Université de Nancy I — U.E.R. Sciences mathématiques

Centre de Recherche en Informatique de Nancy

CONTRIBUTION À L'INTERPRÉTATION AUTOMATIQUE DES SIGNAUX EN PRÉSENCE D'INCERTITUDE

THÈSE

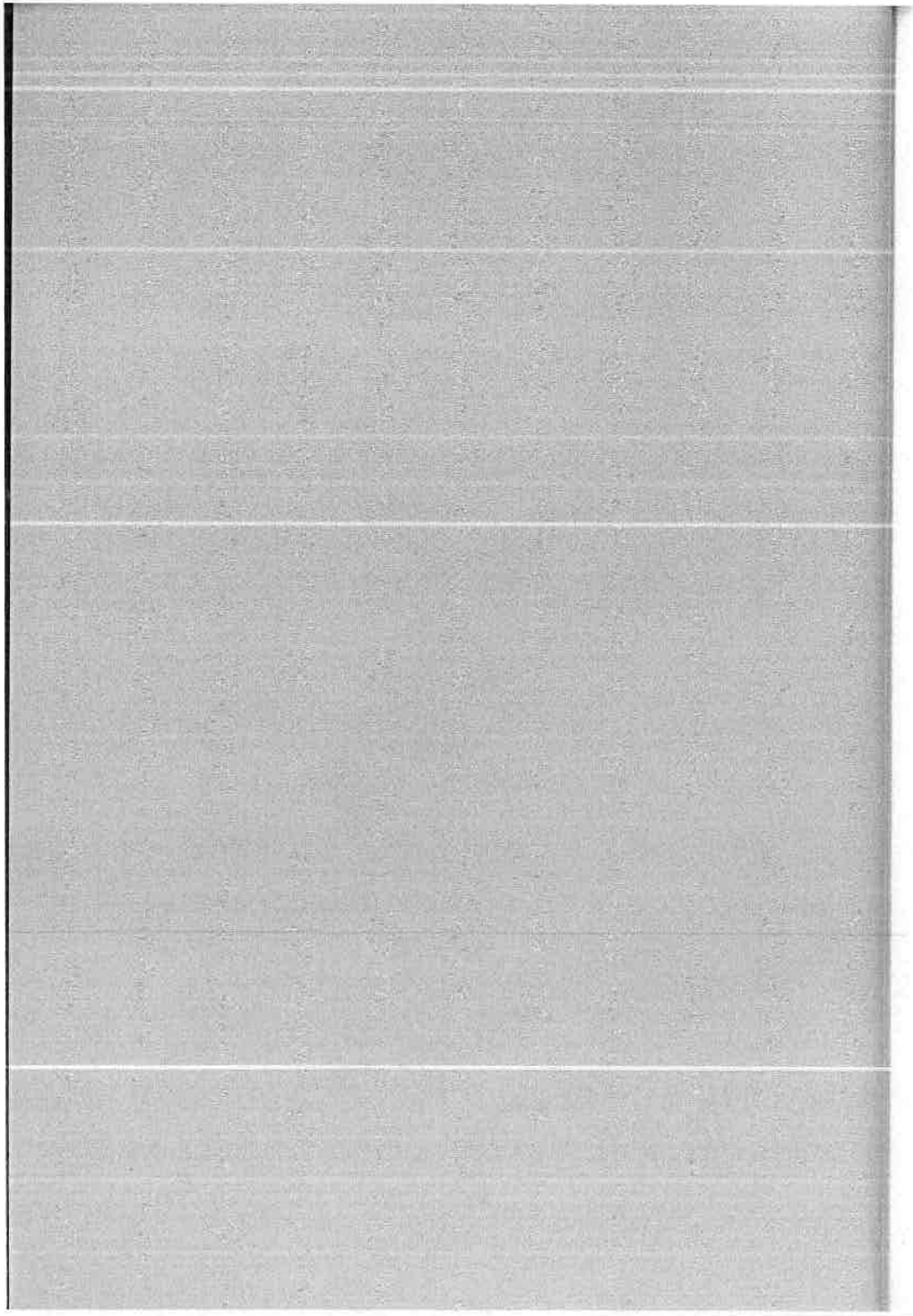
Présentée et Soutenue Publiquement le 11 mai 1988
à Université de Nancy I
pour l'obtention du titre de

DOCTEUR DE L'UNIVERSITÉ DE NANCY I
EN INFORMATIQUE
par

Yifan GONG

Composition du Jury:

Président:	Jean-Pierre	FINANCE
Raporteurs:	Henri	FARRENY
	Jean-Pierre	THOMESSE
Examinateurs:	Jean-Paul	HATON
	Marie-Christine	HATON
	Joseph	MARIANI
	Jean-Marie	PIERREL



à mes parents,
à ma sœur

Jean-Paul Haton, professeur à l'Université de Nancy I, Directeur de recherche à l'INRIA, a bien voulu m'accueillir dans son laboratoire du CRIN et assurer continuellement un excellent environnement de recherche à la présente thèse. Qu'il trouve ici l'expression de ma profonde reconnaissance pour avoir su diriger mon travail.

Jean-Pierre Finance, professeur à l'Université de Nancy I, Directeur du CRIN, me fait l'honneur de présider ce jury. C'est grâce à ses cours, d'une clarté exceptionnelle, que j'ai pu me former en informatique. Je lui exprime ma vive gratitude.

Je suis profondément reconnaissant à Henri Farreny, professeur à l'Université de Toulouse III, pour l'honneur qu'il me fait en ayant accepté d'être rapporteur de cette thèse. Qu'il trouve ici l'expression de tous mes remerciements.

Je remercie sincèrement Jean-Pierre Thomesse, professeur à l'INPL, d'avoir bien voulu être rapporteur de cette thèse malgré l'importance des tâches qu'il assume.

