lex 3 étant utilisé dans les stratégies de décodages alternatives proposées par la suite, il nous permet également d'évaluer ces stratégies par rapport au SRAPC de référence.

---

[1] Le terme "raisonnable" est ici difficile à quantifier tant il est dépendant des ressources de calculs disponibles (puissance, mémoire, etc.) pour une tâche de reconnaissance. À ce propos, il est difficile de trouver, dans la littérature, des informations sur ces ressources et temps de calculs nécessaires aux SRAP.

Le tableau 3.1 présente des exemples de phonétisation en fonction des approches proposées. Ces exemples illustrent notamment la surgénération de variantes dans le cadre de liaisons (cf. caractères gras).

Approche hors contexte (Lex 2)

| | |
|---|---|
| abaissée | aa bb ai ss ei |
| abaissement | aa bb ai ss eu mm an |
| **abaissement** | **aa bb ai ss eu mm an tt** |
| abaissement | aa bb ai ss mm an |
| **abaissement** | **aa bb ai ss mm an tt** |
| abaisser | aa bb ai ss ei |
| abandon | aa bb an dd on |
| abandonnant | aa bb an dd on nn an |
| **abandonnant** | **aa bb an dd oo nn an tt** |

Approche en contexte (Lex 3)

| | |
|---|---|
| abaissée | aa bb ai ss ei |
| abaissement | aa bb ai ss eu mm an |
| abaissement | aa bb ai ss mm an |
| abaisser | aa bb ai ss ei |
| abandon | aa bb an dd on |
| abandonnant | aa bb an dd oo nn an |

Tableau 3.1 : Lexique de décodage et phonétisation

Le tableau 3.2 présente le nombre d'unités phonétiques obtenues sur les lexiques de 20 000 mots Lex 1, Lex 2 et Lex 3 en fonction du mode de phonétisation. Notons la réduction du nombre d'unités phonétiques liées à l'utilisation de l'approche en contexte (cf. Lex 2 et Lex 3).

| | Lex 1 | Lex 2 | Lex 3 |
|---|---|---|---|
| Nb unités phonétiques - Approche hors contexte | 42 146 | 39 475 | X |
| Nb unités phonétiques - Approche en contexte | X | X | 28 682 |

Tableau 3.2 : Nombre d'unités phonétiques dans Lex 1, Lex 2 et Lex 3
(X désigne les lexiques non phonétisés)

Quant aux modèles acoustiques, ils sont identiques à ceux utilisés pour évaluer le module de DAP décrit dans la deuxième partie de ce document.

### 3.2.2. Résultats de reconnaissance

Le tableau 3.3 présente les résultats obtenus à partir du SRAPC de référence et des ressources exposées ci-dessus :

| | % MHV | Nb omissions | Nb substitutions | Nb insertions | % ident. | % reco. |
|---|---|---|---|---|---|---|
| Lex 1 | 3,75 % | 456 | 2 054 | 350 | 69,33 % | 65,05 % |
| Lex 2 | 3,83 % | 476 | 2 102 | 331 | 68,50 % | 64,45 % |
| Lex 3 | 3,83 % | 453 | 2 099 | 340 | 68,81 % | 64,66 % |

Tableau 3.3 : Résultats de reconnaissance - Lexiques et ML 20k

Les critères de l'évaluation (% identification et % reconnaissance) utilisés pour ces expériences sont similaires à ceux définis dans le chapitre *2.9.1.2. Critères de l'évaluation.*

Quelles que soient les ressources utilisées, les résultats obtenus sur le SRAPC de référence sont équivalents. Le taux de reconnaissance moyen est de 64,72 %, avec un intervalle de confiance à 95 % de [63,67 % ; 65,74 %].

Ce résultat montre que l'approche de phonétisation de lexique de prononciation en contexte permet de réduire le nombre d'unités dans le lexique de décodage tout en conservant le même taux de reconnaissance.

Nous pouvons confronter les résultats obtenus par le SRAPC de référence avec ceux présentés lors des JST 97 [Dolmazon, 97]. Tout d'abord avec [Fohr, 97] qui annonce 53 % de reconnaissance (résultats partiels obtenus sur 11 locuteurs représentant 68 phrases issues du corpus) ; puis, avec [Savariaux, 97] qui annonce 58,28 % de reconnaissance ; et enfin, avec [Adda, 97a] qui annonce 87,54 % de reconnaissance. Bien que les paramètres et données utilisés dans ces expériences de reconnaissance ne soient pas strictement identiques (nombre de phrases du corpus de test, phonétisation du lexique, procédure d'évaluation, etc.), les résultats obtenus par le système de référence sont comparables à ceux présentés dans [Fohr, 97] et [Savariaux, 97]. Quant aux performances présentées dans [Adda, 97a], elles sont bien supérieures à celles du SRAPC de référence. Ceci s'explique par la qualité et la cohérence de l'ensemble des modules du système et par une stratégie de décodage éprouvée[1]. Mais, des différences significatives sont à souligner. Nous pensons, en particulier, à l'emploi de modèles acoustiques contextuels et à l'utilisation d'un lexique de 65 000 mots dans les passes de décodage ; les mots identifiés mais non présents dans le lexique de 20 000 mots de référence étant retirés de l'hypothèse de reconnaissance finale. De notre point de vue, c'est essentiellement ce dernier point qui rend délicate une comparaison avec ce troisième système. Rappelons cependant que le SRAPC de référence utilise un ensemble de paramètres et une stratégie de reconnaissance élémentaire (pas de multi-modèles acoustiques, pas de modèles contextuels, pas de stratégies multi-passes, etc.), contrairement aux trois systèmes précités.

Les résultats obtenus sur le SRAP de référence mettent en évidence certaines insuffisances qui poussent à ajouter de nouvelles connaissances. Nous pensons, en particulier, à la mise en place de modèles linguistiques plus puissants comme, par exemple, des tri-grammes et tri-classes. Mais, si le SRAP "idéal" devrait utiliser l'ensemble des connaissances mises à sa disposition dans un processus simultané et synchrone, les contraintes algorithmiques et matérielles poussent à développer des stratégies de décodage alternatives visant à décomposer le problème ; notamment dans les SRAP à grand vocabulaire où les stratégies dites multi-passes s'avèrent maintenant incontournables.

Par ailleurs, à l'instar du DAP présenté dans la deuxième partie, ce SRAPC de référence nécessiterait un certain nombre de perfectionnements qui permettraient, sans aucun doute, une amélioration significative des résultats de reconnaissance. En effet, pour obtenir de bonnes performances, un travail d'expertise (choix des mots du vocabulaire, correction de la phonétisation, etc.) sur le lexique est indispensable ; les processus automatiques ne pouvant résoudre efficacement l'ensemble des problèmes posés par la modélisation lexicale. Du point de vue de l'espace de recherche à explorer par le décodeur, la composante lexicale est l'un des facteurs principaux de complexité. C'est pourquoi, un SRAPC grand vocabulaire intègre nécessairement des processus d'optimisation. En ce qui concerne l'optimisation de l'accès lexical, ce sont essentiellement des techniques de compression du graphe de recherche qui sont utilisées [Lacouture, 95] ; puis, une réduction du nombre d'opérations à effectuer peut être obtenue par l'utilisation de faisceaux d'émondage [Lowerre, 90]. De nombreux exemples d'optimisations (multi-modèles, multi-gaussiennes, partages d'états, phonèmes contextuels, phonèmes généralisés, graphe de recherche compressé, faisceaux d'émondage, N-best, etc.) destinés aux différentes composantes des SRAP sont décrits dans la littérature [Lee, 90a ; Lee, 90b ; Lowerre, 90 ; Lokbani, 93 ; Jouvet, 94 ; Young, 94b ; Lacouture, 95 ; Ravishankar, 96][1].

---

[1] Le nombre et la qualité des publications sur les SRAP du LIMSI en attestent.

## 3.3. Reconnaissance et stratégies de décodage

Parmi les perfectionnements possibles, deux stratégies de décodages alternatives sont proposées dans ce chapitre.

La première est une stratégie multi-passes fondée sur une simplification du lexique de décodage et un processus de filtrage acoustico-linguistique permettant de déterminer l'hypothèse optimale. La seconde vise à inclure des ressources linguistiques diverses (pour le moment un tri-gramme) directement dans le processus de décodage.

### 3.3.1. Approche homophone

Comme nous l'avons vu dans le chapitre *3.1. Présentation d'un SRAPC de référence*, la taille du lexique détermine directement la dimension du graphe de recherche. Nous proposons une nouvelle composante linguistique, fondée sur les propriétés homophoniques de la langue française. Elle autorise une réduction du nombre d'unités dans le lexique lors de la phase de décodage acoustique ; son adoption, dans un SRAPC dit homophone, assure une couverture lexicale identique tout en diminuant significativement la complexité des traitements.

Le propos de cette approche n'est pas de comparer cette option de codage du lexique sous forme d'homophone et les techniques classiques de compression de graphe de recherche. À notre avis, la seconde solution est bien plus performante. Notre objectif est d'étudier un décodeur utilisant des homophones, en particulier du point de vue de l'apport de la phase linguistique finale.

---

[1] Dans le domaine "fécond" de l'optimisation des SRAP, ces références ne constituent qu'un petit exemple.

Ce système est composé de deux passes : à partir des résultats d’une première passe de décodage sur un lexique homophone, une seconde passe permet de déterminer l’hypothèse otpimum.

**Remarque**

L'utilisation d'homophones dans un lexique de décodage n'est intéressante que pour une langue à fort taux d'homophonie telle que le français.

### 3.3.1.1. Réduction du lexique de décodage

La correspondance entre phonème et graphème n'est pas univoque. Certaines suites de sons peuvent être écrites de diverses façons : ce sont les homophones hétérographes.

Nous définissons une *liste homophone* comme l'ensemble des suites de mots ayant le même nombre de mots, une même segmentation et dont chaque mot est homophone pour un même rang (cf. figure 3.2).

$M_{i,x}$ et $M_{i,y}$ sont homophones $\forall i,x,y$

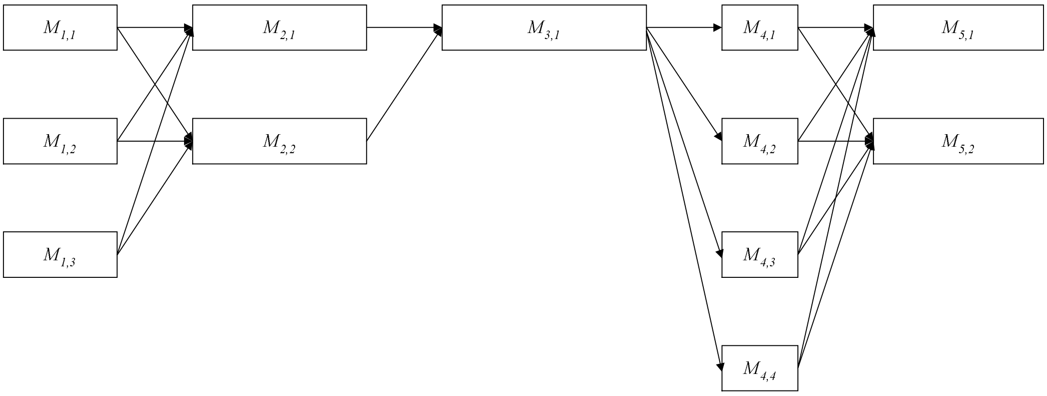


Figure 3.2 : Liste homophone

Habituellement, l'approche statistique du problème de la reconnaissance automatique de la parole, issue de la théorie de l'information, est présentée comme la combinaison d'un module acoustique et d'un module linguistique.

D'un point de vue exclusivement acoustique, les ambiguïtés liées aux homophones ne peuvent être levées. En effet, la probabilité acoustique de toutes les suites de mots d'une *liste homophone* est par définition identique : le décodage acoustique de tous les mots d'une liste homophone est redondant. Nous proposons de reporter le traitement de ce type d'ambiguïté dans une phase linguistique effectuée en post-traitement. Nous remplaçons le lexique de décodage par la liste des classes d'homophones hétérographes correspondantes et le modèle de langage par un ML bi-classe adapté à ce nouveau "lexique".

### 3.3.1.2. Première passe : décodage homophone

La première passe consiste à utiliser un lexique homophonique et son modèle de langage probabiliste associé dans un processus de décodage de type N-meilleures solutions.

Le lexique et la composante linguistique homophone utilisés lors de cette première passe de décodage sont extraits d'un corpus phonétisé en mode contextuel, issu des années 1992 et 1993 du journal *Le Monde*. Le lexique homophone correspond donc au lexique Lex 3 regroupé en classe d'homophone.

La tableau 3.4 présente un extrait d'un lexique de décodage homophonique :

| | |
|---|---|
| abaissée_abaisser | aa bb ai ss ei |
| abaissement | aa bb ai ss eu mm an |
| abaissement | aa bb ai ss mm an |
| abandon_abandons | aa bb an dd on |
| abandonnant | aa bb an dd oo nn an |
| abandonne_abandonnent | aa bb an dd oo nn |
| abandonnés_abandonnées | aa bb an dd oo nn ei zz |
| abandonner_abandonné_abandonnée_<br>abandonnés_abandonnées | aa bb an dd oo nn ei |

Tableau 3.4 : Lexique de décodage homophonique
(les différents homophone sont séparés pas le caractère '_')

Le tableau 3.5 présente le taux de réduction obtenu sur le lexique homophone par rapport au lexique référence de 20 000 mots Lex 3 :

| | Nb unités phonétiques | Nb unités homophones | Taux de réduction |
|---|---|---|---|
| Lex 3 (20k) | 28 682 | 21 559 | 24,86 % |

Tableau 3.5 : Taux de réduction homophone sur 20 000 mots

En sortie de cette première passe, le système produit N liste homophone solution (cf. tableau 3.6) :

```
après  quoi   Lee   parties  pourrons  toujours   fers   appellent
             -les   partis   pourront              fer     appelle
              les    parti                        faire   appels
                    partie                                 appel
```

Tableau 3.6 : Exemple de liste homophone solution
(décodage de la phrase L12M081 du corpus de test)

Dans cette exemple, nous pouvons noter une erreur sur le mot "Lee" qui est phonétisé /lɛ/.

**Remarque**

Le nombre de phrases possibles issues d'une liste homophone peut être élevé. À titre d'exemple, la liste homophone issue de L12M081 a un potentiel de 288 phrases différentes ; quant à la liste homophone issue de L01F088, le nombre de phrases possibles est de l'ordre de $10^{21}$.

### 3.3.1.3. Seconde passe : détermination de l’hypothèse optimale

À partir des N listes d’homophones provenant de la première passe, nous proposons d’appliquer un module de filtrage acoustico-linguistique afin de déterminer l'hypothèse optimale.

Ce module de filtrage est dérivé du système d'étiquetage automatique ECSta développé au LIA.

**Présentation du système ECSta**

Dans la problèmatique de l'étiquetage automatique de texte, deux types d'approches sont généralement proposées : les approches probabilistes [Derouault, 85 ; Church, 89 ; Church, 93] et les approches à base de règles [Brill, 92 ; Chanod, 94]. Le système d'Etiquetage par Contraintes et Statistique (ECSta) propose de montrer le bénéfice double que laisse espérer un rapprochement des ces deux approches.

Le fondement d'ECSta est un système d'étiquetage syntaxique statistique utilisant des combinaisons de modèles tri-classes et tri-grammes [El-Bèze, 93]. Un algorithme permet de fusionner le modèle statistique avec une grammaire décrivant des formes d'agrammaticalité [El-Bèze, 95 ; Spriet, 95 ; Spriet, 97a]. L'objectif est le suivant : même si la grammaire n'est pas exhaustive, toutes les erreurs dues aux imperfections du modèle statistique qu'elle corrige seront autant de points rajoutés au taux de réussite. L'algorithme de parcours de graphe est fondé sur l'algorithme A*. Les règles d'agrammaticalité peuvent être appliquées à chaque nœud du graphe de recherche en pénalisant la fonction de coût global (le chemin concerné est déclassé dans la pile des hypothèses)[1]. Une attention particulière est apportée pour préserver la consistance de l'algorithme de parcours.

ECSta est à la base de plusieurs applications développées au LIA telles que les modules de phonétisation ou un système d'accentuation automatique [Spriet, 97b].

**Module de filtrage acoustico-linguistique**

Nous proposons un module de filtrage acoustico-linguistique fondé sur le système ECSta. L'objectif est de donner le meilleur chemin dans une liste homophone.

---

[1] Nous présentons A* dans le chapitre *3.3.2. Approche A**.

À partir des listes homophones solutions issues de la première passe, nous associons à chacun des mots $M_{i,j}$ l'ensemble des catégories syntaxiques possibles $C_j$ avec leur probabilité acoustique associée. Puis, nous combinons les probabilités acoustiques et linguistiques (modèles de langage probabiliste tri-classe et tri-gramme). Nous définissons la probabilité $P(M_i)$ d'un mot $M_i$ de la phrase $M$ tel que :

$$P(M_i) = s(\alpha P_{tri-classe} + (1-\alpha) P_{tri-gramme})^f \times P_{acoustique} \tag{Eq. 3.1}$$

Avec :

- $s$ le facteur de pénalité linguistique qui influence le nombre de mots de la phrase.
- $\alpha$ le coefficient de combinaison linéaire entre les ML tri-classe et tri-gramme qui détermine leurs poids respectifs dans la composante linguistique.
- $f$ le facteur qui détermine le poids de la composante linguistique par rapport à la composante acoustique.

Enfin, à l'instar de la version initiale ECSta, l'algorithme A* permet de prendre en compte des contraintes diverses sous formes de règles. Ces contraintes peuvent s'appliquer sur l'historique complet du chemin courant. Le résultat de la phase de filtrage acoustico-linguistique est alors la phrase de probabilité maximale.

Les modèles de langages probabilistes tri-classe et tri-gramme utilisés dans les expériences de reconnaissance sont estimés à partir d'un corpus de 40 millions de mots (1,6 millions de phrases) extraits du journal *Le Monde* (années 1992 et 1993). Les ponctuations sont ignorées. Le lexique est composé des 64 000 mots les plus fréquents du corpus. Les règles d'agrammaticalité sont celles du système ECSta.

La tableau 3.7 présente un exemple de filtrage acoustico-linguistique (l'hypothèse retenue est en caractères gras ; l'erreur par rapport à la référence est en caractères gras et italiques) :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **après** | **quoi** | Lee | parties | pourrons | **toujours** | fers | appellent |
| | | -les | ***partis*** | **pourront** | | fer | appelle |
| | | **les** | parti | | | **faire** | appels |
| | | | partie | | | | **appel** |

Référence

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| après | quoi | les | **parties** | pourront | toujours | faire | appel |

Tableau 3.7 : Exemple de filtrage acoustico-linguistique
(décodage de la phrase L12M081 du corpus de test)

Cet exemple choisi montre les limites du modèle. En effet, comment un modèle n-gramme peut-il préférer un contexte juridique (parties) à un contexte politique (partis) ?

Pour corriger cette erreur, correcte sur les plans acoustique et linguistique, il faudrait disposer d'un module prenant en compte le contexte de la phrase.

#### 3.3.1.4. Résultats de reconnaissance

À l'instar du SRAPC de référence, les tests de reconnaissance du SRAPC homophone sont effectués sur les même phrases de test.

|  | % MHV | Nb omissions | Nb substitutions | Nb insertions | % ident. | % reco. |
|---|---|---|---|---|---|---|
| Lex homo | 3,83 % | 499 | 2 095 | 300 | 68,29 % | 64,63 % |

Tableau 3.8 : Résultat de reconnaissance - Lexique 20k homophone

Ce résultat est équivalent à ceux obtenus sur le SRAPC de référence. Il valide et montre en particulier l'intérêt du module de filtrage acoustico-phonétique présenté dans le chapitre précédent. En effet, malgré une réduction du niveau de connaissance utilisé dans la première passe de décodage homophone, la seconde passe permet de retrouver l'information perdue. Cela montre par ailleurs l'importance des connaissances linguistiques dans un processus de reconnaissance.

### 3.3.2. Approche A*

L'objectif de ce chapitre est de discuter les approches multi-passes qui ajoutent les différents niveaux de connaissances disponibles les uns à la suite des autres. En effet, nous pensons que la possibilité de fusionner dans un traitement simultané et synchrone un ensemble de connaissances acoustiques et linguistiques peut être un facteur déterminant.

Nous proposons de réaliser une stratégie de reconnaissance qui permet d'inclure directement des connaissances diverses dans le processus de décodage. Notamment, nous présentons un module de reconnaissance pouvant utiliser des modèles de langage probabilistes locaux d'ordre supérieur à deux, dans notre cas, un modèle tri-gramme.

Ce module de reconnaissance est fondé sur une implémentation de l'algorithme A* [Hart, 68 ; Nilsson, 82][1].

---

[1] En ce qui concerne la présentation de A* dans la suite de ce document, nous avons utilisé [Lokbani, 93].

#### 3.3.2.1. Fonction de coût

L'algorithme A* est fondé sur une fonction de coût $f$ définie pour chaque nœud $n_i$ du graphe de recherche. La fonction $f(n_i)$ représente une estimation du coût du chemin contraint de passer par le nœud $n_i$ et liant le nœud initial $n_0$ au nœud final $n_F$ (cf. figure 3.3).

La fonction de coût $f$ est définie telle que :

$$f(n_i) = g(n_i) + h(n_i) \qquad \text{(Eq. 3.1)}$$

Avec :

- La fonction $g(n_i)$ correspondant à la probabilité du chemin envisagé, entre le nœud initial $n_0$ et le nœud $n_i$.
- La fonction $h(n_i)$ correspondant à une estimation (dans notre cas une maximisation) de la probabilité du meilleur chemin, entre le nœud $n_i$ et le nœud $n_F$.

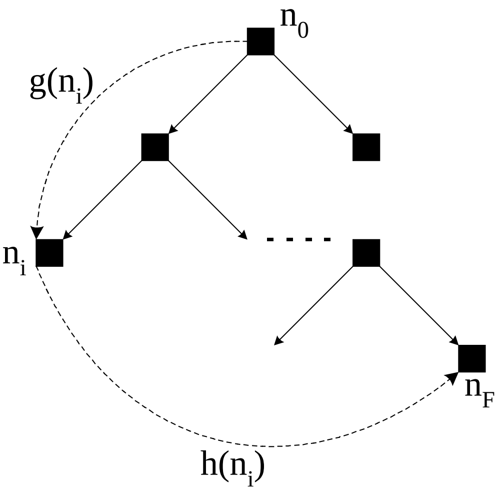


Figure 3.3 : Fonction de coût et graphe de recherche

#### 3.3.2.2. Décodage et algorithme A*

Pour retrouver le nœud final, l'algorithme A* utilise une recherche de type "meilleur d'abord" en fonction de la fonction de coût $f$ définie précédemment. Dans la procédure de recherche, nous distinguons deux sous-ensembles de nœud :

- Les nœuds qui ont été développés (dont les successeurs sont disponibles pour la procédure de recherche) sont appelés "fermés". Ils sont placés dans la liste FERMÉ.
- Les nœuds qui ont été générés et qui attendent leur développement sont appelés "ouverts". Ils sont placés dans la liste OUVERT ; cette liste doit être ordonnée en fonction du meilleur chemin.

L'algorithme de recherche est alors le suivant :

**1. Initialisation**

Mettre le nœud initial $n_0$ dans OUVERT.
Calculer $f(n_0) = g(n_0) + h(n_0)$

**2. Test d'échec**

Si OUVERT est vide sortir avec échec.

**3. Choix du nœud à traiter**

Retirer de OUVERT et mettre dans FERMÉ le nœud *n* pour lequel *f* est maximum.

**4. Est-ce la fin ?**

Si *n* est un nœud but, sortir avec la solution obtenue en remontant les pointeurs de *n* à $n_0$.

**5. Développement**

Sinon, développer *n*, pour engendrer tous les successeurs et affecter à ces derniers les pointeurs indiquant qu'ils sont successeurs de *n*. Pour tout successeur *n'* de *n* :

> **a.** Si *n'* n'est pas encore dans OUVERT ni dans FERMÉ, faire une estimation de $h(n')$ (c'est-à-dire une estimation du coût du meilleur chemin allant de n' jusqu'à un nœud but), et calculer $f(n') = g(n') + h(n')$ où $g(n') = g(n) + c(n, n')$, avec $c(n, n')$ le coût du passage entre les nœuds *n* et *n'*. Insérer *n'* dans la liste OUVERT.
>
> **b.** Si *n'* est déjà dans OUVERT ou FERMÉ, modifier éventuellement ses pointeurs pour le rattacher au chemin fournissant $g(n')$ maximum.
>
> **c.** Si *n'* a nécessité un ajustement des pointeurs et a été trouvé dans FERMÉ, le remettre dans OUVERT (cela signifie que l'on arrive au nœud *n'* par un nouveau chemin, et qu'il faut reconsidérer les développements de ce nœud).

**6. Boucle**

Aller à l'étape 2.

**Remarque**

L'efficacité de l'algorithme A* est liée à la manière dont la fonction *h* est estimée. Si *h* est un estimateur parfait, alors l'algorithme A* convergera immédiatement vers le but sans recherche inutile (sans développement de chemins inutiles). Dans le cas où *h* est sous-estimé, on ne peut avoir l'assurance de trouver le chemin solution le moins coûteux. Par contre, si *h* est surestimé, la meilleure solution sera forcément trouvée. En fait, *h* est une maximisation de la probabilité du meilleur chemin. Meilleure est h (c'est-à-dire, plus *h* est proche de l'estimateur parfait), plus la meilleure solution est atteinte rapidement.

### 3.3.2.3. Pré-calculs et définitions des fonctions de coût

La figure 3.4 présente le synoptique général de la stratégie proposée.

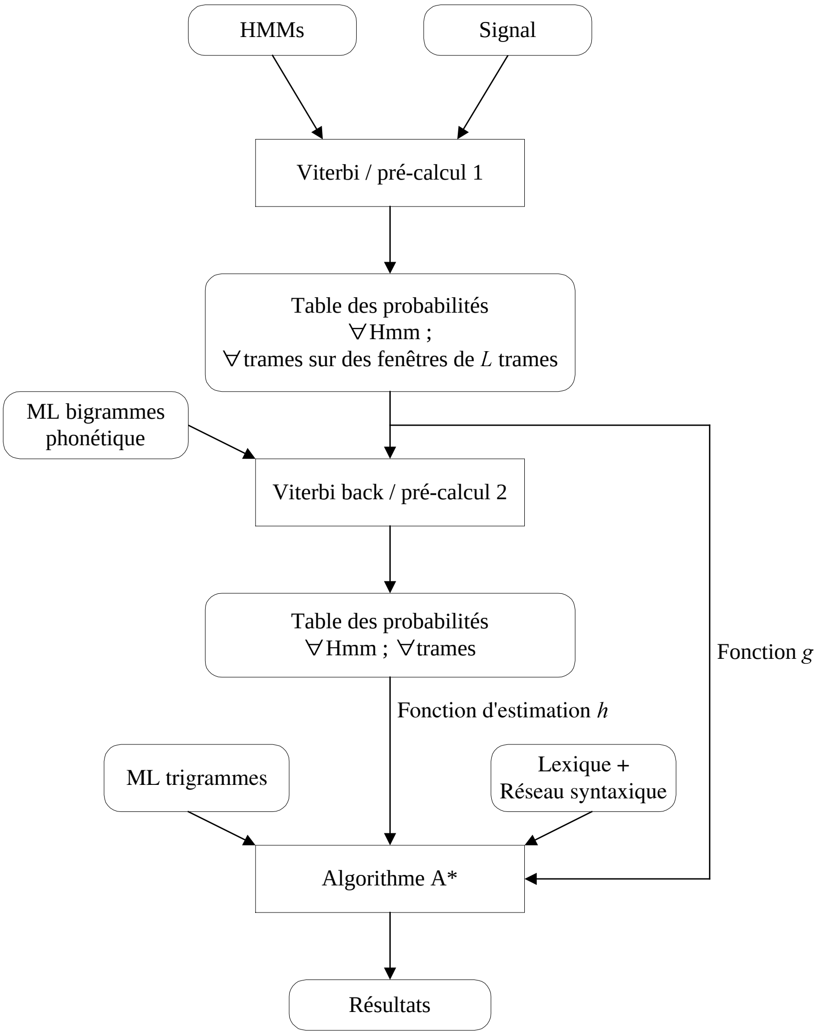

HMMs
Signal
Viterbi / pré-calcul 1
Table des probabilités
∀Hmm ;
∀trames sur des fenêtres de L trames
ML bigrammes phonétique
Viterbi back / pré-calcul 2
Table des probabilités
∀Hmm ; ∀trames
Fonction g
Fonction d'estimation h
ML trigrammes
Lexique + Réseau syntaxique
Algorithme A*
Résultats

Figure 3.4 : Synoptique général du SRAPC (approche A*)

Les étapes de pré-calculs 1 et 2 permettent de calculer les paramètres utilisés par les fonctions $g$ et $h$.

**Étape de pré-calcul 1**

Dans la première phase de pré-calcul, à partir des modèles acoustiques et du signal, nous appliquons l'algorithme de Viterbi pour chaque trame et modèles acoustiques sur des fenêtres de $L$ trames. À l'issue de ce calcul, nous disposons, sur chaque trames $t$ et sur chaque modèle acoustique, du **C**oût **Ac**oustique (*CAc*) de tous les meilleurs chemins sur une fenêtre de $L$ trames.

**Étape de pré-calcul 2 et calcul de la fonction *h***

La fonction d'estimation $h$ consiste à appliquer l'algorithme de Viterbi back avec le modèle de langage bi-gramme phonétique utilisé dans le SRAPC de référence. Cette fonction joue le rôle d'une sonde acoustique. En fait, cette étape permet de pré-calculer pour chaque nœud $n_i$ du graphe de recherche une estimation du coût du meilleur chemin entre le nœud $n_i$ et le nœud $n_F$.

**Calcul de la fonction g**
Dans le cadre du calcul de la fonction *g*, nous définissons la fonction de coût *c(n, n')* modélisant une transition entre les nœuds *n* et *n'* telle que :

- Si *n'* est un nœud correspondant à un modèle acoustique sous-lexical (par exemple, un phonème), *c(n, n') = CAc(n', l, t) + CLiAc (n, n')*, avec *CAc* le **C**oût **Ac**oustique du nœud *n'*, de longeur *l* sur la trame *t* (ou des trames *t* à *t+l*) et *CLiAc* le Coût **Li**nguistique inter-modèle **Ac**oustique entre les nœuds *n* et *n'*.

  Lié à la fonction d'estimation *h*, *CLiAc* permet de garantir la convergence de A*. Dans la mesure où nous proposons une fonction d'estimation *h* fondée sur une application de l'algorithme de Viterbi Back avec un modèle de langage probabiliste phonétique bi-gramme, *CLiAc(n, n')* doit donc être équivalent à ce même modèle de langage probabiliste phonétique bi-gramme.

  Notons que lors des tests de reconnaissance, nous avons utilisé l'algorithme de Viterbi sur des fenêtres de calcul de 40 trames, c'est-à-dire une fenêtre temporelle largement supérieure à la durée d'un phonème (cf. *Tableau 2.7 : Statistiques sur le nombre de représentants et les durées des unités phonétiques*).

- Sinon, *n'* est un nœud lexical (c'est un nœud de transition entre mots), *c(n, n') = CLiMo(n, n')*, avec *CLiMo* le **C**oût **Li**nguistique inter-**Mo**t.

**Processus de décodage**
Enfin, nous appliquons l'algorithme A* associé à un modèle de langage probabiliste tri-gramme (*CLIMo* est un ML tri-gramme).

### 3.3.2.4. Résultats de reconnaissance

Ce chapitre est consacré aux premiers résultats de l'approche A*.

Dans ces expériences, les modèles acoustiques et le ML bi-gramme[1] (pour le calcul de la fonction *h*) utilisés sont issus du module de DAP présenté dans la deuxième partie de ce document.

---

[1] Un facteur 4 est appliqué sur ce ML bi-gramme pour que cette expérience soit comparable au meilleur résultat obtenu précédemment sur le DAP initial.

Avec un lexique de 20 000 mots (Lex 3) et un ML tri-gramme estimé sur les années 92-93 du journal *Le Monde,* nous avons mené quelques expériences de décodage. Mais nous n'avons pas réussi à les mener à terme car nous sommes confrontés à un problème de convergence de l'algorithme A* sur certaines phrases du corpus de test. Ce problème, lié à la fonction d'estimation *h* et à des phases de calculs inutilement répétées, nécessite un travail conséquent d'optimisation.

Par ailleurs, nous présentons une expérience de DAP utilisant une version simplifiée de l'approche A* proposée. Dans le cadre du DAP A*, cette simplification consiste à prolonger le chemin en tête de liste par toutes les hypothèses possibles. Le corpus de test utilisé est issu du module de DAP présenté dans la deuxième partie de ce document. Le ML tri-gramme phonétique (pour le coût linguistique inter-mot) est estimé sur un corpus de 8 330 phrases. Ce corpus est composé des 5 330 phrases utilisées pour l'apprentissage du ML bi-gramme phonétique et des 3 000 phrases phonétisées par un expert. Ces dernières 3 000 phrases sont issues du thème B3 des ARC de l'AUPELF-UREF [Alessandro *et al*, 97].

Le tableau 3.9 présente les résultats obtenus sur le DAP avec l'approche A* :

| Approche | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|
| DAP initial | 2 658 | 5 352 | 847 | 73,53 % | 70,73 % |
| DAP A* | 2 226 | 4 984 | 1 098 | 76,17 % | 72,54 % |

Tableau 3.9 : Résultat de reconnaissance DAP - Approche A*

Le DAP A* conduit à un taux de reconnaissance phonétique de 72,54 %, avec un intervalle de confiance à 95 % de [72,03 % ; 73,04 %] (nous rappelons que pour 70,73 %, l'intervalle de confiance à 95 % est de [70,21 % ; 71,24 %]). Ce résultat, significativement supérieur par rapport au DAP initial, confirme l'intérêt d'une stratégie de décodage qui permet d'ajouter de nouvelles connaissances directement dans le processus de reconnaissance.

## 3.4. Conclusion

Dans cette troisième partie, nous avons étudié l'accès lexical et la modélisation du langage dans le cadre de la réalisation de SRAPC.

Dans un premier temps, nous avons présenté un SRAPC de référence s'appuyant sur les techniques exposées en deuxième partie de ce document : à savoir, un graphe de décodage et son modèle de langage bi-gramme associé. L'application de règles de transcription graphèmes-phonèmes hors et en contexte a permis de proposer deux approches de phonétisation de lexique. Puis, une évaluation des performances de reconnaissance de ce système a été réalisée sur trois lexiques et ML différents. Le premier lexique et son ML bi-gramme sont issus de la campagne AUPELF B1-DV1 ; ils ont permis de confronter le SRAPC de référence avec les systèmes présentés lors des JST 97. Comportant les mêmes unités lexicales, seule l'approche de phonétisation différencie les deuxième et troisième lexiques ; ces derniers ont été utilisés pour étudier l'impact des deux approches de phonétisation de lexique sur la reconnaissance et pour évaluer les stratégies de décodage alternatives proposées par la suite.

Les résultats obtenus sur ces trois lexiques sont équivalents. Ils montrent que l'approche de phonétisation de lexique de prononciation en contexte permet de réduire le nombre d'unités dans le lexique de décodage tout en conservant le même taux de reconnaissance. Quant à la confrontation du SRAPC de référence avec trois systèmes présentés lors des JST 97, elle indique que les performances du SRAPC de référence sont comparables à deux d'entre eux ; le troisième système obtient des performances supérieures. Notons cependant que le SRAP de référence utilise un ensemble de paramètres de base et une stratégie de reconnaissance élémentaire (une passe de décodage avec ML bi-gramme), contrairement aux trois autres systèmes. Ces résultats mettent en évidence certaines insuffisances du SRAPC de référence ; ils poussent à ajouter de nouvelles connaissances. Nous pensons, en particulier, à la mise en place de connaissances linguistiques supplémentaires dans le cadre de stratégies de décodages.

Dans un second temps, nous avons proposé deux stratégies de reconnaissance alternatives.

Tout d'abord, une stratégie multi-passes homophone, où la première passe de décodage produit des listes homophones sur lesquelles une seconde passe de filtrage acoustico-linguistique permet de déterminer l'hypothèse optimale. Le résultat de reconnaissance obtenu par cette approche homophone est d'un niveau équivalent au SRAPC de référence. Il est intéressant de noter que, malgré une réduction du niveau de connaissance utilisé dans la première passe de décodage homophone, la seconde passe permet de retrouver l'information perdue. Ce résultat valide le module de filtrage acoustico-phonétique proposé et confirme l'importance des connaissances linguistiques dans un processus de reconnaissance.