Je tiens à exprimer mes vifs remerciements à Joseph Mariani, Directeur de recherches CNRS au LIMSI, Jean-Marie Pierrel, professeur à l'Université de Nancy I et Directeur de l'ISIAL, Marie-Christine Haton, maître de conférences à l'Université de Nancy I, pour avoir accepté de participer au jury de ma thèse.

Je remercie Gérald Masini et Karl Tombre, qui ont eu la gentillesse de partager avec moi leurs connaissances profondes en langages de programmation dans de nombreuses discussions lors de la réalisation de ce travail.

Mes remerciements vont à tous mes collègues et amis qui ont apporté leur aide et leur encouragement, particulièrement à ceux qui ont consacré leur temps et leur peine dans la lecture de la première version de ce document pour améliorer la qualité du français.

RÉSUMÉ

L'interprétation du signal est un processus de transformations successives d'information vers une forme compréhensible par l'homme ou par la machine. A cause de l'incertitude du signal dûe à la production ou à la transmission du signal, le processus d'interprétation exige l'utilisation des traitements d'intelligence artificielle de différents niveaux d'abstraction.

Dans l'objectif de l'interprétation de la parole continue, nous avons proposé et mis en œuvre un ensemble de travaux sur les quatre aspects importants de l'interprétation: l'édition et modélisation du signal, la conversion du signal en symboles, l'analyse de la structure du signal et l'architecture du système d'interprétation.

Dans la partie I nous présentons d'abord un outil d'édition et de manipulation des signaux qui permet l'extraction et l'évaluation de paramètres et qui aide à l'acquisition des connaissances d'observation sur les signaux. Cet outil fournit également un environnement de simulation des nouveaux algorithmes de traitement à l'aide des opérateurs existants (chapitre 2). Ensuite nous étudions un modèle du signal non-stationnaire destiné à l'estimation de la fréquence fondamentale de la parole, où le signal est modélisé par une séquence de fonctions spécifiées de façon à autoriser la variabilité en fonction du temps de la période et de l'amplitude de l'excitation du signal. Les coefficients du modèle est obtenus par appariement spectral dans le domaine temporel (chapitre 3).

La partie II est consacrée aux problèmes de la conversion du signal vers les symboles en présence d'incertitude. Nous présentons une méthode de classification floue pour l'accès rapide à un grand dictionnaire signal-symbole qui autorise l'association d'un point à plusieurs classes simultanément (chapitre 4) et un système à base de connaissances, capable de traiter des règles imprécises et incertaines et d'effectuer le raisonnement inexact, pour la conversion du signal vers les symboles, lorsque le signal est déformé par le contexte (chapitre 5).

Nous exposons dans la partie III (chapitre 6) un analyseur de structure du signal qui fonctionne en mode contrôlé par les données et par les modèles et qui développe ses solutions en parallèle, avec un mécanisme de recherche en faisceau, afin de propager l'incertitude. Cet analyseur est capable de construire la structure syntaxique à partir de plusieurs îlots de confiance, avec des stratégies différentes.

Une architecture de système d'interprétation des signaux à niveaux et à connaissances multiples est présentée dans la partie IV (chapitre 7). Elle est fondée sur la décomposition du problème d'interprétation en sous-problèmes à des niveaux conceptuels successifs et sur la définition du contrôle explicitement lié à ces niveaux. Chaque niveau a en plus un contrôle local. L'échange d'information est assuré par une structure de données commune entre les sources de connaissances d'un même niveau et par un mécanisme de courrier entre deux niveaux différents.

Enfin, dans la partie V nous illustrons un système d'interprétation de parole continue où l'unité élémentaire de reconnaissance est le phonème et où une segmentation préalable n'est pas nécessaire. Nous proposons un modèle de phonème et l'algorithme de reconnaissance associé, et une méthode de reconnaissance de phrase par localisation des centres syllabiques (chapitre 8). Nous présentons un système opérationel de compréhension du chinois parlé (chapitre 9). Cette partie montre l'utilisation des méthodes développées dans les parties I-IV et valide l'ensemble de nos études dans l'interprétation de signaux incertains.

CONTRIBUTIONS TO AUTOMATIC INTERPRETATION OF UNCERTAIN SIGNALS

ABSTRACT

Signal interpretation is a process of successive information transformation toward a form comprehensible to human or a machine. Because of signal uncertainty introduced during signal production and transmission, the interpretation process necessitates the use of artificial intelligence processing techniques of different abstraction levels.

In the framework of continuous speech understanding, we present our work on the four important aspects of interpretation: signal edition and modeling, signal-to-symbol conversion, structure analysis and A.I. system architecture.

In part I we present first a tool for signal edition and manipulation which permits signal parameter extraction and evaluation and assists knowledge acquisition about signal. This tool provides also an environment for simulating new signal processing algorithms by means of existing operators (chapter 2). We then study a non-stationary signal model for the estimation of the fundamental frequency of speech in which speech signal is modeled as a sequence of specified functions and the time-variability in period and in amplitude of the excitation signal is allowed. The parameters of the model is obtained by spectral matching in the time domain (chapter 3).

Part II concerns the problems of signal-to-symbol conversion in presence of uncertainty. We present a fuzzy classification method for fast access of large signal-symbol dictionary which allows to simultaneously assign a point to several classes (chapter 4) and a knowledge-based system, capable of processing imprecise and uncertain rules and of performing inexact reasoning, for signal-to-symbol conversion in the case of contextually deformed signals (chapter 5).

We propose in part III (chapter 6) a signal structure analyzer which operates both in bottom-up and top-down mode and develops in parallel, with a beam-search mechanism, solutions in order to propagate uncertainty. The analyzer is capable of constructing the syntactic structure from several confidence islands, under different search strategies.

An architecture of specialist society with multiple abstraction levels and knowledge sources is presented in part IV (chapter 7) which is based on the decomposition of the interpretation problem into sub-problems in successive conceptual levels and on the definition of a control mechanism explicitly related to these levels. Information exchange is assured by a data structure common to knowledge sources within the same level and by message passing between different levels.