La seconde stratégie proposée a pour objectif de discuter les approches multi-passes qui ajoutent les différents niveaux de connaissances les uns à la suite des autres. À partir d'une implémentation de l'algorithme A*, cette stratégie permet d'inclure directement des connaissances diverses dans le processus de décodage. Le parcours du graphe de recherche et la fonction de vraisemblance sont fondés sur le pré-calcul des différentes fonctions de coût acoustique ; la fonction d'estimation utilise l'algorithme de Viterbi back et un ML bi-gramme phonétique. Dans le cadre de cette approche A*, avec un lexique de 20 000 mots et un ML tri-gramme, nous sommes confrontés à un problème de convergence de l'algorithme sur certaines phrases du corpus de test. Ce problème, lié à la fonction d'estimation *h* et à des phases de calculs inutilement répétées, nécessite un travail conséquent d'optimisation. Par ailleurs, nous avons obtenu un taux de reconnaissance phonétique de 72,5 % avec l'approche A* et un ML tri-gramme ; soit un gain significatif de 1,8 % par rapport au DAP initial présenté dans la deuxième partie de ce document. Ce résultat montre l'intérêt d'une stratégie de décodage permettant d'ajouter de nouvelles connaissances directement dans le processus de reconnaissance.

## Quatrième Partie

# Modèles de langage : Extraction de Sous-Modèles de Langage

Que ce soit pour la Reconnaissance Automatique de la Parole (RAP) ou, plus largement, pour le Traitement Automatique du Langage Naturel (TALN), les Modèles de Langage (ML) sont des ressources incontournables. Ils jouent en particulier un rôle très important dans les applications de dictée automatique [Jelinek, 90] qui constituent le cadre de notre travail. Mais l'obtention de modèles fiables nécessite la mise en œuvre de procédures lourdes et coûteuses en moyens informatiques. En outre, à partir de grand corpus de textes, l'apprentissage du modèle de langage pour une application donnée suppose qu'un lexique *ad-hoc* ait été défini au préalable. De ce fait, toute modification du lexique implique nécessairement un nouvel apprentissage. Afin d'éviter ces apprentissages répétés, nous proposons une méthode qui permet d'extraire directement un Sous-Modèle de Langage (SML) d'un premier Modèle de Langage (ML) calculé précédemment.

Dans ce quatrième chapitre, nous définissons tout d'abord la problématique de l'extraction des SML, en insistant particulièrement sur l'importance du choix du lexique pour la première collecte. Puis, nous appliquons cette méthode à un lexique de 64 000 mots et à son ML associé sur une suite de Sous-Lexiques. À titre indicatif, nous calculons un ensemble de mesures usuelles pour ces différents lexiques et modèles de langage.

## 4.1. Introduction

Si la présentation des différents chapitres de ce document suit bien une progression naturelle, des bases théoriques du domaine à un SRAP en passant par un DAP, le réel cheminement de nos différents travaux est loin d'être aussi logique. En gardant la métaphore du cheminement, nous pourrions dire avoir emprunté de multiples routes, détours et raccourcis, voire même quelques impasses.

Cette quatrième partie est issue de l'un de ces raccourcis...

Dans le cadre de la réalisation d'un SRAPC, nous voulions commencer par tester le système sur un petit vocabulaire. Disposant d'un lexique et de son ML associé (les données de l'ARC-B1), nous avons imaginé une procédure directe d'extraction de Sous-Modèles de Langage. C'est cette procédure que nous proposons d'exposer dans ce quatrième chapitre.

Cette étude a été présentée dans [Igounet, 97].

## 4.2. Position du problème

Tout d'abord, précisons que nous avons délibérément choisi d'inscrire cette partie de notre travail dans le cadre de la RAP et des systèmes de dictée automatique. Il conviendrait donc d'adapter certains choix si le domaine d'application de la méthode proposée était différent[1].

La fiabilité de l'estimation des paramètres constitutifs des ML ne peut être garantie que si leur apprentissage est effectué sur de grands corpus de textes [Lamel, 95]. Ceci nécessite la mise en œuvre de procédures lourdes et coûteuses en moyens informatiques[2]. En outre, le lexique d'un SRAP doit être défini avant l'estimation des paramètres du ML associé. Le choix des unités lexicales n'est pas une tâche facile et plusieurs essais sont nécessaires avant d'obtenir un résultat satisfaisant [Cerf-Danon, 91]. De ce fait, toute modification du lexique implique une nouvelle estimation des paramètres ; l'ensemble de tous les textes du corpus d'apprentissage doit, lors de chaque itération, être de nouveau intégralement parcouru.

Nous proposons une méthode pour réduire la complexité de la phase d'apprentissage d'un ML. Le principe de l'approche que nous préconisons est le suivant : tout d'abord, calculer un ML à partir d'un Lexique (L) le plus grand possible. Puis, après avoir choisi un Sous-Lexique (SL) issu de L, extraire directement le Sous-Modèle de Langage

---

[1] Notamment, en ce qui concerne les ponctuations.

[2] Pour un ensemble logiciel destiné au calcul de ML probabiliste, nous renvoyons au kit du CMU.

(SML), associé au SL, à partir de ML. Nous montrons, par ailleurs, que cette démarche s'applique également pour extraire un sous-modèle d'un modèle plus large sans être forcement exhaustif. La méthode proposée est directe. Elle ne nécessite pas une estimation des paramètres sur le corpus d'apprentissage pour tout nouveau choix d'un sous-lexique particulier.

## 4.3. Extraction de Sous-Modèles de Langage

Nous définissons formellement la fonction $f$ d'extraction de SML.

Soit :

- Le Modèle de Langage initial (ML) et son Lexique associé (L).
- Un Sous-Lexique (SL) dérivé de L.

La fonction $f$ est définie de telle sorte que :

$$SML = f(ML) \qquad \text{(Eq. 4.1)}$$

Cette fonction donne lieu à une formulation évidente dans le cas où le modèle de langage est une interpolation de modèles de langage probabilistes d'ordres différents. La présente étude consiste à montrer qu'une telle démarche est également possible dans le cas d'une stratégie de type *back-off*. Et ce, même si SL n'est pas exactement un sous-ensemble de L.

### 4.3.1. Choix du lexique de référence

Le choix du vocabulaire de référence (dénoté lexique L) pour l'apprentissage des comptes du ML initial est crucial, notamment en ce qui concerne sa couverture. Parmi les décisions délicates figure le problème des mots composés.

Par exemple, dans le cadre de la mise au point d'un modèle de langage pour un SRAP, il faudra utiliser un lexique de référence suffisamment important (en général, le plus grand possible) pour pouvoir disposer de nombreuses unités lexicales différentes dans les SL potentiels.

Une attention particulière doit être portée à la détermination des unités lexicales simples, des mots composés et, de façon plus générale, aux locutions figées. Notons que les SL peuvent ne pas être rigoureusement inclus dans L. En effet, il nous faut examiner à quel point il est possible de décomposer un mot composé de L dans SL (cf. figure 4.1) et, inversement, de composer des mots de L dans SL (cf. figure 4.2).

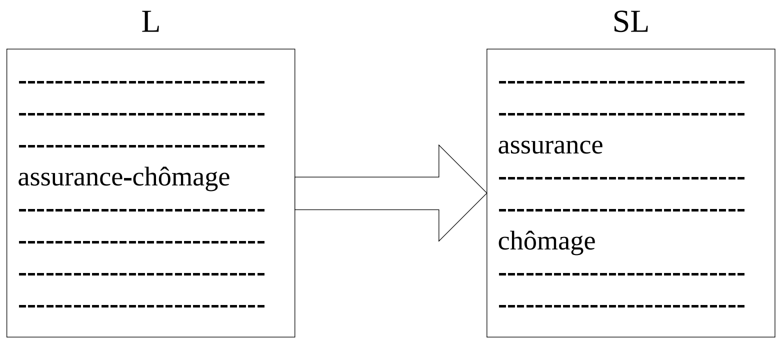


Figure 4.1 : Décomposition d'un mot composé
du lexique L au Sous-Lexique SL

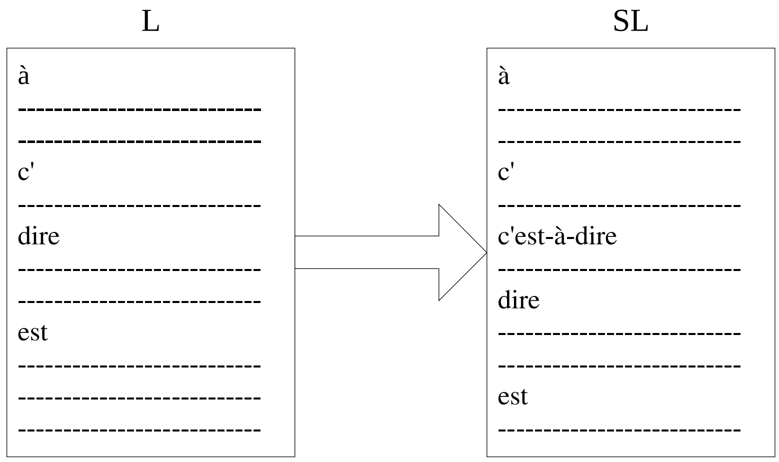


Figure 4.2 : Composition de mots
du Lexique L au Sous-Lexique SL

En fait, lors de l'extraction d'un SML, il est toujours possible de décomposer certaines unités du lexique de référence. Ceci n'est totalement vrai que si la décomposition est systématique et ne nécessite pas une analyse de contexte trop large pour décider, de façon conditionnelle, de la réalisation de telle ou telle décomposition. Mais pour une composition, on ne voit pas comment faire autrement que de ré-estimer les paramètres du modèle sur l'intégralité du corpus d'apprentissage.

Cette caractéristique de l'extraction implique que les unités du lexique de référence doivent être choisies de telle sorte que leur facteur de composition soit le plus élevé possible. De cette façon, le lexique de référence contient tous les mots composés autorisés et il est donc, en principe, inutile de chercher à composer de nouveaux mots.

Par ailleurs, étant donné le domaine d'application retenu, à savoir les SRAP de type machine à dictée, nous avons supposé que les ponctuations ne sont pas dictées pour que le mode d'élocution employé (ici la parole continue) conserve son aspect d'élocution naturelle. Ceci explique pourquoi, lors de la décomposition des mots composés, le tiret n'est pas conservé.

## 4.3.2. Modèle de langage bi-gramme avec *back-off*

Dans cette section, nous définissons l'extraction d'un Sous-Modèle de Langage dans le cadre d'une stratégie *back-off* et d'un bi-gramme de la façon suivante [Katz, 87 ; Jelinek, 90] :

$$P(m_i/m_{i-1}) = \begin{cases} P_{2g}(m_i / m_{i-1}) & si\, C(m_{i-1}, m_i) \geq s \\ dc \times P_{2g}(m_i / m_{i-1}) & si\ s > C(m_{i-1}, m_i) > 0 \\ bg_{1g}(m_{i-1}) \times P_{1g}(m_i) & si\, C(m_{i-1}, m_i) = 0 \end{cases} \qquad \text{(Eq. 4.2)}$$

Avec :

$$P_{2g}(m_i / m_{i-1}) = \frac{C(m_{i-1}, m_i)}{C(m_{i-1})} \qquad \text{(Eq. 4.3)}$$

$$P_{1g}(m_i) = \frac{C(m_i)}{C(m)} \qquad \text{(Eq. 4.4)}$$

Les termes $bg_{1g}(m_i)$ et *dc* représentent les facteurs de repli, *C(m)* le compte du nombre de mots total. Le symbole *INC* est utilisé pour désigner tout mot inconnu, c'est-à-dire tout mot absent du lexique mais aussi l'ensemble des mots de L qui n'appartiennent plus à SL.

### 4.3.2.1. Processus d'extraction des probabilités uni-grammes

Pour tous les mots $m_i$ qui appartiennent à SL et qui sont différents de *INC*, le compte qui leur est affecté dans SML reste celui qu'ils détenaient dans ML.

$$\forall i\ ;\ m_i \in SL,\ m_i \neq INC$$

$$C_{1g,SML}(m_i) = C_{1g,ML}(m_i) \qquad \text{(Eq. 4.5)}$$

En ce qui concerne l'entrée *INC* dans SML, il convient d'ajouter à son compte dans ML la somme des comptes de ML affectés aux mots de L qui n'ont pas été retenus dans SL.

$$\forall i\ ;\ m_i \notin SL$$

$$C_{1g,SML}(INC) = C_{1g,ML}(INC) + \sum_{m_i \notin SL} C_{1g,ML}(m_i) \qquad \text{(Eq. 4.6)}$$

#### 4.3.2.2. Processus d'extraction des probabilités bi-grammes

Pour tous les mots $m_i$ qui appartiennent à L, pour tous les mots $m_j$ qui appartiennent à SL et qui sont différents de *INC*, le compte qui leur est affecté dans SML reste celui qu'ils détenaient dans ML.

$$\forall i,\ \forall j\ ;\ m_i \in L,\ m_j \in SL,\ m_j \neq INC$$

$$C_{2g,SML}(m_i, m_j) = C_{2g,ML}(m_i, m_j) \qquad \text{(Eq. 4.7)}$$

Pour tous les mots $m_i$ qui appartiennent à L et qui sont suivis par le mot *INC*, il convient d'ajouter à leur compte dans SML la somme des comptes de ML de $m_i$ suivi des mots de L qui n'ont pas été retenus dans SL.

$$\forall i\ ;\ m_i \in L$$

$$C_{2g,SML}(m_i, INC) = \sum_{m_j \notin SL} C_{2g,ML}(m_i, m_j) + C_{2g,ML}(m_i, INC) \qquad \text{(Eq. 4.8)}$$

Avec :

$$m_i = INC \text{ dans } C_{2g,SML} \text{ si } m_i \notin SL$$

Le nombre de mots total $C(m)$ reste identique. En ce qui concerne les facteurs de repli $bg_{1g}(m_i)$ et $dc$, ils doivent nécessairement être recalculés. Ces calculs peuvent être réalisés de manière classique [Jelinek, 90].

### 4.3.3. Traitement des mots composés[1]

Soit un mot *mc* composé de *n* mots défini de la façon suivante :

$$mc = mc_1 - mc_2 - \ldots - mc_i - \ldots - mc_n \qquad \text{(Eq. 4.9)}$$

Trois cas doivent être envisagés :

- Un mot *mi* est suivi par un mot composé *mc*.
- Un mot composé *mc* est suivi par un mot *mi*.
- Un mot composé *mc* est suivi par un mot composé *mc'*.

---

[1] Par souci de lisibilité, nous avons choisi de présenter le traitement des mots composés dans une section séparée.

Dans les lignes qui suivent, seul le cas d'un mot $m_i$ suivi par un mot composé *mc* est traité dans le détail. Par ailleurs, les processus d'extraction décrits ci-dessous doivent être appliqués pour chaque *mc* à décomposer.

#### 4.3.3.1. Processus d'extraction des probabilités uni-grammes pour les *mc*

Pour tous les mots de *mc*, $mc_i$ qui appartiennent à SL, il convient d'ajouter à leur somme le compte de *mc* ; et pour tous les mots $mc_i$ de *mc* qui n'appartiennent pas à SL, il convient d'ajouter à la somme du mot *INC* le compte de *mc*.

$$\forall i, m_i \in mc \;;$$

$$C_{1g,SML}(mc_i) += C_{1g,ML}(mc) \qquad \text{(Eq. 4.10)}$$

Avec :

$$mc_i = INC \text{ si } mc_i \notin SL$$

Le nombre total de mots est alors :

$$C_{1g,SML}(m) += C_{1g,ML}(mc) \times (n-1) \qquad \text{(Eq. 4.11)}$$

**Note**

> Pour un mot composé *mc* donné, il faut ajouter au nombre total de mots le compte de *mc* multiplié par le nombre *n* de mots qui le composent (soit la décomposition du mot composé) et soustraire une fois le compte de *mc* (soit le mot composé dans sa totalité). Ce qui correspond à l'équation 4.11.

#### 4.3.3.2. Processus d'extraction des probabilités bi-grammes pour les *mc*

Pour tous les mots $m_i$ suivis de $mc_1$ appartenant à SL, il convient d'ajouter à la somme dans SML de $m_i$ suivi de $mc_1$ le compte dans ML de $m_i$ suivi de *mc* ; et pour tous les mots $m_i$ suivis de *mc* n'appartenant pas à SL, il convient d'ajouter à la somme dans SML de $m_i$ suivi de *INC* le compte dans ML de $m_i$ suivi de *mc*.

Un raisonnement similaire doit être appliqué pour la suite des mots composés $mc_j$ suivis de $mc_{j+1}$ pour *j* variant de *1* à *n-1*.

Pour $mc_1$ et pour tous les mots $m_i$ précédant *mc* :

$$C_{2g,SML}(m_i, mc_1) += C_{2g,ML}(m_i, mc) \qquad \text{(Eq. 4.12)}$$

Avec :

$$mc_1 = INC\ si\ mc_1 \notin SL$$

Pour tous les $mc_j$ tels que $1 < j < n$ :

$$C_{2g,SML}(mc_j, mc_{j+1}) += C_{2g,ML}(m_i, mc) \qquad \text{(Eq. 4.13)}$$

Avec :

$$mc_j = INC\ si\ mc_j \notin SL$$

$$mc_{j+1} = INC\ si\ mc_{j+1} \notin SL$$

À partir des équations 4.12 et 4.13, les formulations de l'extraction des mots composés pour les comptes *mc* sachant $m_i$ et des *mc'* sachant *mc* sont triviales. En effet, la première formulation se déduit de l'équation 4.12 par symétrie ; dans le second cas, la décomposition peut être effectuée en deux parties, un mot composé après l'autre. Quant à l'équation 4.13, elle ne doit être appliquée qu'une fois par mot composé (c'est-à-dire dans le contexte gauche ou droit et pas pour les deux contextes).

**Remarque**

Il se peut qu'un élément $mc_i$ d'un mot composé soit lui-même un mot composé et/ou que $mci \notin$ L. Par exemple, pour le mot composé *au-fur-et-à-mesure*, le mot *fur* à fort peu de chance d'appartenir à L.

## 4.4. Expériences et résultats

Cette méthode est appliquée sur un lexique L de 64 000 mots et son modèle de langage ML bi-gramme de type *back-off*[1]. À partir des SL correspondant aux 5 000, 10 000, 15 000, ..., 55 000 et 60 000 mots les plus fréquents, nous avons extrait les SML associés. Le processus d'extraction utilise directement les valeurs probabilistes du modèle de langage de référence. Les équations d'extractions données précédemment en fonction des comptes sont adaptées aux probabilités.

À titre indicatif, nous avons calculé un ensemble de mesures usuelles pour ces différents lexiques et modèles de langage sur un corpus d'environ 200 000 phrases (soit près de 5,5 millions de mots[1]) issues du *Monde Diplomatique*[2]. Dans ces expériences, les ponctuations ne sont pas prises en compte.

---

[1] Ces données font partie des ressources disponibles pour les participants de la campagne de test B1 de l'Aupelf-Uref.

Les mesures de mots inconnus et de couvertures sont définies de la façon suivante :

- Le nombre de mots inconnus (Nb *INC*) correspond au nombre total de mots inconnus.
- Le nombre de mots inconnus dynamique (Nb *INC* dyn.) correspond au nombre de mots inconnus sans répétition. C'est-à-dire qu'un mot inconnu n'est compté comme tel que la première fois où il est rencontré dans le corpus.

Le calcul de la couverture et de la perplexité est standard [Jelinek, 90] et nous avons choisi d'ignorer les ponctuations. Par ailleurs, dans le cas de la perplexité, nous prenons en compte les débuts et fins de phrases mais seules les fins de phrases sont prédites. Le nombre total de mots utilisés pour le calcul de la perplexité est donc le nombre de mots ajoutés au nombre de phrases du corpus.

Le tableau 4.1 présente les résultats obtenus :

| Lexique | Nb INC | Nb INC dyn. | Couv. stat. | Couv. dyn. | Perplexité |
|---|---|---|---|---|---|
| 5k | 835 138 | 98 150 | 84 % | 98 % | 86 |
| 10k | 533 354 | 93 203 | 89 % | 98 % | 131 |
| 15k | 392 752 | 88 354 | 92 % | 98 % | 162 |
| 20k | 310 285 | 83 616 | 94 % | 98 % | 184 |
| 25k | 256 089 | 79 063 | 95 % | 98 % | 201 |
| 30k | 218 080 | 74 786 | 95 % | 98 % | 214 |
| 35k | 190 291 | 70 750 | 96 % | 98 % | 225 |
| 40k | 167 795 | 66 912 | 96 % | 98 % | 234 |
| 45k | 150 035 | 63 345 | 97 % | 98 % | 242 |
| 50k | 136 312 | 60 122 | 97 % | 98 % | 248 |
| 55k | 124 887 | 57 154 | 97 % | 98 % | 253 |
| 60k | 115 315 | 54 337 | 97 % | 98 % | 258 |
| 64k | 108 112 | 52 432 | 97 % | 99 % | 262 |

Tableau 4.1 : Mesures usuelles sur différents lexiques et ML

---

[1] Ce nombre de mots ne comprend pas les ponctuations (nous avons choisi de cadrer notre étude sur les SRAP continus).

[2] Le corpus expérimental utilisé dans cet article provient du CD-ROM du *Monde Diplomatique*. Ces textes ont été traités et mis à notre disposition par Gilles Adda du LIMSI/CNRS.

La figure 4.3 présente quant à elle les mesures de couverture statique, de couverture dynamique et de perplexité sous la forme d'une courbe :

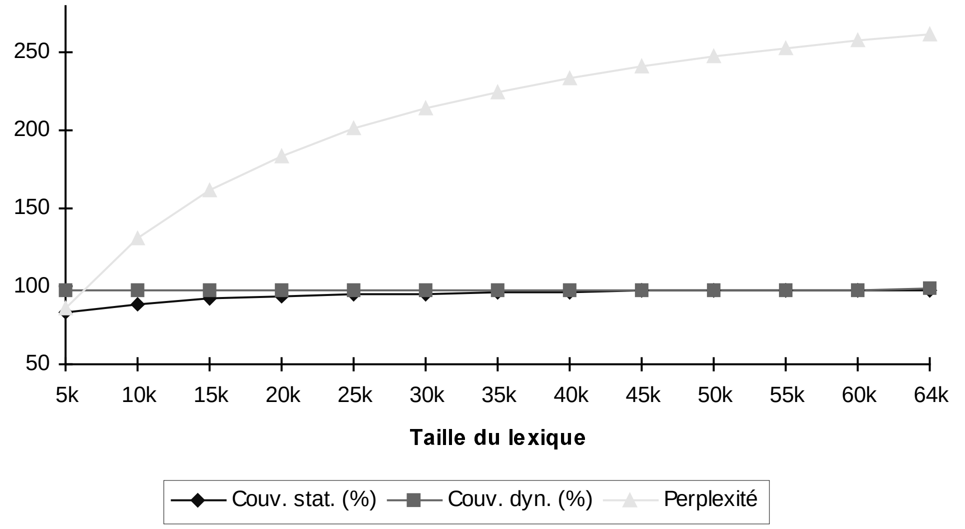


Figure 4.3 : Mesures usuelles sur différents lexiques et ML

Les valeurs obtenues pour la perplexité illustrent bien l'influence de la taille du vocabulaire sur ce critère. En effet, plus la taille du vocabulaire diminue, plus le facteur de branchement moyen décroît. Cette tendance est amplifiée par l'intégration des mots inconnus dans les calculs. Toutefois, en prenant par exemple les lexiques 5k et 20k, on peut noter que la perplexité sur SML est environ deux fois plus petite que celle sur ML pour un rapport de taille quatre sur le vocabulaire. Mais il faut se garder de toute conclusion hâtive car s'il avait été possible de réajuster les facteurs de repli, le rapport entre les deux perplexités aurait été autre.

Par ailleurs, dans la mesure où, dans nos expériences, le processus d'extraction utilise directement les probabilités (et non les comptes), les facteurs de repli ne peuvent pas être recalculés. Dans ce cas, il est indispensable de valider l'intégrité du modèle de langage SLM, en vérifiant pour tous les mots $m_i$ tels que $m_i \in SL$ (y compris *INC*) l'équation suivante :

$$\forall m_i\,;\ \sum_{j;P(m_i,m_j)>0} P(m_j / m_i) \ + bg_{1g}(m_i) \times \sum_{j;P(m_i,m_j)=0} P(m_j) = 1 \qquad \text{(Eq. 4.14)}$$

Cette procédure de test, qui a été appliquée sur les modèles ML et SML, nous a permis de valider expérimentalement l'intégrité de l'approche proposée.

## 4.5. Conclusion

Dans cette quatrième partie, nous avons proposé une méthode qui permet d'extraire directement un SML d'un premier ML calculé précédemment.

Tout d'abord, nous avons présenté la problématique de l'extraction des SML, en insistant sur l'importance du choix des unités lexicales pour la première collecte, notamment en ce qui concerne les mots composés et, de façon plus générale, les locutions figées. Les caractéristiques de l'extraction de SML impliquent que les unités du lexique de référence soient choisies de telle sorte que leur facteur de composition soit le plus élevé possible. Puis, nous avons définis formellement l'extraction d'un SML dans le cadre d'un ML bi-gramme ; une attention particulière a été portée sur le traitement des mots composés. Enfin, nous avons appliqué cette méthode à un lexique de 64 000 mots et à son ML associé sur une suite de Sous-Lexiques. À titre indicatif, le calcul d'un ensemble de mesures usuelles pour ces différents lexiques et modèles de langage a été présenté.

Le principal intérêt de l'approche proposée réside dans une économie de moyens et de temps lors de la phase d'apprentissage de modèles de langage. À partir d'un modèle de langage générique, le calcul de sous-modèles de langage est alors simple et rapide.

Par ailleurs, cette approche ouvre également d'autres perspectives. Ainsi, elle peut s'avérer utile dans le cadre des SRAP qui autorisent l'enrichissement du vocabulaire actif par un lexique passif [Baker, 89]. Dans ce cas, à partir d'un modèle de langage pré-estimé, le calcul des paramètres des nouveaux modèles de langage des lexiques actifs et variables du SRAP est élémentaire. L'évolution du composant acoustique peut être vue comme un parcours dans une famille de lexiques emboîtés auxquels la fonction d'extraction permet d'associer un ML adéquat.

# Conclusions et perspectives

Les travaux présentés dans cette thèse concernent la Reconnaissance Automatique de la Parole (RAP).

Malgré la maîtrise des fondements théoriques et pratiques, force est de constater que la réalisation d'un Système de Reconnaissance Automatique de la Parole Continue (SRAPC) reste un problème difficile. En effet, le niveau de performance d'un système de reconnaissance est également lié à la qualité des données utilisées et à la pertinence des choix effectués *a priori*. C'est pourquoi les meilleurs SRAPC mettent en œuvre des ressources substantielles, tant humaines que matérielles, afin de disposer de l'ensemble des technologies, connaissances et savoir faire pluridisciplinaires nécessaires à leur réalisation. Il est à noter que le coût global d'un tel système devient rapidement prohibitif.

Nous avons proposé l'étude, le développement et l'intégration des éléments constitutifs d'un SRAPC. D'une manière générale, la problématique traitée recouvre l'ensemble des problèmes rencontrés lors de l'automatisation des processus de construction d'un moteur de RAP.

Dans le cadre de la réalisation de ces différents éléments, plusieurs travaux et résultats originaux ont été présentés :

- Un module de transcription graphèmes-phonèmes pour la RAP fondé sur une approche de transcription automatique à base de règles et sur un module d'étiquetage a été réalisé. Il permet notamment de traiter les sigles, les noms propres, génère les liaisons et des variantes de prononciations. Deux approches de phonétisation ont été proposées : l'approche hors contexte qui consiste à appliquer systématiquement les règles de génération de variantes ; l'approche en contexte qui utilise les différentes phonétisations obtenues par un système de transcription graphèmes-phonèmes pour la synthèse sur un corpus de phrases.

  Une expérience de décodage exploitant un lexique de prononciation phonétisé avec ces deux approches a montré que l'approche de phonétisation en contexte permet de réduire la complexité du graphe de recherche tout en conservant le même taux de reconnaissance.

- Une étude portant sur de l'impact de l'alignement des données acoustiques et phonétiques sur l'initialisation des modèles probabilistes acoustiques a montré qu'une initialisation globale moyenne des HMM est plus efficace qu'une initialisation basée sur un alignement acoustico-phonétique automatique non contrôlé.

- Une stratégie de reconnaissance multi-passes, où la première passe de décodage produit des listes homophones sur lesquelles une seconde passe de filtrage acoustico-linguistique permet de déterminer l'hypothèse optimale, a été mise en place.

  Malgré une réduction du niveau de connaissance utilisé dans la première passe de décodage homophone, la seconde passe permet de retrouver l'information perdue. Ce résultat valide le module de filtrage acoustico-phonétique proposé et montre l'importance des connaissances linguistiques dans un processus de reconnaissance.

- Une stratégie de reconnaissance A* qui permet d'inclure directement des connaissances diverses dans le processus de décodage a été mise en place. Elle est fondée sur le pré-calcul des différentes fonctions de coût acoustique et sur une fonction d'estimation qui utilise l'algorithme de Viterbi back et un ML bi-gramme phonétique.

  Avec un lexique de 20 000 mots et un ML tri-gramme, nous sommes confrontés à un problème de convergence de l'algorithme sur certaines phrases du corpus de test. Ce problème, lié à la fonction d'estimation *h* et à des phases de calculs inutilement répétées, nécessite un travail conséquent d'optimisation. Cependant, l'augmentation du taux de reconnaissance obtenue grâce à un ML tri-gramme phonétique sur le DAP atteste l'intérêt d'une stratégie de décodage de ce type.

- À partir d'un Modèle de Langage (ML) calculé au préalable, un module d'extraction direct de Sous-Modèles de Langage (SML) a été réalisé.

  Le principal apport de l'extraction de SML est une économie de moyens et de temps lors de la phase d'apprentissage de modèles de langage. À partir d'un ML générique, la calcul de SML est alors simple et rapide.

Que ce soit pour le DAP, le SRAPC de référence ou les stratégies de reconnaissance que nous avons mis en place, les résultats obtenus valident notre approche de construction automatique d'un moteur de RAP. L'ensemble des éléments exposés dans ce document permettent de construire un SRAP automatiquement et à moindre coût.

Malgré ces résultats, nous sommes conscients que ce travail présente certaines limites. La principale limite concerne le manque de ressources. Nous pensons notamment à l'apprentissage des modèles acoustiques qui necessite une augmentation de la taille de la base de données. Le passage de BREF-80 à BREF est incontournable. Il devrait apporter une amélioration significative des taux de reconnaissance [Adda, 97a]. Cela permettrait notamment de mettre en place des modèles dépendants du contexte et des multi-modèles hommes/femmes. Par ailleurs, une optimisation des différents paramètres du système tels que les poids (acoustiques et linguistiques) et le choix des mots du lexique (locution, mot composé, mot outil, etc.) devrait être entreprit.

Mais au-delà de l'accroissement de la taille des bases de données d'entraînement, du nombre d'unités de parole élémentaires modélisées, du nombre de paramètres décrivant ces modèles et du réglage des différents paramètres d'un SRAP, nous avons recensé certaines perspectives.

Tout d'abord, nous proposons de réfléchir sur l'approche homophone qui ouvre de nombreuses questions. Nous pensons en particulier à une étude du niveau de "bruit" engendré par les homophones dans le processus de décodage. Une évaluation de ce niveau de "bruit" pourrait être réalisée en développant les listes homophones issues de la première passe avec un lexique homophone étendu. Dans ce cas, la première passe de décodage reste strictement identique ; nous proposons de rajouter des nouveaux mots homophones dans les listes homophones avant la seconde passe de filtrage acoustico-linguistique. Cette expérience ouvrirait également des perspectives dans le cadre de la réduction du taux de mots hors vocabulaire et de la correction d'erreurs de reconnaissance dues, en très large partie, aux mots homophones [Adda, 97a].

Nous nous interrogeons également sur les liens qui pourraient être mis en place entre les deux stratégies de décodage proposées.

Dans ce cadre, les perspectives de recherche que nous envisageons visent à intégrer le filtre acoustico-phonétique de l'approche homophone avec le module de décodage de l'approche A*. Cette intégration peut être aisément réalisée *via* l'algorithme A* qui sert de base à ces deux modules. Elle permettrait d'ajouter des contraintes sous forme de règles dans le processus de décodage. À l'instar du module d'étiquetage automatique ECSta qui utilise des règles d'agrammaticalité, nous pourrions mettre en place des règles qui pénaliseraient certaines hypothèses de décodage. Nous pensons, par exemple, à des règles sur des suites catégories syntaxiques ou de liaisons irréalistes. D'une manière générale, ce type d'approche permettrait d'intégrer tout type de connaissance au cœur de la stratégie de décodage.

# Annexes

# Annexe A

# Table des abréviations

| | |
|---|---|
| **API** | **A**lphabet **P**honétique **I**nternational |
| **ARC** | **A**ctions de **R**echerches **C**oncertées |
| **ARPA** | **A**dvanced **R**esearch **P**rojects **A**gency |
| **ASSP** | **A**coustics, **S**peech and **S**ignal **P**rocessing |
| **AT&T** | American Telephone and Telegraph |
| **AUPELF** | **A**ssociation des **U**niversités **P**artiellement ou **E**ntièrement de **L**angue **F**rançaise |
| **BBN** | **B**olt, **B**eranek and **N**ewman |
| **BDLEX** | **B**ase de **D**onnée **LEX**icale |
| **CMU** | **C**arnegie Mellon University |
| **CNET** | **C**entre **N**ational d'**É**tudes des **T**élécommunications |
| **CRIM** | **C**entre de **R**echerche en **I**nformatique de **M**ontréal |
| **CUED** | **C**ambridge **U**niversity **E**ngineering **D**epartement |
| **DAP** | **D**écodage **A**coustico-**P**honétique |
| **DARPA** | **D**efense **A**dvanced **R**esearch **P**rojects **A**gency |
| **DTW** | **D**ynamic **T**ime **W**arping |
| **ENST** | **É**cole **N**ationale **S**upérieure des **T**élécommunications |
| **EUROSPEECH** | **EURO**pean conference on **SPEECH** communication and technology |
| **ESPRIT** | **E**uropean **S**trategic **P**rogramme for **R**esearch and development in **I**nformation **T**echnology |
| **HMM** | **H**idden **M**arkov **M**odel |
| **HTK** | **H**idden markov model **T**ool**K**it |
| **IA** | **I**ntelligence **A**rtificielle |
| **IBM** | **I**nternational **B**usiness **M**achines |
| **ICASSP** | **I**nternational **C**onference on **A**coustics, **S**peech and **S**ignal **P**rocessing |
| **ICSLP** | **I**nternational **C**onference on **S**peech and **L**anguage **P**rocessing |
| **IEEE** | **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers |
| **ILOR** | **I**nformatique, **L**inguistique et corpus **OR**aux |

| | |
|---|---|
| **IRIT** | **I**nstitut de **R**echerche en **I**nformatique de **T**oulouse |
| **JEP** | **J**ournées d'**É**tudes sur la **P**arole |
| **JST** | **J**ournées **S**cientifiques et **T**echniques |
| **LIA** | **L**aboratoire **I**nformatique d'**A**vignon |
| **LIMSI** | **L**aboratoire d'**I**nformatique pour la **M**écanique et les **S**ciences de l'**I**ngénieur |
| **LPC** | **L**inear **P**redictive **C**oefficient |
| **MFCC** | **M**el **F**requency **C**epstral **C**oefficient |
| **MHV** | **M**ots **H**ors **V**ocabulaire |
| **MIT** | **M**assachussets **I**nstitute of **T**echnology |
| **ML** | **M**odèle de **L**angage |
| **MLD** | **M**ulti-**L**evel **D**ecoding |
| **MLE** | **M**aximum **L**ikelihood **E**stimation |
| **MMI** | **M**aximum **M**utual **I**nformation |
| **MPJ** | **M**odèle de **P**ropagation de **J**eton |
| **OOV** | **O**ut **O**f **V**ocabulary |
| **RAP** | **R**econnaissance **A**utomatique de la **P**arole |
| **RAPC** | **R**econnaissance **A**utomatique de la **P**arole **C**ontinue |
| **SML** | **S**ous-**M**odèle de **L**angage |
| **SRAP** | **S**ystème de **R**econnaissance **A**utomatique de la **P**arole |
| **SRAPC** | **S**ystème de **R**econnaissance **A**utomatique de la **P**arole **C**ontinue |
| **TAL** | **T**raitement **A**utomatique du **L**angage |
| **TALEO** | **T**raitement **A**utomatique du **L**angage **É**crit et **P**arlée |
| **TALN** | **T**raitement **A**utomatique du **L**angage **N**aturel |
| **TAP** | **T**raitement **A**utomatique de la **P**arole |
| **UREF** | **U**niversité des **R**éseaux d'**E**xpression **F**rançaise |

# Annexe B

# Alphabet Phonétique International

L'Alphabet Phonétique International (API) est la notation adoptée pour désigner les sons. Dans cette annexe, nous présentons les codes API, suivis des équivalences informatiques adoptées au LIA et d'exemples associés.