In part V we illustrate a continuous speech understanding system in which the basic recognition unit is phoneme and a pre-segmentation is not necessary. We propose a phoneme model and the associated recognition algorithm and a method for sentence recognition based on syllable localization (chapter 8). We present an operational spoken Chinese understanding system (chapter 9). This part shows the use of the methods and techniques developed in parts I-IV and validates our studies on uncertain signal interpretation.

Table des matières

Chapitre 1

Introduction

L'homme a inventé les machines mécaniques pour libérer ses mains et agir sur la nature. L'homme a également inventé les machines calculatrices pour libérer son cerveau et traiter l'information. Cependant, l'homme n'est pas encore entièrement capable de construire le lien de la nature à la machine: la possibilités pour la machine de sentir, d'identifier, de reconnaître et d'interpréter les phénomènes du monde réel. L'absence de ce lien est un obstacle dans la construction de machines automatiques et autonomes capables de capter les phénomènes du monde réel et de réagir sur celui-ci.

L'intelligence artificielle est un domaine des études scientifiques et techniques sur les facultés produites par l'homme et non par la nature de connaître et de comprendre [Larousse 66]. L'étude sur la faculté des machines de reconnaître des objets réels est devenue une branche active en intelligence artificielle. C'est grâce à cette étude que l'intelligence artificielle est passée de l'époque rudimentaire, où seule la manipulation des symboles formels, ou la manipulation des objets génériques ou fabriqués par machine était considérée à une nouvelle époque où à la fois les symboles formels et les objets du monde réel sont manipulés.

L'intermédiaire inévitable entre l'homme et la nature est le signal. Le signal est l'observation physique d'une description codifiée d'une suite de changements d'états conventionnés, qui porte des messages sur un phénomène. Les états constituent un ensemble fini de symboles, appellé vocabulaire. Le changement d'états vérifie un ensemble de règles de composition, la syntaxe, plus d'autres lois respectées. Pour interpréter les messages, ces connaissances doivent être connues et exploitées également par le recepteur.

Le signal est soumis, avant d'arriver au recepteur, à une succession de transformations de support: la tranmission. Ces mécanismes de transformation introduisent différentes sortes de dégradation: déformation contextuelle, distorsion linéaire et non-linéaire, bruits. La conséquence de ces dégradations est qu'il n'existe plus une partition du signal telle que, pour chaque élément de la partition la projection signal-symbole soit bijective. Le signal qui ne vérifie pas la projection bijective est défini comme un signal incertain. A cause de cette incertitude,

- un symbole dans différents contextes peut produire des signaux différents. Nous appellons le phénomène qu'un symbole peut avoir des réalisations différentes *projection symbole-signal dispersée*;

- plusieurs symboles peuvent produire des signaux dont la différence entre chacun est pratiquement non significative. Nous entendons ceci par *projection symbole-signal superposée.*

En présence de l'incertitude, l'association d'un symbole unique à une portion de signal est non déterministe.

L'interprétation automatique du signal consiste à rendre le signal intelligible et à retrouver sa signification. Le processus d'interprétation consiste donc à transformer, à travers des niveaux d'abstraction successifs, l'information codée dans le message porté par le signal, en une forme explicite: soit un texte compréhensible par l'homme soit une fonction spécifiée et exécutable par la machine. L'information est une connaissance nouvelle, non-déductible. Un message est une codification d'information par un ensemble de symboles conventionnés. L'interprétation du signal est l'objectif ultime du traitement du signal. L'interprétation du signal incertain est d'identifier le plus possible est d'identifier le plus possible le signal et de découvrire sa structure interne sous la contrainte de l'ensemble des connaissances disponibles a priori.

Le problème se caractèrise par

- la quantité énorme de données. Il est donc indispensable de réduire cette quantité tout en concervant l'information transportée par le signal,
- les sources d'erreurs intrinsèques multiples dans les données. La variabilité du signal du fait du contexte, le bruit de fond, le manque de données, l'incapacité de détection de primitives et l'application des théories incomplètes et imprécises rendent l'interprétation non déterministe,
- le manque d'une approche systématique et unique pour trouver la solution. Il est impossible de construire ou il est impraticable d'utiliser un générateur des mouvements autorisés. Des connaissances humaines et des heuristiques sont utilisées dans le processus d'interprétation,
- l'existence dans chaque domaine des connaissances a priori qui ne sont pas transportées par le signal mais doivent néanmoins être utilisées de façon coopérative au cours du processus d'interprétation.

L'interprétation du signal de parole [Erman 80,Haton 85,Pierrel 87], d'images [Hanson 78, Nagao 79], de spectrogrammes de masse [Lindsay 80], du signal sonar [Maksym 83,Nii 86b] sont des exemples de l'application de l'interprétation du signal incertain.

L'incertitude du signal rend l'interprétation extrêmement difficile. Non seulement les propriétés statistiques, mais aussi le vocabulaire, la syntaxe et la sémantique du signal doivent être étudiés et utilisés. Le processus de l'interprétation exige une recherche pluridisciplinaire, de la production jusqu'à la compréhension du signal.

Nous présentons dans cette thèse nos travaux dans le domain de l'interprétation de signaux incertains. La présentation sera divisée en cinq parties. Les quatre premières parties sont consacrées aux aspects essentiels du processus d'interprétation: la manipulation, le traitement et la modélisation du signal, la conversion du signal en symboles en présence

de l'incertitude, le mécanisme d'analyse de structure des messages portés par le signal et l'organisation en niveaux d'abstraction multiples des sources de connaissances en interprétation. La cinquième partie présente un système d'interprétation de parole continue et montre l'utilisation des méthodes développées dans les premières parties. Dans les paragraphes qui suivent, nous introduisons brièvement ces cinq parties.