## Les voyelles

| Codes API | Codes LIA | Exemples |
|---|---|---|
| i | ii | sour**i**s |
| u | ou | p**ou**le |
| y | uu | ill**u**sion |
| e | ei | f**é**e |
| ɛ | ai | syst**è**me |
| o | au | r**o**se |
| ɔ | oo | **o**céan |
| ø | eu | p**eu** |
| œ | oe | p**eu**r |
| a | aa | ciném**a** |
| ɑ̃ | an | ch**an**ter |
| ɔ̃ | on | r**on**fler |
| ɛ̃ | in | jard**in** |
| œ̃ | un | br**un** |

## Les semi-voyelles

| Codes API | Codes LIA | Exemples |
|---|---|---|
| j | yy | l**i**eu |
| w | uy | **ou**i |

## Les consonnes

| Codes API | Codes LIA | Exemples |
|---|---|---|
| p | pp | **p**arole |
| t | tt | **t**able |
| k | kk | **c**adeau |
| b | bb | **b**anane |
| d | dd | **d**ate |
| g | gg | **g**âteau |
| f | ff | **f**aim |
| s | ss | **s**el |
| ʃ | ch | **ch**ocolat |
| v | vv | **v**acance |
| z | zz | mai**s**on |
| ʒ | jj | an**ge** |
| m | mm | **m**iel |
| n | nn | **n**appe |
| ɲ | gn | si**gn**e |
| l | ll | **l**ire |
| ʀ | rr | **r**ouler |

# Annexe C

# Classes syntaxiques

Cette annexe présente les classes syntaxiques utilisées :

| Codes | Signification |
|---|---|
| ADV | adverbe |
| ADVNE | ne |
| ADVPAS | pas |
| AFP | adjectif féminin pluriel |
| AFS | adjectif féminin singulier |
| AINDFP | adjectif indéfini féminin pluriel |
| AINDFS | adjectif indéfini féminin singulier |
| AINDMP | adjectif indéfini masculin pluriel |
| AINDMS | adjectif indéfini masculin singulier |
| AMP | adjectif masculin pluriel |
| AMS | adjectif masculin singulier |
| CHIF | chiffre ou nombre |
| COCO | conjonction de coordination |
| COSUB | conjonction de subordination |
| DETFP | déterminant féminin pluriel |
| DETFS | déterminant féminin singulier |
| DETMP | déterminant masculin pluriel |
| DETMS | déterminant masculin singulier |
| DINTFP | déterminant interrogatif féminin pluriel |
| DINTFS | déterminant interrogatif féminin singulier |
| DINTMP | déterminant interrogatif masculin pluriel |
| DINTMS | déterminant interrogatif masculin singulier |
| MOTINC | mots inconnus |
| NFP | nom féminin pluriel |
| NFS | nom féminin singulier |
| NMP | nom masculin pluriel |
| NMS | nom masculin singulier |
| PDEMFP | pronom démonstratif féminin pluriel |
| PDEMFS | pronom démonstratif féminin singulier |

| Codes | Signification |
|---|---|
| PDEMMP | pronom démonstratif masculin pluriel |
| PDEMMS | pronom démonstratif masculin singulier |
| PINDFP | pronom indéfini féminin pluriel |
| PINDFS | pronom indéfini féminin singulier |
| PINDMP | pronom indéfini masculin pluriel |
| PINDMS | pronom indéfini masculin singulier |
| PINTFP | pronom interrogatif féminin pluriel |
| PINTFS | pronom interrogatif féminin singulier |
| PINTMP | pronom interrogatif masculin pluriel |
| PINTMS | pronom interrogatif masculin singulier |
| PPER1P | pronom personnel 1$^{ère}$ personne du pluriel |
| PPER1S | pronom personnel 1$^{ère}$ personne du singulier |
| PPER2P | pronom personnel 2$^{ème}$ personne du pluriel |
| PPER2S | pronom personnel 2$^{ème}$ personne du singulier |
| PPER3FP | pronom personnel 3$^{ème}$ personne du féminin pluriel |
| PPER3FS | pronom personnel 3$^{ème}$ personne du féminin singulier |
| PPER3MP | pronom personnel 3$^{ème}$ personne du masculin pluriel |
| PPER3MS | pronom personnel 3$^{ème}$ personne du masculin singulier |
| PPOBJFP | pronom personnel objet féminin pluriel |
| PPOBJFS | pronom personnel objet féminin singulier |
| PPOBJMP | pronom personnel objet masculin pluriel |
| PPOBJMS | pronom personnel objet masculin singulier |
| PREFFP | pronom réfléchi féminin pluriel |
| PREFFS | pronom réfléchi féminin singulier |
| PREFMP | pronom réfléchi masculin pluriel |
| PREFMS | pronom réfléchi masculin singulier |
| PRELFP | pronom relatif féminin pluriel |
| PRELFS | pronom relatif féminin singulier |
| PRELMP | pronom relatif masculin pluriel |
| PRELMS | pronom relatif masculin singulier |
| PREP | proposition |
| PREPADE | à de |
| PREPAU | au |
| PREPAUX | aux |
| PREPDES | des |
| PREPDU | du |
| V1P | verbe 1$^{ère}$ personne du pluriel |

| Codes | Signification |
|---|---|
| V1S | verbe 1$^{ère}$ personne du singulier |
| V2P | verbe 2$^{ème}$ personne du pluriel |
| V2S | verbe 2$^{ème}$ personne du singulier |
| V3P | verbe 3$^{ème}$ personne du pluriel |
| V3S | verbe 3$^{ème}$ personne du singulier |
| VA1P | auxiliaire avoir 1$^{ère}$ personne du pluriel |
| VA1S | auxiliaire avoir 1$^{ère}$ personne du singulier |
| VA2P | auxiliaire avoir 2$^{ème}$ personne du pluriel |
| VA2S | auxiliaire avoir 2$^{ème}$ personne du singulier |
| VA3P | auxiliaire avoir 3$^{ème}$ personne du pluriel |
| VA3S | auxiliaire avoir 3$^{ème}$ personne du singulier |
| VAINF | auxiliaire avoir infinitif |
| VE1P | auxiliaire être 1$^{ère}$ personne du pluriel |
| VE1S | auxiliaire être 1$^{ère}$ personne du singulier |
| VE2P | auxiliaire être 2$^{ème}$ personne du pluriel |
| VE2S | auxiliaire être 2$^{ème}$ personne du singulier |
| VE3P | auxiliaire être 3$^{ème}$ personne du pluriel |
| VE3S | auxiliaire être 3$^{ème}$ personne du singulier |
| VEINF | auxiliaire être infinitif |
| VINF | verbe à l'infinitif |
| VPPFP | participe passé féminin pluriel |
| VPPFS | participe passé féminin singulier |
| VPPMP | participe passé masculin pluriel |
| VPPMS | participe passé masculin singulier |
| VPPRE | participe présent |
| XFAMIL | nom propre : nom de famille |
| XPAYFP | nom propre : nom de pays féminin pluriel |
| XPAYFS | nom propre : nom de pays féminin singulier |
| XPAYMP | nom propre : nom de pays masculin pluriel |
| XPAYMS | nom propre : nom de pays masculin singulier |
| XPREF | nom propre : prénom féminin |
| XPREM | nom propre : prénom masculin |
| XSOC | nom propre : société |
| XVILLE | nom propre : ville |
| YPFAI | ponctuation faible , : ; ( ) |
| YPFOR | ponctuation forte . ? ! |
| ZTRM | début ou fin de phrase |

# Annexe D

# Résultats de reconnaissance du DAP

Cette annexe présente le détail des résultats de reconnaissance du DAP, obtenus sur les corpus de test et d'apprentissage en fonction du type d'initialisation de l'apprentissage, du nombre de gaussiennes et du poids du modèle de langage.

## En fonction d'une Initialisation à partir de l'Alignement Acoustique et Phonétique (IAAP)

### Sur le corpus de test

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 1 | 0 | 3 032 | 11 009 | 3 022 | 53,60 % | 43,61 % |
| 1 | 1 | 3 618 | 10 154 | 2 239 | 54,49 % | 47,09 % |
| 1 | 2 | 4 070 | 9 629 | 1 762 | 54,73 % | 48,90 % |
| 1 | 4 | 4 736 | 8 958 | 1 248 | 54,74 % | 50,62 % |
| 1 | 6 | 5 427 | 8 441 | 983 | 54,17 % | 50,92 % |
| 2 | 0 | 2 477 | 10 475 | 3 538 | 57,20 % | 45,50 % |
| 2 | 1 | 2 947 | 9 809 | 2 448 | 57,84 % | 49,75 % |
| 2 | 2 | 3 367 | 9 344 | 1 917 | 57,99 % | 51,66 % |
| 2 | 4 | 4 042 | 8 641 | 1 324 | 58,09 % | 53,71 % |
| 2 | 6 | 4 667 | 8 174 | 1 031 | 57,56 % | 54,16 % |
| 4 | 0 | 2 079 | 9 282 | 3 318 | 62,45 % | 51,49 % |
| 4 | 1 | 2 523 | 8 660 | 2 315 | 63,04 % | 55,39 % |
| 4 | 2 | 2 825 | 8 269 | 1 751 | 63,34 % | 57,55 % |
| 4 | 4 | 3 394 | 7 754 | 1 181 | 63,16 % | 59,26 % |
| 4 | 6 | 3 900 | 7 425 | 920 | 62,57 % | 59,53 % |
| 8 | 0 | 1 989 | 8 567 | 2 988 | 65,11 % | 55,24 % |
| 8 | 1 | 2 393 | 7 948 | 1 997 | 65,83 % | 59,23 % |
| 8 | 2 | 2 678 | 7 535 | 1 537 | 66,25 % | 61,17 % |
| 8 | 4 | 3 197 | 6 997 | 1 035 | 66,31 % | 62,89 % |
| 8 | 6 | 3 702 | 6 712 | 799 | 65,58 % | 62,94% |

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 12 | 0 | 1 855 | 8 229 | 2 892 | 66,67 % | 57,12 % |
| 12 | 1 | 2 246 | 7 702 | 1 913 | 67,12 % | 60,80 % |
| 12 | 2 | 2 569 | 7 330 | 1 453 | 67,29 % | 62,48 % |
| 12 | 4 | 3 094 | 6 822 | 992 | 67,23 % | 63,95 % |
| 12 | 6 | 3 613 | 6 494 | 766 | 66,60 % | 64,07 % |
| 16 | 0 | 1 800 | 7 943 | 2 888 | 67,80 % | 58,26 % |
| 16 | 1 | 2 201 | 7 462 | 1 905 | 68,07 % | 61,77 % |
| 16 | 2 | 2 541 | 7 095 | 1 421 | 68,15 % | 63,46 % |
| 16 | 4 | 3 102 | 6 607 | 949 | 67,91 % | 64,78 % |
| 16 | 6 | 3 615 | 6 264 | 752 | 67,32 % | 64,83 % |
| 20 | 0 | 1 797 | 7 757 | 2 890 | 68,43 % | 58,88 % |
| 20 | 1 | 2 179 | 7 311 | 1 899 | 68,64 % | 62,36 % |
| 20 | 2 | 2 515 | 6 919 | 1 449 | 68,82 % | 64,03 % |
| 20 | 4 | 3 049 | 6 461 | 997 | 68,57 % | 65,80 % |
| 20 | 6 | 3 548 | 6 167 | 737 | 67,89 % | 65,46 % |
| 24 | 0 | 1 639 | 6 749 | 2 583 | 72,28 % | 63,74 % |
| 24 | 1 | 1 974 | 6 295 | 1 680 | 72,67 % | 67,12 % |
| 24 | 2 | 2 266 | 5 939 | 1 274 | 72,88 % | 68,67 % |
| 24 | 4 | 2 790 | 5 526 | 900 | 72,52 % | 69,54 % |
| 24 | 6 | 3 266 | 5 312 | 704 | 71,65 % | 69,32 % |

### Sur le corpus de d'apprentissage

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 24 | 0 | 15 679 | 70 090 | 30 878 | 75,72 % | 66,97 % |
| 24 | 1 | 18 895 | 65 345 | 20 476 | 76,15 % | 70,35 % |
| 24 | 2 | 21 888 | 61 778 | 15 566 | 76,31 % | 71,91 % |
| 24 | 4 | 27 148 | 57 817 | 10 828 | 75,94 % | 72,88 % |
| 24 | 6 | 32 400 | 55 798 | 8 503 | 75,03 % | 72,62 % |

## En fonction d'une Initialisation Globale Moyenne (IGB)

### Sur le corpus de test

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 1 | 0 | 2 873 | 10 221 | 2 974 | 56,73 % | 46,90 % |
| 1 | 1 | 3 391 | 9 461 | 2 176 | 57,53 % | 50,34 % |
| 1 | 2 | 3 737 | 8 983 | 1 757 | 57,87 % | 52,16 % |
| 1 | 4 | 4 421 | 8 351 | 1 302 | 57,79 % | 53,49 % |
| 1 | 6 | 5 037 | 7 934 | 1 065 | 57,13 % | 53,61 % |
| 2 | 0 | 2 223 | 9 477 | 3 508 | 61,33 % | 49,74 % |
| 2 | 1 | 2 652 | 8 815 | 2 484 | 62,10 % | 53,89 % |
| 2 | 2 | 2 988 | 8 374 | 1 939 | 62,45 % | 56,04 % |
| 2 | 4 | 3 582 | 7 818 | 1 365 | 62,33 % | 57,81 % |
| 2 | 6 | 4 181 | 7 499 | 1 112 | 61,40 % | 57,72 % |
| 4 | 0 | 1 962 | 8 251 | 3 142 | 66,25 % | 55,86 % |
| 4 | 1 | 2 322 | 7 675 | 2 109 | 66,96 % | 59,99 % |
| 4 | 2 | 2 621 | 7 298 | 1 651 | 67,22 % | 61,76 % |
| 4 | 4 | 3 138 | 6 827 | 1 153 | 67,07 % | 63,26 % |
| 4 | 6 | 3 631 | 6 576 | 910 | 66,27 % | 63,26 % |
| 8 | 0 | 1 810 | 7 530 | 2 749 | 69,13 % | 60,05 % |
| 8 | 1 | 2 167 | 6 973 | 1 910 | 69,79 % | 63,48 % |
| 8 | 2 | 2 445 | 6 619 | 1 444 | 70,05 % | 65,27 % |
| 8 | 4 | 2 898 | 6 197 | 997 | 69,94 % | 66,65 % |
| 8 | 6 | 3 394 | 5 961 | 799 | 69,08 % | 66,44 % |
| 12 | 0 | 1 734 | 7 116 | 2 689 | 70,75 % | 61,87 % |
| 12 | 1 | 2 080 | 6 657 | 1 823 | 71,13 % | 65,10 % |
| 12 | 2 | 2 362 | 6 292 | 1 384 | 71,40 % | 66,83 % |
| 12 | 4 | 2 863 | 5 834 | 980 | 71,26 % | 68,02 % |
| 12 | 6 | 3 312 | 5 610 | 775 | 70,51 % | 67,95 % |

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 16 | 0 | 1 703 | 6 822 | 2 629 | 71,83 % | 63,14 % |
| 16 | 1 | 2 038 | 6 381 | 1 708 | 72,18 % | 66,53 % |
| 16 | 2 | 2 313 | 6 030 | 1 310 | 72,43 % | 68,10 % |
| 16 | 4 | 2 793 | 5 660 | 942 | 72,06 % | 68,95 % |
| 16 | 6 | 3 262 | 5 415 | 743 | 71,32 % | 68,87 % |
| 20 | 0 | 1 650 | 6 600 | 2 562 | 72,74 % | 64,27 % |
| 20 | 1 | 1 966 | 6 123 | 1 695 | 73,27 % | 67,67 % |
| 20 | 2 | 2 245 | 5 856 | 1 284 | 73,23 % | 68,98 % |
| 20 | 4 | 2 718 | 5 489 | 893 | 72,88 % | 69,93 % |
| 20 | 6 | 3 159 | 5 280 | 688 | 72,11 % | 69,84 % |
| 24 | 0 | 1 643 | 6 488 | 2 491 | 73,13 % | 64,90 % |
| 24 | 1 | 1 955 | 6 062 | 1 649 | 73,51 % | 68,06 % |
| 24 | 2 | 2 207 | 5 742 | 1 218 | 73,73 % | 69,70 % |
| 24 | 4 | 2 670 | 5 363 | 860 | 73,45 % | 70,61 % |
| 24 | 6 | 3 155 | 5 147 | 683 | 72,56 % | 70,31 % |

### Sur le corpus d'apprentissage

| Nb gaussiennes | Facteur ML | Nb omissions | Nb substitutions | Nb insertions | % identification | % reconnaissance |
|---|---|---|---|---|---|---|
| 24 | 0 | 15 190 | 67 958 | 30 814 | 76,46 % | 67,74 % |
| 24 | 1 | 18 354 | 63 196 | 20 695 | 76,91 % | 71,05 % |
| 24 | 2 | 21 024 | 60 023 | 15 721 | 77,05 % | 72,60 % |
| 24 | 4 | 26 050 | 56 306 | 10 892 | 76,68 % | 73,60 % |
| 24 | 6 | 31 112 | 54 414 | 8 523 | 75,79 % | 73,37 % |

# Liste des publications

[Jardino, 98] M. Jardino, F. Bimbot, S. Igounet, K. Smaïli, I. Zitouni, M. El-Bèze, "A First Evaluation Campaign for Language Modles", First International Conference on Language Resources & Evaluation, Granada; pp. 801-805, 28-30 mai 1998.

[Béchet, 97] F. Béchet, S. Igounet, "Transcription phonème-graphème à l'aide de méthodes mixtes", BULAG, Actes du colloque international, Fractal 97, Linguistique et Informatique : Théories et Outils pour le Traitement Automatique des Langues, Revue Annuelle, pp. 1-6, décembre1997.

[Igounet, 97] S. Igounet, T. Spriet, M. El-Bèze, "Extraction de Sous-Modèles de Langage sans phase d'apprentisage", 1$^{ères}$ Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 75-78, 15-16 avril 1997.

[Igounet, 96] S. Igounet, "Pour un système hybride de reconnaissance automatique de la parole continue", XXIèmes Journées d'Étude sur la Parole, Avignon, pp. 313-316, 10-14 juin 1996.

[Linares, 96] G. Linares, P. Nocera, S. Igounet, "Adaptation au locuteur par conversions spectrales à l'aide de réseaux neuro-mimétiques", XXIèmes Journées d'Étude sur la Parole, Avignon, pp. 387-390, 10-14 juin 1996.

[Méloni, 95] H. Méloni, S. Igounet, M. El-Bèze, "Parole - TALN : l'espace de la perception", Traitement Automatique du Langage Naturel, TALN 95, Marseille, pp. 107-114, 14-16 juin 1995.

[Igounet, 95] S. Igounet, "Stratégie de décodage pour la parole continue : fondements, exemples et perspectives", Actes des Premières Journées Jeunes Chercheurs en Parole, Paris, 17-18 novembre 1995.

[Béchet, 94] F. Béchet, P. Gilles, H. Méloni, S. Igounet, "Techniques de filtrage lexical : composante lexicale du système SPEX", Actes du séminaire GDR-PRC "Reconnaissance automatique de la parole", CRIN/INRIA Nancy, 10-11 mars 1994.

[Béchet, 94] F. Béchet, S. Igounet, H. Méloni, P. Gilles, "Détection de mots clés dans la parole continue", XXèmes Journées d'Étude sur la Parole, Trégastel, 1-3 juin 1994.

[Igounet, 93] S. Igounet, F. Béchet, H. Méloni, P. Gilles, "*Un système analytique de reconnaissance de mots clés dans la parole continue*", Acte du séminaire GDR-PRC Reconnaissance automatique de la parole, CRIN/INRIA Nancy, 10-11 mars 1994.

[Béchet, 93] F. Béchet, P. Gilles, H. Méloni, S. Igounet, "*Techniques de filtrage lexical : composante lexicale du système SPEX*", Acte du séminaire GDR-PRC Reconnaissance automatique de la parole, CRIN/INRIA Nancy, 10-11 mars 1994.

[Igounet, 93] S. Igounet, "Un système analytique de reconnaissance de mots clés dans la parole continue", Mémoire de D.E.A. Informatique et Mathématiques, Université d'Avignon et des Pays de Vaucluse, septembre 1993.

# Bibliographie

[Adda, 97a] G. Adda, M. Adda-Decker, J. L. Gauvain, L. F. Lamel, "Le système de dictée du LIMSI pour l'évaluation AUPELF'97", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 35-40, 15-16 avril 1997.

[Adda, 97b] G. Adda, M. De Calmès, L. F. Lamel, G. Pérennou, M. Rajman, S. Rosset, J. Zeiliger, "Ressources pour l'apprentissage, de développement et l'évaluation des systèmes de dictée vocale en français : corpus de texte, de parole et lexical", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 305-309, 15-16 avril 1997

[Aho, 72] A. Aho, J. Ullman, "The Theory of Parsing, Translation and Compiling", Vol. 1 : Parsing, Englewood Cliffs, Prentice-Hall, 1972.

[Alessandro, 97] C. d'Alessandro *et al*, "Vers l'évaluation de sustèmes de synthèse de parole à partir du texte en français", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 393-397, 15-16 avril 1997.

[Anglade, 94] Y. Anglade, "Robustesse de la reconnaissance automatique de la parole : étude et application dans un système d'aide vocal pour une standardiste mal-voyante", Thèse de doctorat en informatique, Université de Nancy I, février 1994

[Bahl, 83] L. R. Bahl, F. Jelinek, R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, N°2, pp. 179-190, mars 1983.

[Bahl, 86] L. R. Bahl, P. F. Brown, P. V. De Souza, R. L. Mercer, "Maximum, Mutual Information of Hidden Markov Model Parameters", ICASSP, Tokyo, Vol. 1, pp. 49-52, 1986.

[Bahl, 89] L. R. Bahl, P. Brown, R. Bakis, J. Bellegarda, P. F. Brown, D. Burshtein, S. K. Das, P. V. de Souza, P. S. Gopalakrishnan, F. Jelinek, D. Kanevsky, R. L. Mercer, A. J. Nadas, D. Nahamoo, M. A. Picheny, "Large Vocabulary Natural Language Continuous Speech Recognition", Proc. ICASSP, pp. 465-467, 1989.

[Baker, 75] J. K. Baker, "Stochastic modeling for automatic speech understanding", Speech Recognition, edited by D. Raj Reddy, Academic Press, 1975.

[Baker, 89] J. M. Baker, "DRAGONDICTATE (TM) - 30k : Natural Language Speech Recognition With 30,000 Words", Eurospeech 89, Paris, Vol. 2, pp. 161-163, Septembre 1989.

[Bakis, 76] R. Bakis, "Continuous speech recognition via centisecond acoustic states", 91th. Meeting of the ASA, Washington DC, avril 1976.

[Barras, 96] C. Barras, "Reconnaissance de la parole continue : adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés", Thèse de doctorat en informatique, Université de Paris VI, mai 1996.

[Baum, 72] L. Baum, "An Inequality and Association Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes", Academic Press, Inequalities, Vol. 3, pp. 1-8, 1972.

[Béchade, 92] H. D. Béchade, "Phonétique et morphologie du français moderne et contemporain", Presses Universitaires de France, avril 92.

[Béchet, 94a] F. Béchet, "Un système de traitement de connaissances phonétiques et lexicales : application à la reconnaissance de mots isolés sur de grands vocabulaires et à la recherche de mots cibles dans un discours continu", Thèse de doctorat en informatique, Université d'Avignon et des Pays de Vaucluse, janvier 1994.

[Béchet, 94b] F. Béchet, P. Gilles, H. Méloni, S. Igounet, "Techniques de filtrage lexical : composante lexicale du système SPEX", Actes du séminaire GDR-PRC "Reconnaissance automatique de la parole", CRIN/INRIA Nancy, 10-11 mars 1994.

[Béchet, 94c] F. Béchet, S. Igounet, H. Méloni, P. Gilles, "Détection de mots clés dans la parole continue", XXèmes Journées d'Étude sur la Parole, Trégastel, 1-3 juin 1994.

[Béchet, 95] F. Béchet, S. Derderian, M. El-Bèze, "Conversion graphèmes-phonèmes automatique : le système GRIPHON", IA 95 - Génie Linguistique, 15èmes Journées Internationales, Montpellier, juin 1995.

[Béchet, 97a] F. Béchet, M. El-Bèze, "Automatic Assignment of Part-Of-Speech to Out-Of-Vocabulary Words for Text-To-Speech Processing", Eurospeech, Rhodes, Vol. N°2, pp. 983-988, septembre 1997.

[Béchet, 97b] F. Béchet, T. Spriet, M. El-Bèze, "Traitement Spécifique des Nom Propres dans un Système de Transcription Graphème-Phonème", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 469-472, 15-16 avril 1997.

[Béchet, 97c] F. Béchet, S. Igounet, "Transcription phonème-graphème à l'aide de méthodes mixtes", BULAG, Actes du colloque international, Fractal 97, Linguistique et Informatique : Théories et Outils pour le Traitement Automatique des Langues, Revue Annuelle, pp. 1-6, décembre1997.

[Belrhali, 95a] R. Belrhali, "Phonétisation automatique d'un lexique général du français : systémique et émergence linguistique", Thèse l'Université Stendhal de Grenoble 3, UFR Sciences du Langage, novembre 1995.

[Belrhali, 95b] R. Belrhali, D. Dujardin, J. Courtin, L. J. Boë, "BDPHO : Une base de données lexicales orthographique-phonétique lemmatisée du français parlé", JADT, IIIe Journées Internationales d'Analyse Statistique des Données Textuelles, Rome, 11-13 décembre 1995.

[Bimbot, 97a] F. Bimbot, "Projet SIROCCO : Système générique de reconnaissance de la parole", Document de travail, septembre 1997.

[Bimbot, 97b] F. Bimbot, "Projet SIROCCO : Spécifications de conception", Document de travail, décembre 1997.

[Boite, 97] J. M. Boite, F. Bataille, O. Deroo, S. Dupont, V. Fontaine, Ch. Ris, L. Zanoni, "STRUT : un logiciel complet pour l'entraînement et la reconnaissance de la parole", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 41-45, 15-16 avril 1997.

[Boula, 94] P. Boula de Mareuil, "Vers une phonémisation automatique des sigles", XXèmes Journées d'Étude sur la Parole, Trégastel, pp. 95-100, 1er au 3 juin 1994.

[Bridle, 79] J. S. Bridle, M. D. Brown, "Connected Word Recognition Using Whole Word Templates", Proc. Institute of Acoustics, Autumn Conf., pp. 25-28, novembre 1979.

[Bridle, 82] J. S. Bridle, M. D. Brown, R. M. Chamberlain, "An algorithm for connected word recognition", in Proceeding of the International Conference on Acoustics, Speech and Signal Processing, pp. 899-902, 1982.

[Brill, 92] E. Brill, "A Simple Rule-Based Part of Speech Tagger", 3rd Conference on Applied Natural Language Processing, Trento, Italie, 1992.

[Calliope, 89] Calliope, "La parole et son traitement automatique", CNET, Masson, 1989.

[Cerf-Danon, 90] H. Cerf-Danon, P. De La Noue, L. Diringer, M. El-Bèze, J. C. Marcadet, "A 20 000 words, automatic speech recognizer. Adaptation to French of the US TANGORA system", Nato, Cetrato, Italie, 1990.

[Cerf-Danon, 91] H. Cerf-Danon, S. DeGennaro, M. Ferretti, J. Gonzalez, E. Keppel, "TANGORA - A Large Vocabulary Speech Recognition System For Five Languages", Eurospeech, Genova, Vol. 1, pp. 183-192, 24-26 septembre 1991.

[Chanod, 94] J. P. Chanod, P. Tapanainen, "Développements en analyse syntaxique automatique", Traitement Automatique du Langage Naturel, TALN 94, Marseille, pp. 87-91, avril 1994.

[Chollet, 82] G. Chollet, C. Cagnoulet, "On the evaluation of speech recognizers and data bases using a reference system", Proc. ICASSP, Paris, pp. 2026-2029, 1982.

[Chollet, 95] G. Chollet, "Evaluation of ASR systems, algorithms, and databases", in Speech Recognition and Coding New Advances and Trends, A. Rubio Ayuso et J. Lopez Soler, NATO ASI Series, Vol. F 147, Springer-Verlag, pp. 32-40, 1995.

[Chomsky, 52] N. Chomsky, "Syntactic Structures", La Haye, Mouton, 1957.

[Church, 89] K. Church, "A Stochastic Parts Program Noun Phrase Parser for Unrestricted Text", ICASSP 1989, Glasgow, pp. 695-698, 1989.

[Church, 89] K. Church, R. Mercer, "Introduction to the Special Issues on Computional Linguistics Using Large Corpora", Computational Linguistics, Vol. 19, N°1, pp. 1-24, mars 1993.

[Clarkson, 97] P. Clarkson, R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit", Eurospeech, Rhodes, Vol. N°5, pp. 2707-2710, septembre 1997.

[Cravero, 86] M. Cravero, R. Pieraccini, F. Raineri, "Definition and Evaluation of Phonetic Units for Speech Recognition by Hidden Markov Models", ICASSP 1986, Tokyo, pp. 2235-2238, 1986.

[Davis, 80] S. B. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Trans. Acoust. Speech, Signal Procesing, Vol. 23, N°4, pp. 357-366, 1980.

[De Mori, 85] R. De Mori, P. Laface, Y. Mong, "Parallel Algorithms for Syllable Recognition in Continuous Speech", IEEE Trans. PAMI, Vol. 7, N°1, pp. 56-59, janvier 1985.

[Delattre, 66] P. Delattre, "Studies in French and Comparative phonetics", Mouton & Co, La Hague, 1966.

[Derouault, 84] A. M. Derouault, F. Jelinek, "Modèle Probabiliste d'un Langage en Reconnaissance de la Parole", Annales des Télécommunications, 39, N° 3-4, pp. 143-151, 1984.

[Derouault, 85] A. M. Derouault, "Modélisation d'une langue naturelle pour la désembiguation des chaînes phonétiques", Thèse de Doctorat d'État ès Sciences, Université de Paris VII, avril 1985.

[Derouault, 87a] A. M. Derouault, "Context-dependant phonetic Markov Models for large vocabulary speech recognition", ICASSP 87, Dallas, pp. 360-363, 1987.

[Derouault, 87b] A. M. Derouault, "Context-dependant phonetic Markov Models for Speech Recognition", NATO Advanced Institute on Pattern Recognition, pp. 171-175, 5-18 juillet 1987.

[Dolmazon, 97] J. M. Dolmazon, F. Bimbot, G. Adda, M. El-Bèze, J. C. Caërou, J. Zeiliger, M. Adda-Decker, "Organisation de la première campagne AUPELF pour l'évaluation des systèmes de dictée vocale", 1$^{\text{ères}}$ Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 13-18, 15-16 avril 1997.

[Dupond, 96] P. Dupond, "Utilisation et Apprentissage de Modèles de Langage pour la Reconnaissance de la Parole Continue", Thèse de Doctorat spécialité Signal et Images, École Nationale Supérieure des Télécommunications, avril 1996.

[El-Bèze, 90] M. El-Bèze, "Choix d'Unités Appropriées et Introduction de Connaissances dans des Modèles Probabilistes pour la Reconnaissance Automatique de la Parole", Thèse de Doctorat en Informatique Fondamentale, Université Paris VII, novembre 1990.

| | |
|---|---|
| [El-Bèze, 93] | M. El-Bèze, "Les modèles de langage probabilistes : quelques domaines d'application", Habilitation à diriger des recherches (informatique), LIPN : Université Paris-Nord, janvier 1993. |
| [El-Bèze, 95] | M. El-Bèze, T. Spriet, "Intégration de Contraintes syntaxiques dans un système d'étiquetage probabiliste", Traitement Automatique des Langues, Vol. 36, Numéro 1-2, pp.47-66, 1995. |
| [Fisher, 86] | W. M. Fisher, G. R. Doddington, K. M. Goudie-Marshall, "The DARPA Speech Recognition Research Database : Specifications and Status", Proc. DARPA Speech Recognition Workshop, 1986. |
| [Forh, 96] | D. Fohr, J. F. Mari, J. P. Haton, "Utilisation de modèles de Markov pour l'étiquetage automatique et la reconnaissance de BREF80", XXIèmes Journées d'Étude sur la Parole, Avignon, pp. 339-342, 10-14 juin 1996. |
| [Forh, 97] | D. Fohr, J. P. Haton, J. F. Mari, K. Smaïli, I. Zitouni, "MAUD : Un prototype de machine à dicter vocale", 1$^{ères}$ Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 25-30, 15-16 avril 1997. |
| [Forney, 73] | G. D. Forney, "The Viterbi algorithm", Proceedings of the IEEE, Vol. 61, N°3, pp. 268-278, mars 1973. |
| [Gauvain, 90] | J. L. Gauvain, L. F. Lamel, M. Eskénazi, "Design considerations and text selection for BREF, a large french read-speech corpus", Proceeding ICSLP-90, Kobe, novembre 1990. |
| [Gauvain, 93] | J. L. Gauvain, L. F. Lamel, G. Adda, M. Adda-Decker, "Speaker-Independant Continuous Speech Dictation", Eurospeech, Berlin, Vol. 1, pp. 125-128, 21-23 septembre 1993. |
| [Gauvain, 94] | J. L. Gauvain, C. H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains", IEEE Trans. on Speech and Audio Precessing, Vol. 2, N°2, pp. 291-298, 1984. |
| [Gauvain, 97] | J. L. Gauvain, "Évaluation et avancées en reconnaissance de la parole : de *Ressource Management* à *Broadcast News*", 1$^{ères}$ Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 11-12, 15-16 avril 1997. |
| [Gazdar, 89] | G. Gazdar, C. Mellish, "Natural Language Processing in Prolog : An Introduction to Computational Linguistics", Reading, Massachusetts, Addison-Wesley, 1989. |
| [Ghneim, 95] | N. Ghneim, V. Aubergé, "Optimisation de la grammaire de phonétisation du français TOPH en vue de la correction d'orthographe", 3$^{rd}$ Int. Conf. on Statistical Analysis of Textual Data, JADT, Rome, 1995. |
| [Gilles, 93] | P. Gilles, "Décodage phonétique de la parole et adaptation au locuteur", Thèse de doctorat en informatique, Université d'Avignon et des Pays de Vaucluse, janvier 1993. |
| [Godin, 89] | C. Godin, P. Lockwood, "DTW schemes for continuous speech recognition : a unified view", Computer Speech and Language, Vol. 3, pp. 169-198, 1989. |
| [Grévisse, 93] | M. Grévisse, A. Goosse, "Le bon usage - Grammaire française", Treizième édition, 1993. |
| [Harrison, 78] | M. A. Harrison, "Introduction to Formal Language Theory", Reading, Massachusetts, Addison-Wesley, 1978. |
| [Hart, 68] | P. Hart, N. Nilsson, B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Trans. Systems Sci. Cyberne, Vol. 4, N°2, pp. 100-107, juillet 1968. |

[Haton, 91] J.P. Haton, J.M. Pierrel, G. Perennou, J. Caelen, J.L. Gauvain, "Reconnaissance automatique de la parole", DUNOD Informatique, 1991.

[Hochberg, 94] M. Hochberg, S. Renals, A. Robinson, "ABBOT : The CUED hybrid connectionist-HMM large-vocabulary recognition system", in Proc. ARPA Spoken Language Technology Workshop, 1994.

[Hwang, 93a] M. Y. Hwang, F. Alleva, X. Huang, "Senones, Multi-Pass Search, and Unified Stochastic Modeling in SPHINX-II", Eurospeech, Berlin, Vol. 3, pp. 2143-2146, 21-23 septembre 93.

[Hwang, 93b] M. Y. Hwang, X. Huang, "Shared Distribution Hidden Markov Model for Speech recognition", IEEE Trans. on Speech and Audio Processing, Vol. 1, N°4, pp. 414-420, 1993.

[Igounet, 93] S. Igounet, "Un système analytique de reconnaissance de mots clés dans la parole continue", Mémoire de D.E.A. Informatique et Mathématiques, Université d'Avignon et des Pays de Vaucluse, septembre 1993.