La première étape de l'interprétation est la paramétrisation ou la modélisation du signal. Elle consiste à décrire le signal sous une forme mathématique concise qui doit reproduire les caractéristiques du signal [Gong 85a]. La proposition d'un modèle pour un type de signal particulier nécessite l'acquisition de connaissances d'observation sur le signal et donc l'examen complet et le traitement de manière différente de grande quantité du signal. La partie I est divisée en deux chapitres. Le chapitre 2 présente un outil logiciel interactif, à fenêtrage multiple et commandé par menus, d'édition et de traitement du signal. Ce système permet également de composer de nouveaux algorithmes de traitement l'aide des opérations primitives existantes [Gong 85b] [Gong 85c]. Nous proposons dans le chapitre 3 un modèle de signal destinés à l'estimation de la fréquence fondamentale de la parole, où le signal est modélisé par une séquence d'une fonction spécifiée de façon dépendante du temps [Gong 87c]. En utilisant ce modèle, nous décrivons un nouvel algorithme d'estimation de la fréquence fondamentale qui a obtenu un résultat d'estimation presque sans erreur.

L'interprétation nécessite un raisonnement et le raisonnement est à son tour fondé sur l'information symbolique et non numérique. Il est donc indispensable d'effectuer une conversion d'information du signal vers des symboles. C'est la classification et la reconnaissance des objets – la partition optimale du signal et l'association de chaque partie à un ou plusieurs symboles. En présence de l'incertitude du signal, ce processus doit posséder la propriété d'être flou et incertain. Dans la partie II nous présentons notre travail sur le traitement de cette propriété. Le chapitre 4 décrit une méthode rapide d'accès au dictionnaire de conversion signal-symbole, fondée sur la notion de classification floue. La méthode permet de faire le compromis entre le temps et la qualité de recherche et, pour un dictionnaire de N mots, donne un nombre de comparaisons propotionnel à $\log_2 N$. L'incertitude du signal par la déformation contextuelle est difficile de modéliser par un modèle explicite et est insoluble par des méthodes statistiques et locales. Dans le chapitre 5 de la partie II, nous présentons un système à base de règles de production destiné à interpréter le résultat de preclassification signal-symbole, notamment à corriger l'effet de la déformation contextuelle [Gong 86a]. Dans ce système, la décision de la présence d'un symbole dans le signal dépend des symboles voisins et des descriptions en termes de quantités mesurables sur le signal. Ce système accepte des descriptions floues dans les règles et traite l'incertitude des connaissances dans le raisonnement. Il est utilisé pour interpréter des courbes des tons du chinois parlé [Gong 86b].

L'objectif de l'interprétation automatique est de donner une représentation structurelle et exploitable du signal. Il faut inférer la structure syntaxique à partir de l'information portée par le signal. A cause de l'incertitude, il est impossible de déterminer de manière fiable les éléments de base de l'analyse structurelle – les symboles terminaux et les séparateurs de symboles. Un mecanisme d'inférence spécialisé est donc exigé. La base de notre proposition est qu'en présence de l'incertitude les décisions ne doivent pas être mutuellement exclusives. Nous avons conçu un moteur d'inférence qui construit en parallèle plusieurs interprétations

plausibles, à partir de plusieurs regions de confiance du signal, sous une stratégie de recherche de solutions en faisceau. Afin d'utiliser efficacement l'information obtenue au cours de l'interprétation, ce moteur peut mener le raisonnement en mode mixte: contrôlé par les données et en mode dirigé par le modèle syntaxique du signal [Gong 87b]. Nous étudions ce mécanisme d'inférence dans le chapitre 6 de la partie III.

L'interprétation automatique du signal incertain est un processus complexe et demande une utilisation multiple et compliquée de différentes sources de connaissances qui ont participé à la production et à la transmission du signal et qui peuvent aider à sa compréhension. Des connaissances et des stratégies dans l'activité humaine d'interprétation et des techniques d'intelligence artificielle sont également utilisées pour augmenter la performance du système et pour réduire les calculs. Le contrôle, l'exploitation et la coopération des sources de connaissances sont donc des problèmes fondamentaux en interprétation. Nous proposons dans le chapitre 7 de la partie V une architecture de système d'interprétation dite "société de spécialistes", où l'ensemble de sources de connaissances est organisées en groupes dont chacun correspond à un niveau d'abstraction de concept [Gong 87a,Gong 88]. A l'intérieur d'un groupe, chaque source de connaissance prend indépendemment un ensemble de décisions dans son domaine de compétence. A chaque niveau d'abstraction, une structure de données commune permet la communication entre les sources de connaissances du même niveau et permet de construire les solutions finales de façon incrementale. L'échange d'information entre les niveaux est assuré par un mécanisme de courrier. L'interprétation se complète par l'exécution de groupe de sources de connaissances en phases multiples.

La parole humaine est un signal incertain typique. Nous abordons le problème de l'interprétation de la parole continue dans les chapitre 8 et 9 de la partie V. Le problème peut s'énnoncer de la façon suivante:

> Etant donnée une image acoustique du signal de la parole, inférer la séquence de symboles linguistiques produisant cette image et le sens transmis par la parole, en disposant d'un ensemble des connaissances a priori qui contraignent la variation du signal.

Nous étudions plus particulièrement le problème de l'organisation de sources de connaissances de différents niveaux d'abstraction dans le processus de communication orale, de stratégie de reconnaissance, de modélisation de l'unité phonémique, le problème de représentation de connaissances phonologiques et de localisation des unités élémentaires de reconnaissance. Nous avons construit un système de compréhension de la parole qui n'exige pas une segmentation préalable, utilise le phonème comme unité élementaire de reconnaissance et permet donc l'exploitation des connaissance phonétiques et phonologiques. Il s'adapte facilement à une nouvelle application sans apprentissage. Utilisant actuellement une grammaire du type contexte-libre comportant 250 règles et 250 terminaux, le système a obtenu un taux de reconnaissance au niveau de la phrase supérieur à 90% [Gong 87b]. Dans cette partie nous montrons l'utilisation de l'ensemble de méthodes et techniques présentées dans les parties I-IV.

Partie I

TRAITEMENT ET MODELISATION DE SIGNAL

Chapitre 2

Outils Interactifs de Traitement du Signal

L'interprétation des signaux utilise des connaissances diverses, en particulier des connaissances sur la correspondance signal-symbole. L'acquisition de ces connaissances nécessite le traitement et l'examen de grande quantité de données. Nous présentons le système interactif de manipulation de signaux ASSIA – **Système pour Analyse de Signal et Simulation Interactive d'Algorithmes** – [1] *permettant de traiter le signal et d'extraire de l'information, d'observer et d'éditer le signal et les résultats de traitements spécifiques et enfin de développer et de simuler des nouveaux algorithmes de traitements à l'aide des opérateurs primitifs. La manipulation du signal est facilitée par des buffers et des fénêtres multiples simultanément présents sur l'écran. Les commandes peuvent être chaînées et mémorisées sous forme de macro-commandes. Les opérateurs plus complexes peuvent être composés et programmés à partir des opérateurs primitifs. Nous discutons la représentation des objets dans le système et les composants principaux: l'éditeur (menus, fenêtres) de signaux, l'analyseur de signaux, simulation d'algorithmes. Nous dévcloppons dans ce chapitre les points suivants: Fenêtres et tempons multiples, Systèmes de menus, Structures de données, Critères de représentation paramétrique, Traitements de base, et Enchaînement de traitements.*

1 Introduction

En interprétation du signal, il est essentiel de connaître ce que les événements dans le signal signifient, c'est à dire qu'il faut les comprendre. Pour cela, il faut d'abord pouvoir les décrire. A travers l'œil, l'homme est capable de capturer de petites nuances dans son environnement mais cela est passé entièrement inconsciemment. La visualisation des signaux n'a pas encore une histoire suffisamment longue pour que l'on dispose d'un vocabulaire assez riche de description de l'écran. Ce manque de vocabulaire est un obstacle qui empêche l'acquisition de connaissances nécessaires en interprétation du signal. Par exemple on a commencé à utiliser des outils graphiques pour examiner la parole il y a 50 ans [Rabiner 78b] et tout ce dont on pourrait disposer pour décrire la parole est le niveau de gris, la forme, la

[1] Ce projet a été sélectionné par l'Agence de l'Informatique dans le cadre du *concours SM90*.

couleur plus quelques termes élementaires tels que fort, faible, montant, descendant, haut, bas, bruit, etc.

L'étude d'un signal se repose à la fois sur le phénomène physique et la théorie. Par conséquent, la recherche dans le domaine du traitement du signal est un processus ayant deux aspects liés:

- Le premier aspect part de phénomènes concrets tel que la paramétrisation de la parole. Il s'agit alors de traiter le signal d'un grand nombre de manières différentes. L'objectif de ces traitements est d'extaire de l'information transportée par le signal ou de modifier le signal lui-même.

- Le deuxième aspect part de constructions abstraites telles que le développement et l'implantation de nouveaux algorithmes. Cet aspect consiste à définir et réaliser des opérations sur les phénomènes. On étudie la structure interne des traitements et on définit de nouveaux signaux à l'aide de ces traitements. Par ailleurs, il est nécessaire d'établir la liaison entre l'utilisateur et le système.

- Ceci constitue un troisième aspect important, la visualisation des phénomènes.

Nous avons donc besoin d'un système de manipulation de signaux qui doit accomplir les trois tâches suivantes:

- La première consiste à fournir un outil interactif pour la recherche en manipulation et traitement de signaux, spécialisé en analyse et reconnaissance de signaux. Cela demande la possibilité de visualiser, d'éditer et d'analyser des signaux.

- La deuxième est de travailler à un niveau de programmation relativement élevé par rapport à un langage de programmation standard, ce qui nécessite la manipulation des objets relatifs au signal et non directement les entiers, réels et/ou des tableaux.

- La troisième doit répondre à la question suivante: Face à la rapilité du développement des algorithmes de traitement du signal, comment organiser le système de façon que de nouveaux algorithmes puissent être pris en compte? Nous devons donc constuire un système évolutif, avec possiblité de définir de nouveaux algorithmes de traitement à partir de primitives existantes.

2 Présentation générale

La recherche et les développements en traitement automatique de la parole ou de tout autre type de signaux physiques nécessitent des outils logiciels et matériels évolués et interactifs, permettant à l'utilisateur de mettre au point de nouvelles méthodes et de visualiser au mieux les phénomènes, sans être forcément un informaticien chevronné. Cependant, la réalisation de ces outils n'est pas aisée. Les problèmes des tâches à effectuer, de l'organisation du système, de l'interface entre l'ordinateur et l'utilisateur, des objets à représenter et de l'adaptation à la fois à des utilisateur non-informaticiens et informaticiens soulèvent

beaucoup de difficultés. En plus, le fait que de nombreux algorithmes de traitement du signal, et de la parole en particulier, aient été proposés ces derniéres années entraîne qu'aucun système figé, incapable d'évoluer, ne peut être totalement satisfaisant pour des applications futures.

Différents travaux ont été menés pour tenter de résoudre ces problèmes. Dans le système OBSERVATEUR [Sanchez 84], les différentes procédures de traitement sont considérées comme des actions et sont activées par un multiplexeur à la demande de l'utilisateur. ILS (Interactive Laboratory System) [Technology 84] a été conçu plus particulièrement pour l'analyse, la synthèse et la transmission de la parole. Il peut être considéré comme une collection de programmes de traitement du signal; l'organisation du système et les structures de données ne permettent pas facilement d'enchaîner et de recombiner des programmes. Un système plus formel, ISP (Integrated Signal Processing System) [Kopec 84], est fondé sur la création et la manipulation d'"objets signal". L'une des caractéristiques importantes de ISP est que c'est un système initialement conçu pour traiter des signaux individualisés et dans lequel les signaux à manipuler sont réferencés par une pile. Par conséquent le système est relativement limité du point de vue de la sélection des signaux en cours d'examen et du traitement des grands corpus de signaux.