[Igounet, 95] S. Igounet, "Stratégie de décodage pour la parole continue : fondements, exemples et perspectives", Actes des Premières Journées Jeunes Chercheurs en Parole, Paris, 17-18 novembre 1995.

[Igounet, 96] S. Igounet, "Pour un système hybride de reconnaissance automatique de la parole continue", XXIèmes Journées d'Étude sur la Parole, Avignon, pp. 313-316, 10-14 juin 1996.

[Igounet, 97] S. Igounet, T. Spriet, M. El-Bèze, "Extraction de Sous-Modèles de Langage sans phase d'apprentisage", 1ères Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 75-78, 15-16 avril 1997.

[Itakura, 75] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans. ASSP, Vol. 23, N°1, pp. 67-72, février 1975.

[Jelinek, 76] F. Jelinek, "Continuous Speech Recognition by Statistical Methods", Proc. of the IEEE, Vol. 64, N°4, pp.532-556, avril 1976.

[Jelinek, 80] F. Jelinek, R. L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data", Pattern Recognition in practice, E. Gelsema and L. N. Kanal, North-Holland, pp. 381-397, 1980.

[Jelinek, 82] F. Jelinek, R. L. Mercer, L. R. Bahl, "Continuous Speech Recognition : Statistical Methods", © North-Holland Publishing Company, Handbook of Statistics, Vol. 2, Classification, pattern recognition and reduction of dimensionality, P. R. Krishnaiah and L. N. Kanal eds, pp. 549-573, 1982.

[Jelinek, 90] F. Jelinek, "Self-Organized Langage Modeling for Speech Recognition", Readings in Speech Recognition, Ed. by A. Waibel & Kai-fu Lee, Morgan Kaufmann, pp. 450-506, 1990.

[Jouvet, 88] D. Jouvet, "Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques", Thèse de Doctorat spécialité Traitement du Signal, École Nationale Supérieure des Télécommunications, juin 1988.

[Jouvet, 94] D. Jouvet, M. Dautremont, A. Gossart, "Comparaison des multimodèles et des densités multigaussiennes pour la reconnaissance de la parole par modèle de Markov", XXèmes Journées d'Étude sur la Parole, Trégastel, pp. 159-164, 1er au 3 juin 1994.

[Jouvet, 95] D. Jouvet, "Modèle de Markov pour la reconnaissance de la parole", École thématique : Fondements et Perspectives en Traitement Automatique de la Parole, Marseille, pp. 99-108, 17-25 juillet 1995.

[Kapadia, 93] S. Kapadia, V. Valtchev, S. J. Young, "MMI Training for Continous Phoneme Recognition on the TIMIT Database", ICASSP, Minneapolis, Vol. 2, pp. 491-494, 1993.

[Katz, 87] S. M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-35, pp. 400-410, mars 1987.

[Klatt, 77] D. H. Klatt, "Review of the ARPA Speech Understanding Project ", JASA, Vol. 62, N°6, pp. 1345-1366, décembre 1977.

[Lacouture, 95] R. Lacouture, "Au sujet des algorithmes de recherche des systèmes de reconnaissance de la parole à grand vocabulaire", Thèse de doctorat en vue de l'obtention du grade de Philosophiae Doctor, Faculté des Études Supérieures et de la Recherche, Université McGill, Montréal, septembre 1995.

[Lamel, 86] L. F. Lamel, R. H. Kassel, S. Seneff, "Speech Database Development : Design and analysis of the acoustic-phonetic corpus", Proc. DARPA Speech Recognition Workshop, 1986.

[Lamel, 91] L. F. Lamel, J. L. Gauvain, M. Eskénazi, "BREF, a Large Vocabulary Spoken Coprus for French", Eurospeech, Gênes, Vol 2, pp. 505-508, 24-26 septembre 1991.

[Lamel, 93a] L. F. Lamel, J. L. Gauvain, "High Performance Speaker-Independent Phone Recognition Using CGHMM", Eurospeech, Berlin, Vol. 1, pp. 121-124, 21-23 septembre 1993.

[Lamel, 95] L. F. Lamel, M. Adda-Decker, G. Adda, J. L. Gauvain, "Reconnaissance Multilingue de Grands Vocabulaires", École thématique : Fondements et Perspectives en Traitement Automatique de la Parole, Marseille, pp. 119-128, 17-25 juillet 1995.

[Lamel, 96] L. F. Lamel, G. Adda, M. Adda-Decker, "Les lexiques de prononciation dans les systèmes de reconnaissance de la parole", Séminaire GDR-PRC, Lexique et communication parlée, Toulouse, pp. 1-10, 15-16 octobre 1996.

[Laporte, 89] E. Laporte, M. Silberztein, "Vérification et correction orthographique assistées par ordinateur", Actes de la 1ère conférence européenne sur les techniques et les applications de l'Intelligence Artificielle en milieu industriel et de service, Hermès, 1989.

[Lee, 89] K. F. Lee, H. W. Hon, M. Y. Hwang, S. Mahajan, R. Reddy, "The SPHINX Speech Recognition System", ICASSP 89, Glasgow, pp. 445-448, 1989.

[Lee, 90a] K. F. Lee, H. W. Hon, R. Reddy, "An Overview of the SPHINX Speech Recognition System", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, N°1, pp. 35-45, 1990.

[Lee, 90b] K. F. Lee, "Context-Dependant Phonetic Hidden Markov Models for Continuous Speech Recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, N°4, pp. 599-609, 1990.

[Levinson, 83] S. E. Levinson, L. R. Rabiner, M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", © American Telephone and Telegraph Compagny, The Bell System Technical Journal, Vol. 62, N°4, pp. 1035-1074, avril 1983.

[Levinson, 85] S. E. Levinson, "Structural Method in Automatic Speech Recognition", Proceedings of the IEEE, Vol. 73, N°11, pp. 1625-1650, novembre 1985.

[Linares, 96] G. Linares, P. Nocera, S. Igounet, "Adaptation au locuteur par conversions spectrales à l'aide de réseaux neuro-mimétiques", XXIèmes Journées d'Étude sur la Parole, Avignon, pp. 387-390, 10-14 juin 1996.

| | |
|---|---|
| [Lokbani, 93] | M. N. Lokbani, "Recherche des N Meilleurs Solutions et Post-Traitements, en Reconnaissance de la Parole", Thèse de Doctorat en Sciences, Université de Paris-Sud, septembre 1993. |
| [Lowerre, 90] | B. Lowerre, R. Reddy, "The Harpy Speech Understanding System", Readings in Speech Recognition, Ed. by A. Waibel & Kai-fu Lee, Morgan Kaufmann, pp. 576-586, 1990. |
| [Marcadet, 91] | J. C. Marcadet, "Élaboration d'un poste de travail pour la Dictée Automatique", Thèse de Doctorat en Instrumentation et Informatique, Université de Paris 7, juillet 1991. |
| [Mari, 94] | J. F. Mari, D. Fohr, J. P. Haton, "Modèles stochastiques d'ordres 1 et 2", XX$^{èmes}$ Journées d'Étude sur la Parole, Trégastel, pp. 199-202, 1$^{er}$ au 3 juin 1994. |
| [Mari, 96] | J. F. Mari, "Perception de signaux complexes et interaction homme-machine", Habilitation à Diriger des Recherches spécialité Informatique, Centre de Recherche en Informatique de Nancy, 1996. |
| [Mariani, 90] | J. Mariani, "Reconnaissance automatique de la parole : progrès et tendances", Traitement du Signal, Volume 7, N°4, pp. 239-266, 1990. |
| [Méloni, 91] | H. Méloni, P. Gilles, "Décodage Acoustico-Phonétique ascendant", Traitement du Signal, Volume 8, N°2, pp. 107-114, 1991. |
| [Méloni, 95] | H. Méloni, S. Igounet, M. El-Bèze, "Parole - TALN : l'espace de la perception", Traitement Automatique du Langage Naturel, TALN 95, Marseille, pp. 107-114, 14-16 juin 1995. |
| [Mercier, 85] | G. Mercier, M. Gilloux, C. Tarridec, J. Vaissière, "A New Rule Based Expert System for Speech Recognition", in New Systems and Architecture for Automatic Speech Recognition and Synthesis", R. De Mori and Suen, NATO ASI Series, Serie F, Vol. 16, Springer-Verlag, 1985. |
| [Mérialdo, 87] | B. Mérialdo, A. M. Derouault, M. El-Bèze, S. Soudoplatoff, "Reconnaissance de la parole avec un très grand vocabulaire", 16$^{èmes}$ Journées d'Étude sur la Parole, Hammamet, pp. 161-164, 5-9 octobre 1987. |
| [Mérialdo, 88a] | B. Mérialdo, "Phonetic recognition using Hidden Markov Models and Maximum Mutual Information Training", ICASSP, Ney-York, Vol. S1, pp. 111-114, 1988. |
| [Mérialdo, 88b] | B. Mérialdo, "Multilevel decoding for Very-Large-Size-Dictionary speech recognition", IBM, Journal of Research and Development, Vol. 32, N°2, 1988. |
| [Miller, 90] | P. Miller, T. Torris, "Formalismes Syntaxiques pour le Traitement Automatique du Langage Naturel", Paris, Hermès, 1990. |
| [Morgan, 95] | N. Morgan, H. Bourlard, "Continous Speech Recognition - An introduction to the hybrid HMM/connectionist approach", IEEE Signal Processing Magazine, pp. 25-42, mai 1995. |
| [Myers, 81] | S. Myers, L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, N°2, pp. 284-296, 1981. |
| [Néel, 95] | F. Néel, "Évaluation de la Reconnaissance et de la Synthèse de la Parole", École thématique : Fondements et Perspectives en Traitement Automatique de la Parole, Marseille, pp. 235-239, 17-25 juillet 1995. |
| [Néel, 96] | F. Néel, "L'interaction vocale avec la machine : applications et perspectives", Nouvelles Interfaces Homme-Machine, Observatoire Français des Techniques Avancées, ARAGO 18, pp. 93-125, décembre 1996. |

[Newell, 73] A. Newell, J. Barnett, J. W. Forgie, C. C. Green, D. H. Klatt, J. C. R. Licklider, J. Munson, D. R. Reddy, W. A. Woods, "Speech Understanding Systems : Final Report of a Study Group", North-Holland/American Elsevier, Amsterdam, 1973.

[Ney, 84] H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-32, N°2, pp. 263-271, avril 1984.

[Ney, 97] H. Ney, S. Ortmanns, "Integration of Language Models into search" NATO ASI, Jersey, 8 au 18 juillet 1997.

[Nilsson, 82] N. Nilsson, "Principles of Artificial Intelligence", Springer-Verlag, Berlin, 1982.

[Normandin, 94] Y. Normandin, R. Cardin, R. De Mori, "High-Performance Connected Digit Recognition Using Maximum Mutual Information Estimation", IEEE Transactions on Speech and Audio Processing, Vol. 2, N°2, pp.299-311, 1994.

[Pérennou, 92] G. Pérennou, M. de Calmès, I. Ferrané, J. M. Pécatte, "Le projet BDLEX de base de données lexicales du français écrit et parlé", Actes du Séminaire Lexique, IRIT-UPS Toulouse, pp. 41-56, 1992.

[Pérennou, 96] G. Pérennou, M. de Calmés, L. Pousse, "LexMHAT : lexiques phonologiques pour le traitement automatique de la parole", Séminaire GDR-PRC, Lexique et communication parlée, Toulouse, pp. 21-32, 15-16 octobre 1996.

[Plénat, 93] M. Plénat, "Observation sur le mot minimal français - L'oralisation des sigles", De Natura Sonorum : Essais de phonologie, Presses Universitaires de Vincennes, pp. 143-172, 1993.

[Poritz, 88] A. B. Poritz, "Hidden Markov Models : a Guided Tour", ICASSP, New-York, Vol. S1, pp. 7-13, 1988.

[Prouts, 80] B. Prouts, "Contribution à la synthèse de la parole à partir de texte, transcription graphème-phonème en temps réel sur micro-processeur", Thèse de docteur ingénieur, Université de Paris Sud (Paris XI, Orsay), 1980.

[Pye, 95] D. Pye, P. C. Woodland, S. J. Young, "Large Vocabulary Multilingual Speech Recognition Using HTK", Eurospeech, Madrid, Vol. 1, pp. 181-184, 18-21 septembre 1995.

[Rabiner, 79] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques", IEEE Trans. ASSP, Vol. 27, N°4, pp. 336-349, août 1979.

[Rabiner, 85] Rabiner, S.E. Levinson "A speaker-independent, syntax-directed, connected word recognition system based on Hidden Markov Models and level building", IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-33, N°3, pp. 561-573, 1985.

[Rabiner, 89] L. R. Rabiner, "A Tutorial on Hidden Markov Models ans Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, N°2, pp. 257-286, février 1989.

[Sakoe, 78] H. Sakoe, S. Chiba, "Dynamic Programming algorithm Optimization for Spoken Word Recognition", IEEE Trans. ASSP, Vol. 26, N°1, pp. 43-49, février 1978.

[Sakoe, 79] H. Sakoe, "Two-level DP-matching - a dynamic programming based pattern matching algorithm for connected word recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-27, N°6, pp. 588-595, 1979.

| | |
|---|---|
| [Savariaux, 97] | C. Savariaux, A. Farhat, M. Héon, D. O'Shaughnessy, C. Z. Lee, "Nouvelles avancées en reconnaissance de la parole continue grand vocabulaire du français basées sur le système de reconnaissance de l'INRS-Télécommunications", 1$^{ères}$ Journées Scientifiques et Techniques FRANCIL 1997, AUPELF-UREF, pp. 31-34, 15-16 avril 1997. |
| [Schwartz, 85] | R. Schwartz, Y. Chow, "Context-Dependant Modeling for Acoustic-Phonetic Recognition of Continuous Speech", Proc. ICASSP, pp. 1205-1208, 1985. |
| [Schwartz, 92] | R. Schwartz, S. Austin, S. Kubala, J. Makhoul, L. Nguyen, P. Placeway, G. Zavaliagkos, "New uses for the N-Best sentence hypotheses within the BYBLOS speech recognition system", Proc. ICASSP, pp.11-14, 1992. |
| [Spriet, 93] | T. Spriet, "Traitement formel de connaissances linguistiques dans un système automatique de traitement de la parole continue : SYRAPAC", Thèse de doctorat en informatique, Université d'Avignon et des Pays de Vaucluse, janvier 1993. |
| [Spriet, 95] | T. Spriet, M. El-Bèze, "Étiquetage Probabiliste et Contraintes Syntaxiques", TALN 95, Marseille, pp. 115-123, 14-16 juin 1995. |
| [Spriet, 96] | T. Spriet, F. Béchet, M. El-Bèze, C. De Loupy, L. Khouri, "Traitement Automatique des Mots Inconnus", TALN 96, Marseille, pp. 170-179, 22-24 mai 1996. |
| [Spriet, 97a] | T. Spriet, M. El-Bèze, "Introduction of Rules into a Stochastic Approach for Language Modeling", NATO ASI Summer school, juillet 1997. |
| [Spriet, 97b] | T. Spriet, M. El-Bèze, "Réaccentuation automatique de textes", BULAG, Actes du colloque international, Fractal 97, Linguistique et Informatique : Théories et Outils pour le Traitement Automatique des Langues, Revue Annuelle, pp. 391-398, décembre1997. |
| [Véronis, 87] | J. Véronis, "Correction of phonographic errors in natural language processing", 8th Int. Conf. on Computers in the Humanities", Colombia, 1987. |
| [Vintsyuk, 71] | T. K. Vintsyuk, "Element-wise recognition of continuous speech composed of words from a specified dictionary", Kibernetica, N°2, pp. 133-143, 1971. |
| [Viterbi, 67] | A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Vol. 3, N°2, pp. 260-267, avril 1967. |
| [Woodland, 92] | P. C. Woodland, S. J. Young, "Benchmark DARPA RM Results with the HTK Portable HMM Toolkit", Ptoc. DARPA Continuous Speech Recognition Workshop, Stanford, septembre 1992. |
| [Woodland, 95] | P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, S. J. Young, "The 1994 HTK Large Vocabulary Speech Recognition System", Proc. ICASSP, Detroit, pp. 73-76, 1995. |
| [Young, 89] | S. J. Young, N. H. Russel, J. H. S. Thornton, "Token Passing : a Simple Conceptual Model for Connected Speech Recognition Systems", Cambridge University Engineering Departement, rapport technique, 1989. |
| [Young, 93] | S. J. Young, P. C. Woodland, "HTK : Hidden Markov Model Toolkit V1.5", Cambridge University Engineering Departement Speech Group and Entropic Research Laboratories Inc, 1993. |
| [Young, 94a] | S. J. Young, "The HTK Hidden Markov Model Toolkit : Design and Philosophy", Cambridge University Engineering Departement, rapport technique, 1994. |
| [Young, 94b] | S. J. Young, P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition", Computer Speech and Language, Vol. 8, pp. 369-384, 1994. |

[Young, 96] S. J. Young, J. Jansen, J. Odell, D. Ollason, P. C. Woodland, "The HTK Book - For HTK V2.0", Entropic Cambridge Research Laboratory Ltd, Copyright 1995-1996 Cambridge University, 1996.

[Yvon, 93] F. Yvon, "A Tidy rude-based grapheme to phoneme transcriber for ONOMASTICA", Actes du 1er colloque ONOMASTICA, Londres, pp. 47-56, 1993.

## Articles de presse

[Cornuau, 93] J. Cornuau, F. Ille, "La saisie vocale des comptes-rendus médicaux en langage naturel", Informatique & Santé - La revue, N°14, pp. 40-41, 3$^{ème}$ trimestre 1993.

[Gaveau, 96] F. Gaveau, "Il dicte son courrier à son ordinateur !", Le Monde Informatique, pp. 39-40, novembre 96.

[Jovanovic, 94] S. Jovanovic, "Le radiologue dicte à l'ordinateur", AIX mag, N°4, p. 19, février 1994.

[Le Hir, 97] P. Le Hir, "L'homme s'efforce de converser avec les ordinateurs", Le Monde, jeudi 2 janvier 1997.

[Neuman, 95a] F. Neuman, "Comment ça marche : la reconnaissance vocale", Science & Vie Micro, pp. 220-223, juin 1995.

[Neuman, 95b] F. Neuman, "Dragon Dictate 1.3 version francaise : la voix comme seule interface", Science & Vie Micro, pp. 144-147, novembre 1995.

[Neuman, 97] F. Neuman, "ViaVoice : je parle donc j'écris", Science & Vie Micro, pp. 90-93, septembre 1997.