Nous considérons qu'un système capable de fournir un environnement agréable pour la recherche doit intégrer les aspects suivants: manipulation, visualisation, modification et étiquetage de différents signaux, traitement numérique de signaux individuels et de grand corpus pour créer des nouveaux signaux, et enfin développement (conception, simulation et évaluation) de nouveaux algorithmes. Partant de cela nous avons défini et construit un systeme pour *l'Analyse de Signal et la Simulation Interactive d'Algorithmes* (ASSIA), système logiciel destiné à l'édition, au traitement numérique et statistique du signal et des données et au développement d'algorithmes de traitement du signal. La conception du système est orientée vers la reconnaissance acoustico-phonétique de la parole. Ce système est écrit en *C* et est implanté sur *SM*90 sous le système d'exploitation Unix. Le poste de travail est construit autour d'un écran bit map avec clavier et souris.

Le principe de base du système consiste à définir des types liés aux signaux, et des opérations associées pour modèliser le signal, le graphisme et les algorithmes de traitement. Le système est évolutif et permet à l'utilisateur de définir des enchaînements d'opérateurs ou des opérateurs complexes à partir d'opérateurs existants ou définis par l'utilisateur. Pour cela nous avons tiré profit des possibilités du système Unix, et en particulier du Shell. L'interface avec l'utilisateur a été particulièrement soigné. Il utilise les possibilités fournies par les techniques de multi-fenêtrage sur écran et de menus à plusieurs niveaux. L'utilisateur peut aussi définir des macro-commandes qui amènent le système dans un état de fonctionnement voulu.

Dans ce chapitre, nous allons décrire les quatre parties du système:

Editeur de signal

C'est un manipulateur de signal à fenêtres et à buffers multiples. Comme l'éditeur de texte Emacs [Gosling 81], l'éditeur fournit la possibilité d'effectuer les opérations d'édition telles que ouverture et fermeture d'un buffer, déplacement d'une fenêtre, suppression et

insersion, recherche, etc.

Analyseur de signal

L'aspect du traitement du signal est modélisé par l'application successive d'opérateurs à l'objet signal. Les signaux sont directement référencés par leur images sur l'écran à travers la souris. Les opérateurs sont présentés sous forme de menus.

Interface utilisateur

Le système est guidé par des menus hiérarchiques. L'utilisateur a la possibilité de définir des macro-commandes pour simplifier l'exécution d'une liste de commandes et pour créer des commandes plus complexes.

Simulation d'algorithmes

Les opérateurs dans l'éditeur de l'analyseur sont compilés séparément sous forme des programmes ayant une structure de données entrée-sortie compatible. De nouveaux opérateurs définissant un nouvel algorithme de traitement peuvent être construits à l'aide des opérateurs existant ou spécifiés par l'utilisateur. Le processus d'implantation, de modification et de test d'un algorithme est ainsi simplifié grâce aux opérateurs primitifs et à l'introduction de l'objet signal.

—

Après avoir exposé des structures de données principales, quatre sous-systèmes d'ASSIA: l'éditeur du signal, l'analyseur du signal, l'interface avec l'utilisateur et la simulation d'algorithmes seront présentés. Nous illusterons notre propos à l'aide d'exemples pratiques d'utilisation.

3 Représentation des objets

Avant de construire un système de manipulation et d'analyse de signaux, il est nécessaire d'étudier comment définir, représenter et organiser les différents types d'informations à traiter. Les objets ayant des propriétés communes peuvent être regroupés sous une même description générale. Cela permet d'avoir un modèle conceptuellement unique et d'éviter la programmation complètement empirique. Deux types d'objets sont intensivement utilisés dans notre système – *signal* et *buffer*, un signal étant un sous-objet dans un buffer.

Au moment de la définition de la structure de données "signal" et "buffer", nous avons été amenés à trouver des structures qui vérifient les critères suivants:

- *Compréhensible:* Les structures doivent être conceptuellement significatives et doivent être compréhensibles; Il faut regrouper de façon naturelle et consistante les objets et les opérations;

signal							
point	vector				matrix		
-	real	imag	n	V-pt	m	n	M-pt

Table 2.1: objet signal

univers			
file	name		block
-	work	real	-

Table 2.2: objet univers

- *Simple:* Il faut tenir compte de la complexité des opérations qui manipulent ces objets et trouver un compromis entre la redondance de stockage et l'efficacité de calcul;

- *Général:* Les structures utilisées doivent être suffisamment générales pour englober toutes les informations manipulées;

- *Extensible:* Il est nécessaire de prévoir des extensions au système dans les futurs développements.

Les objets de base introduits dans ASSIA sont les types signal et buffer. Ces objets, avec les opérations associées, permettent de résoudre notre problème plus efficacement et systématiquement. Ce sont des représentations générales des données ou des informations graphiques. Les objets sont définis hiérarchiquement par des objets de structure plus simple et par des pointeurs sur des structures, le plus bas niveau étant constitué des données scalaires. Pour le traitement du signal, des points, des vecteurs complexes et des matrices sont représentés par l'objet "signal".

Un fichier de signaux est représenté par l'objet "univers" comprenant le nom d'un fichier de données, la longeur de l'enregistrement, des pointeurs sur le fichier et dans le fichier, etc.

En ce qui concerne le graphisme nous avons utilisé l'objet "buffer" auquel est associée une image sur l'écran. Il se compose d'un certain nombre de sous-objets tels que "viewport" (la position et la taille de l'image du buffer sur l'écran), "markers" (les données définissant la marque, le curseur et la fenêtre), "display" (l'échelle et l'origine de l'image), "operations" (les opérations déjà appliquées sur l'objet), "state" (le type, le numéro et l'état de modification du buffer), etc. L'objet "buffer" peut être schématisé comme indiqué dans les tableaux suivants.

buffer						
state	viewport	univers	display	markers	signal	operation

Table 2.3: objet buffer

state			
time	No	type	state

Table 2.4: objet state

viewpoint				
position		color	dimension	
x	y	-	wide	high

Table 2.5: objet viewpoint

display			
scale	normalize	polarity	amplitude

Table 2.6: objet display

markers					
mark		cursor		frame	
-	-	-	-	-	-

Table 2.7: objet markers

operations	
opt-names	opt-state

Table 2.8: objet operations

4 Editeur de signal

4.1 Introduction

L'éditeur de signaux est un sous-système d'ASSIA qui a pour but d'assister l'utilisateur pour manipuler ou éditer différents signaux grâce aux moyens graphiques du système. Le fonctionnement est semblable à un éditeur de textes mais l'objet manipulé est le signal et non une chaîne de caractères. Le rôle essentiel de notre éditeur sur un signal consiste en la création et la suppression, la visualisation à travers une fenêtre d'observation mobile, la modification temporelle (par exemple, suppression et insertion d'une zone sur un signal) et la consultation permettant de calculer des paramètres et d'extraire une information à partir du signal. L'objet géré par l'éditeur du système est une liste de l'objet *buffer*.

Pour profiter des possibilités de l'écran bit map nous avons défini un système de multi-buffer et multi-fenêtrage qui permet la coexistence de plusieurs images de "buffer" sur l'écran. Chaque image peut être créée et effacée indépendamment des autres afin d'utiliser efficacement la ressource d'affichage. Les images peuvent être créées en superposition et les parties couvertes se recupèrent rapidement par la souris à la demande de l'utilisateur. La fenêtre de visualisation sur l'image d'un "buffer" est translatable dans tous les sens pour que l'utilisateur puisse mieux observer les phénomènes portés par le signal.

4.2 Buffer

Un buffer est un espace de travail temporel d'ASSIA dans lequel sont stockés le signal en cours d'édition et son état de modification. ASSIA peut gérer plusieurs buffers dans une session de travail. Chaque buffer a un nom unique et contient le fichier du signal que l'on est en train d'éditer. Un buffer peut être créé ou supprimé dynamiquement. Un buffer est visible à travers une fenêtre de visualisation qui lui est associée.

4.3 Fenêtre

Une fenêtre est un rectangle sur l'écran définissant la zone de représentation graphique du buffer d'un signal [Robson 86]. Plusieurs fenêtres d'édition de buffers différents ou non peuvent exister simultanément sur l'écran. Les fenêtres peuvent se superposer l'une sur l'autre. La partie cachée d'une fenêtre peut être restituée facilement par une simple sélection avec la souris. Les fenêtres permettent à l'utilisateur d'effectuer des opérations directement sur l'image du signal. Il est possible de modifier avec la souris l'emplacement et la taille d'une fenêtre même après sa création. Une fenêtre peut être fermée si on n'en a pas besoin.

Il est possible d'afficher une fenêtre et son contenu de deux manières:

- La première est de créer l'image et la stocker en mémoire jusqu'au moment de sa destruction. A chaque fois que l'on a besoin d'afficher l'image, on copie simplement l'image en mémoire sur l'écran. Cette méthode nécessite un espace de mémoire important et une grande quantité d'accès à la memoire pour l'affichage. Traditionnellement, elle très employée à cause de la rapilité de reproduction de l'image.

- La deuxième mémorise le mode de création de l'image et à chaque affichage on réapplique le processus de création pour obtenir à nouveau l'image. Cette technique ne mémorise qu'une structure définissant la manière dont on fabrique l'image.

Dans ASSIA les graphismes traités sont essentiellement des lignes et des courbes, les cadres de fenêtre et les traces de spectre d'un signal par exemple. Comparé à une image, ce type de graphisme donne un faible rapport du nombre de points utilisés sur le nombre total de points utilisables sur l'écran. Donc le principe d'affichage des fenêtres par le stockage point par point d'une image en mémoire devient très inefficace. Nous avons implanté les fenêtres avec la seconde méthode pour économiser la mémoire.

5 Opérations de l'éditeur

Nous présentons les opérations associées à un "buffer" dans l'éditeur. Elles sont réparties en cinq catégories détaillées dans les paragraphes suivents.

5.1 Gestion des buffers

- *OPEN:* ouvre un buffer et crée une fenêtre d'observation du signal pour que l'utilisateur puisse avoir accès au signal par visualisation et par les outils du système.
- *CLOSE:* ferme un buffer existant et efface la fenêtre associée permettant de libérer la partie de l'écran occupée.
- *KILL:* supprime une zone de signal dans un buffer, délimitée par une marque précédemment posée et le curseur. Le signal supprimé est mémorisé et peut être rappelé par la commande *INSERT* dans les fénêtres visibles à l'écran.
- *INSERT:* insère une zone de signal du buffer temporel dans un buffer, à l'endroit où se trouve le curseur. Le contenu du buffer temporel ne change pas après l'insertion.
- *CHANGE:* change le fichier de signal que l'on est en train d'éditer.
- *COLLECT:* collectionne des morceaux de signal des buffers actifs; les résultats sont concanénés et mémorisés un buffer.
- *CREAT:* crée un signal à partir du buffer de stockage des signaux collectionnés et ouvre un buffer et la fenêtre associée.
- *SAVE:* sauvegarde un buffer modifié. Les modifications peuvent être des suppressions, insertions et créations.
- *SHOW:* affiche l'état (l'ensemble de variables) des buffers et des fenêtres actifs.
- *VALUE:* calcule une valeur dans le signal à l'instant indiqué par le curseur.
- *REFRESH:* detruit tous les buffers et efface les fenêtres associées sur l'écran.