[Romain, 94] D. Romain, "Pilotez votre PC à la voix", InfoPC, pp.175-178, avril 1994.

# Table des figures et des tableaux

## Table des figures

Figure 1.1 : Synoptique d'un SRAP utilisant une approche analytique Schéma général des modules analytiques du LIA ....15
Figure 1.2 : Exemple d'un treillis de phonèmes suite au processus de DAP sur le mot "abaisser" ....16
Figure 1.3 : Synoptique d'un SRAP utilisant une approche globale ....18
Figure 1.4 : Chaîne de traitement de l'information....19
Figure 1.5 : Exemple de HMM ....21
Figure 1.6 : Exemple de HMM de Bakis ....23
Figure 1.7 : Chemin optimal pour la stratégie *Une Passe*....32
Figure 1.8 : Chemin optimal pour la stratégie *Niveaux Successifs* ....33
Figure 1.9 : Structure d'un modèle ....36
Figure 1.10 : Matrice du calcul des coûts....37
Figure 1.11 : Algorithme de base du Modèle de Propagation de Jeton....38
Figure 1.12 : Reconnaissance de mots connectés avec le Modèle de Propagation de Jeton....39
Figure 1.13 : Réseau syntaxique et modèle de langage probabiliste bi-gramme ....40
Figure 1.14 : Topologie des algorithmes *Une Passe* et *Niveaux Successifs*....40

Figure 2.1 : Structure d'un SRAP avec HTK ....48
Figure 2.2 : Synoptique du système de phonétisation pour la RAP....53
Figure 2.3 : Exemple d'alignement pour le mot *entretenu* (phrase I0ME0185)....62
Figure 2.4 : Topologie trois états sans saut de type Bakis ....63
Figure 2.5 : Topologie deux états avec saut de type Bakis ....63
Figure 2.6 : Première étape de la phase d'initialisation....66
Figure 2.7 : Seconde étape de la phase d'initialisation....67
Figure 2.8 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre de gaussiennes ....72
Figure 2.9 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du poids du modèle de langage....75
Figure 2.10 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre d'itérations de la phase de ré-estimation....76

Figure 3.1 : Structure d'un module de décodage en RAP....81
Figure 3.2 : Liste homophone ....87
Figure 3.3 : Fonction de coût et graphe de recherche....92
Figure 3.4 : Synoptique général du SRAPC (approche A*)....94

Figure 4.1 : Décomposition d'un mot composé du lexique L au Sous-Lexique SL....102
Figure 4.2 : Composition de mots du Sous-Lexique SL au Lexique L....102
Figure 4.3 : Mesures usuelles sur différents lexiques et ML....108

## Table des tableaux

Tableau 2.1 : Outils logiciels de base de HTK (Version 2.0)....47
Tableau 2.2 : Exemple de phonétisation avec variantes phonétiques....58
Tableau 2.3 : Statistiques sur le nombre de représentants et les durées des unités phonétiques....71
Tableau 2. 4 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre de gaussiennes....73
Tableau 2.5 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du poids du modèle de langage....74
Tableau 2.6 : Résultats de reconnaissance en fonction du type d'initialisation de l'apprentissage et du nombre d'itérations de la phase de ré-estimation....76

Tableau 3.1 : Lexique de décodage et phonétisation....83
Tableau 3.2 : Nombre d'unités phonétiques dans Lex 1, Lex 2 et Lex 3....84
Tableau 3.3 : Résultats de reconnaissance - Lexiques et ML 20k....84
Tableau 3.4 : Lexique de décodage homophonique....88
Tableau 3.5 : Taux de réduction homophone sur 20 000 mots....88
Tableau 3.6 : Exemple de liste homophone solution (décodage de la phrase L12M081 du corpus de test)....88
Tableau 3.7 : Exemple de filtrage acoustico-linguistique (décodage de la phrase L12M081 du corpus de test)....90
Tableau 3.8 : Résultat de reconnaissance - Lexique 20k homophone....91
Tableau 3.9 : Résultat de reconnaissance DAP - Approche A*....96

# Table des matières

Remerciements ....................................................................................................1

Sommaire.............................................................................................................3

Avant-propos .......................................................................................................6

Introduction.........................................................................................................8

Bases théoriques ...............................................................................................11
- 1.1. Caractérisation du problème .................................................................12
- 1.2. Méthodologies de reconnaissance .........................................................14
  - 1.2.1. L'approche analytique ....................................................................14
    - 1.2.1.1. Schéma général .......................................................................15
    - 1.2.1.2. Le Décodage Acoustico-Phonétique........................................16
    - 1.2.1.3. Accès lexical.............................................................................17
  - 1.2.2. L'approche globale .........................................................................17
  - 1.2.3. Commentaires ................................................................................18
- 1.3. L'approche globale probabiliste ...............................................................19
- 1.4. Modélisation Markovienne des probabilités acoustiques et linguistiques........21
  - 1.4.1. Modélisation acoustique probabiliste : les modèles de Markov cachés ....21
    - 1.4.1.1. Présentation...............................................................................21
    - 1.4.1.2. Topologie des HMM pour la RAP...........................................22
    - 1.4.1.3. Reconnaissance d'un modèle ...................................................23
    - 1.4.1.4. Apprentissage des paramètres d'un modèle .............................24
  - 1.4.2. Modélisation linguistique probabiliste............................................25
    - 1.4.2.1. Présentation des modèles de langage probabilistes locaux.................26
    - 1.4.2.2. Modèles n-grammes.................................................................26
    - 1.4.2.3. Autres types de modèles ..........................................................27
  - 1.4.3. Contraintes linguistiques syntaxiques et décodage acoustique..................28
  - 1.4.4. Remarques sur l'implémentation...................................................28
    - 1.4.4.1. Changement d'échelle et accélération des calculs.............................28
    - 1.4.4.2. Facteurs de pénalités................................................................29
  - 1.4.5. Commentaires ................................................................................29
- 1.5. Stratégies de décodage en parole continue ..............................................30
  - 1.5.1. Stratégies de décodage fondatrices ...............................................30
    - 1.5.1.1. Algorithme *Une Passe* .............................................................32
    - 1.5.1.2. Algorithme *Niveaux Successifs*...............................................32
    - 1.5.1.3. Algorithme *Deux Niveaux* ......................................................34
    - 1.5.1.4. Résumé ...................................................................................34
  - 1.5.2. Le système CUED-HTK et le Modèle de Propagation de Jeton (MPJ).....35
    - 1.5.2.1. Reconnaissance de mots isolés ...............................................36

1.5.2.2. Reconnaissance de mots connectés....................................................38
1.5.2.3. Prise en compte des contraintes syntaxiques dans le MPJ..................39
1.5.2.4. Relation du MPJ avec les stratégies de décodage fondatrices ............40
1.5.3. Autres SRAP.........................................................................................41
1.5.3.1. Le système BYBLOS (BBN) ..........................................................41
1.5.3.2. Le système du LIMSI.......................................................................41
1.5.3.3. Le système PARSYFAL (IBM) ......................................................41
1.5.3.4. Le système TANGORA (IBM)........................................................42
1.5.3.5. Les systèmes SPHINX I et II (CMU)...............................................42
1.6. Conclusion..................................................................................................43

Réalisation d'un système de reconnaissance : Décodage Acoustico-Phonétique .......44
2.1. Cadre de l'étude............................................................................................45
2.1.1. Le cadre...................................................................................................45
2.1.2. L'ambition ..............................................................................................46
2.1.3. Les limites...............................................................................................46
2.2. Outils de développement ............................................................................47
2.3. Bases de données .........................................................................................49
2.3.1. Les corpus BREF et BREF-80 ...............................................................49
2.3.2. Les corpus des Actions de Recherches Concertées "Linguistique, Informatique et Corpus Oraux"...........................................................50
2.3.3. Bases de données utilisées .....................................................................51
2.4. Module de phonétisation .............................................................................51
2.4.1. Ensemble des phonèmes ........................................................................52
2.4.2. Phonétisation pour la synthèse de la parole ...........................................52
2.4.3. Phonétisation pour la RAP.....................................................................53
2.4.3.1. Formatage de texte...........................................................................54
2.4.3.2. Étiquetage syntaxique.......................................................................54
2.4.3.3. Phonétisation ...................................................................................54
2.4.3.4. Génération des liaisons.....................................................................56
2.4.3.5. Variantes et phonétisation.................................................................56
2.4.4. Transcription phonèmes-graphèmes .......................................................59
2.4.5. Phonétisation de BREF-80......................................................................60
2.5. Paramétrisation des données acoustiques ....................................................60
2.6. Alignement des données acoustiques et phonétiques ..................................60
2.6.1. Principe de l'alignement automatique ....................................................61
2.6.2. Procédure d'alignement automatique .....................................................61
2.7. Modèles de Markov .....................................................................................62
2.7.1. Topologies .............................................................................................63
2.7.2. Perfectionnements...................................................................................64
2.8. Apprentissage ..............................................................................................65

2.8.1. Initialisation de l’apprentissage ........................................................65
2.8.1.1. Initialisation à partir de l'alignement acoustique et phonétique..........65
2.8.1.2. Initialisation globale moyenne ..........................................................67
2.8.2. Phases de ré-estimation.......................................................................68
2.8.2.1. Principe de la ré-estimation ..............................................................68
2.8.2.2. Ajout de gaussiennes ........................................................................69
2.9. Reconnaissances et résultats ..................................................................69
2.9.1. Protocole expérimental ........................................................................69
2.9.1.1. Expériences de reconnaissance.........................................................69
2.9.1.2. Critères de l'évaluation .....................................................................70
2.9.2. Modèle de langage bi-gramme phonétique pour le DAP ..........................70
2.9.3. Résultats du DAP..................................................................................72
2.9.3.1. Résultats en fonction du nombre de gaussiennes...............................72
2.9.3.2. Résultats en fonction des poids du modèle de langage.......................74
2.9.3.3. Résultats en fonction du nombre d'itérations de la phase de ré-estimation ..................................................................................................75
2.9.3.4. Résultats en fonction du type d'initialisation ......................................77
2.9.4. Comparaison des résultats du DAP ........................................................77
2.10. Conclusion ..........................................................................................78

Réalisation d'un système de reconnaissance : Accès Lexical et Modélisation du Langage........................................................................................................80
3.1. Présentation d'un SRAP de référence ....................................................81
3.1.1. Lexique ...............................................................................................81
3.1.2. Représentation du lexique et prise en compte d'un Modèle de Langage bi-gramme .....................................................................................................82
3.2. Reconnaissance.....................................................................................82
3.2.1. Paramètres du SRAPC de référence ......................................................82
3.2.2. Résultats de reconnaissance..................................................................84
3.3. Reconnaissance et stratégies de décodage ..............................................86
3.3.1. Approche homophone...........................................................................86
3.3.1.1. Réduction du lexique de décodage ....................................................87
3.3.1.2. Première passe : décodage homophone .............................................88
3.3.1.3. Seconde passe : détermination de l’hypothèse optimale.....................89
3.3.1.4. Résultats de reconnaissance..............................................................91
3.3.2. Approche A* ........................................................................................91
3.3.2.1. Fonction de coût ..............................................................................92
3.3.2.2. Décodage et algorithme A* ...............................................................92
3.3.2.3. Pré-calculs et définitions des fonctions de coût..................................94
3.3.2.4. Résultats de reconnaissance..............................................................95
3.4. Conclusion ...........................................................................................96

Modèles de langage : Extraction de Sous-Modèles de Langage ..... 99
  4.1. Introduction ..... 100
  4.2. Position du problème ..... 100
  4.3. Extraction de Sous-Modèles de Langage ..... 101
    4.3.1. Choix du lexique de référence ..... 101
    4.3.2. Modèle de langage bi-gramme avec *back-off* ..... 103
      4.3.2.1. Processus d'extraction des probabilités uni-grammes ..... 103
      4.3.2.2. Processus d'extraction des probabilités bi-grammes ..... 104
    4.3.3. Traitement des mots composés ..... 104
      4.3.3.1. Processus d'extraction des probabilités uni-grammes pour les *mc* ..... 105
      4.3.3.2. Processus d'extraction des probabilités bi-grammes pour les *mc* ..... 105
  4.4. Expériences et résultats ..... 106
  4.5. Conclusion ..... 109

Conclusions et perspectives ..... 110

Annexes ..... 114
  Table des abréviations ..... 115
  Alphabet Phonétique International ..... 117
  Classes syntaxiques ..... 119
  Résultats de reconnaissance du DAP ..... 122

Liste des publications ..... 126

Bibliographie ..... 128

Table des figures et des tableaux ..... 139

Table des matières ..... 142

Index ..... 147

# Index

A

Adda 13; 30; 31; 41; 45; 49; 50; 53; 56; 81; 85; 100; 112; 113
Adda-Decker 13; 30; 31; 41; 45; 49; 50; 53; 56; 81; 85; 100; 112; 113
Aho 28
Alessandro 96
Alleva 42
Anglade 12
Austin 30; 31; 41

B

Bahl 21; 24; 25; 30; 31; 64
Baker 19; 109
Bakis 23; 64
Barnett 9
Barras 45; 60
Bataille 29
Baum 24; 25
Béchade 56
Béchet 16; 17; 19; 52; 53; 57; 59
Bellegarda 64
Belrhali 55; 56
Bigorne 15
Bimbot 45; 46; 50; 85
Boë 56
Boite 29
Bourlard 22
Bridle 18; 31; 32
Brill 89
Brown 18; 25; 31; 32; 64
Burshtein 64

C

Caelen 12
Caërou 45; 50; 85
Cagnoulet 45
Calliope 12; 15; 18; 21
Calmès 56
Cardin 25
Cerf-Danon 24; 27; 30; 42; 100
Chamberlain 31; 32
Chanod 89
Chiba 18
Chollet 45; 47
Chomsky 28
Chow 64
Church 89
Clarkson 29
Cornuau 9; 14
Courtin 56
Cozannet 15
Cravero 23

D

Das 64
Dautremont 60; 64
Davis 60
de Calmès 50
de la Noue 24; 27
De Loupy 53
De Mori 15; 25
De Souza 25; 64
DeGennaro 30; 42; 100
Delattre 56
Derderian 52
Deroo 29
Derouault 23; 27; 41; 89
Diringer 24; 27
Doddington 49
Dolmazon 45; 50; 85
Dujardin 56
Dupond 27; 28
Dupont 29

E

El-Bèze 24; 27; 30; 31; 41; 45; 50; 52; 53; 54; 56; 57; 59; 85; 89; 100
Eskénazi 49

F

Farhat 85
Ferretti 30; 42; 100
Fisher 49
Fohr 22; 51; 67; 77; 78; 85
Fontaine 29
Forgie 9
Forney 24

G

Gauvain 30; 31; 41; 45; 46; 49; 56; 60; 67; 77; 78; 85; 100; 112; 113
Gaveau 9
Gazdar 28
Gilles 16; 17; 19
Gilloux 15
Godin 31; 32; 33; 34
Gonzales 100
Gonzalez 30; 42
Gopalakrishnan 64
Gossart 60; 64
Goudie-Marshall 49
Green 9
Grévisse 56

H

Harrison 28
Hart 91
Haton 12; 22; 51; 67; 77; 78; 85
Héon 85
Hochberg 22
Hon 23; 30; 42; 86
Huang 42; 64
Hwang 23; 42; 64

I

Igounet 16; 17; 19; 30; 31; 59; 61; 100
Ille 9; 14
Itakura 18

J

Jansen 29; 47; 61; 70
Jelinek 19; 21; 24; 26; 27; 30; 31; 64; 99; 103; 104; 107
Jouvet 19; 21; 30; 31; 60; 64; 86
Jovanovic 9; 14

K

Kanevsky 64
Kapadia 25
Kassel 49
Katz 27; 103
Keppel 30; 42; 100
Khouri 53
Klatt 9; 15
Kubala 30; 31; 41

L

Lacouture 86
Laface 15
Lamel 13; 30; 31; 41; 46; 49; 50; 53; 56; 67; 77; 78; 81; 85; 100; 112; 113
Le Hir 9
Lee 23; 30; 42; 60; 64; 85; 86
Leggetter 47; 60
Levinson 18; 21; 30
Licklider 9
Linares 61
Lockwood 31; 32; 33; 34
Lokbani 86; 91
Lowerre 86

M

Mahajan 23
Makhoul 30; 31; 41

Marcadet 27; 41; 42; 63

Mari 22; 51; 67; 77; 78; 85

Mariani 12; 15

Mellish 28

Méloni 16; 17; 19; 31

Mercer 21; 24; 25; 27; 30; 31; 64; 89

Mercier 15

Mérialdo 25; 30; 41

Mermelstein 60

Miller 28

Mong 15

Morgan 22

Munson 9

Myers 30; 31; 32

N

Nadas 64

Nahamoo 64

Néel 9; 45

Neuman 9

Newell 9

Ney 28; 31; 32; 82

Nguyen 30; 31; 41

Nilsson 91

Nocera 61

Normandin 25

O

Odell 29; 47; 60; 61; 70

Ollason 29; 47; 61; 70

Ortmanns 28

O'Shaughnessy 85

P

Perennou 12

Pérennou 50; 56

Picheny 64

Pieraccini 23

Pierrel 12

Placeway 30; 31; 41

Poritz 21

Pousse 56

Prouts 52

Pye 47

R

Rabiner 18; 21; 30; 31; 32

Raineri 23

Rajman 50

Raphael 91

Ravishankar 86

Reddy 9; 23; 30; 42; 86

Renals 22

Ris 29

Robinson 22

Romain 9

Rosenberg 18

Rosenfeld 29

Rosset 50

Russel 30; 31; 35

S

Sakoe 18; 30; 31; 34

Savariaux 85

Schwartz 30; 31; 41; 64

Seneff 49

Smaïli 85

Sondhi 21

Soudoplatoff 41

Spriet 16; 53; 54; 57; 59; 89; 100

T

Tapanainen 89

Tarridec 15

Thornton 30; 31; 35

Torris 28

U

Ullman 28

V

Vaissière 15
Valtchev 25; 47; 60
Vintsyuk 32
Viterbi 24

W

Wilpon 18
Woodland 29; 35; 47; 49; 60; 61; 64; 65; 70; 86
Woods 9

Y

Young 25; 29; 30; 31; 35; 47; 49; 60; 61; 64; 65; 70; 86
Yvon 55

Z

Zanoni 29
Zavaliagkos 30; 31; 41
Zeiliger 45; 50; 85
Zitouni 85