5.2 Gestion des fenêtres

- *MOVE:* déplace une fenêtre d'observation sur l'écran. Le nouvel emplacement et sa dimension sont rédéfinis à l'aide de la souris.
- *NEXT:* translate la fenêtre d'observation en avant. Le signal de la fenêtre suivante est affiché.
- *PREVIOUS:* translate la fenêtre d'observation en arrière. Le signal de la fenêtre précédente est affiché.
- *SHIFT:* déplace un nombre de blocs de signal dans une fenêtre d'observation. La direction du déplacement peut être avant ou arrière selon le bouton appuyé.
- *SCROLL:* translate de façon continue et graduelle une fenêtre en avant ou en arrière. La direction de translation est donnée par les boutons de la souris. La vitesse de translation est contrôlable.
- *COLOR:* inverse les couleurs noir ou blanc d'une fenêtre ("inverse vidéo").

5.3 Gestion de marques

- *SET-MARK:* positionne une marque sur le signal. La marque sera affichée si sa position est dans la fenêtre de visualisation.
- *JUMP:* déplace une fenêtre de façon que la marque prédéfinie soit positionnée au milieu de cette fenêtre. Cette commande est utilisée pour retrouver un point déjà repéré dans le signal.
- *ADJUST:* ajuste la position d'une marque de façon précise.
- *WINDOW:* définit une zone sur le signal. Cette zone peut être utilisée pour effectuer des analyses du signal.
- *TIME:* calcule le temps entre une marque et la position du curseur.

5.4 Gestion d'étiquettes

- *LABEL:* définit ou supprime une étiquette dans le signal. Si une étiquette existe déjà à l'endroit demandé, l'ancienne étiquette sera détruite. Une confirmation sera activée dans ce cas. L'information sur l'étiquetage du fichier de signal "f" est mémorisée dans le fichier "f.lab" et est réutilisable par les traitements ultérieurs. Cette commande permet de réaliser l'association d'une zone du signal à un symbole.
- *SEARCH:* recherche une étiquette et déplace la fenêtre pour visualiser le signal étiqueté. La fenêtre sera réaffichée de telle manière que l'étiquette cherchée soit positionnée au centre de la fenêtre. Cette recherche peut s'effectuer dans les deux directions. La liste d'étiquettes sera imprimée si le bouton du milieu est appuyé.

5.5 Gestion du système

- !<command>: exécute une commande d'Unix et puis revient au programme. Le contenu de l'écran est sauvegardé avant l'exécution et sera récupéré après.
- *PROCESS:* lance une opération en paralèlle en arrière plan.
- *HELP:* affiche le manuel d'utilisation en ligne.
- *EXIT:* quitte le menu de l'éditeur.

6 Analyseur de signal

6.1 Introduction

L'analyseur est un autre sous-système d'ASSIA. Il est chargé d'effectuer des traitements numériques sur les zones du signal définies par l'utilisateur. Le résultat est associé à un affichage sur l'écran qui est une image s'il s'agit d'un vecteur ou alors un texte s'il s'agit d'un point.

Dans un grand nombre de problèmes concernant le traitement du signal et la reconnaissance des formes, l'hypothèse de la stationnarité est souvent faite. Cela conduit à la notion d'analyse à court terme, qui s'attache à traiter le signal à l'intérieur d'une fenêtre de largeur convenablement fixée. Le signal à l'extérieur de la fenêtre est supposé avoir la même propriété que le signal isolé par la fenêtre. Le signal choisi contient la plupart des informations sur le signal original et en constitue donc une bonne représentation. Dans ASSIA, la représentation est modèlisée par l'introduction du type "signal" défini par les opérations qui lui sont associées.

Le traitement d'un signal est considéré dans notre analyseur comme l'application successive d'opérations sur l'objet "signal". Un algorithme de traitement est donc modélisé par un certain nombre d'opérateurs primitifs. Les opérations sur un objet sont fermées au sens que le résultat est aussi un objet. Les opérateurs sont présentés dans le menu tandis que les objets sont directement référencés par leur image sur l'écran et accessibles à l'aide de la souris.

Certaines combinaisons de type d'objets créés et d'opérations fournies par l'analyseur n'ont pas de sens physique. Il faut pouvoir les distinguer et les interdire. Le contrôle de compatibilité des types d'objets et de la vérification des opérateurs applicables se fait par un mécanisme simple à base de connaissances. Ce mécanisme dispose de règles préformulées permettant de déterminer la légalité d'une opération sur un objet en fonction des opérations précédentes, de l'objet initial et de l'état du système.

Dans l'analyseur, un signal est représenté comme un vecteur complexe comprenant la partie réelle et la partie imaginaire. Cette représentation générale fournit la possibilité de travailler sur une structure de signal uniforme et de pouvoir enchaîner les opérations sur les résultats des autres opérations.

Le contenu du traitement du signal est:

- représentation d'un signal,
- restitution ou séparation d'un signal,
- transformation d'un signal,
- estimation des paramètres caractéristiques d'un signal.

Nous avons trois catégories d'opérateurs:

- La première catégorie sont des opérateurs de transformation orthogonale dont le résultat conserve la même information que la donnée de départ et dont la donnée peut être restituée entièrement à partir du résultat.
- La deuxième catégorie d'opérateurs est celle du type filtrage qui réalise des modifications sur le signal en spectre ou en amplitude.
- La dernière catégorie d'opérateurs est celle des opérateurs d'extraction d'information permettant de caractériser le signal par un vecteur de dimension réduite.

Notre analyseur est adapté à l'analyse et la reconnaissance de la parole mais l'ensemble de primitives fournies constituent une base de traitement du signal générale. Nous présenton maintenant les principales techniques de traitement de signal utilisées pour la représentation paramètrique en reconnaissance de parole. Ce sont le filtrage linéaire, l'analyse de la transformée de Fourier, l'analyse homomorphique (méthode cepstrale) et l'analyse par prédiction linéaire.

6.2 Transformée de Fourier

La transformée de Fourier convertit l'information du signal du domaine temporel vers le domaine fréquenciel. La représentation du signal dans le domaine fréquentiel permet de montrer certaines propriétés du signal dont l'examen dans le domaine temporel est difficile. Cette transformation représente le signal par la somme pondérée de composants à différentes fréquences. Dans ce cas une fonction de forme compliquée peut s'exprimer en une combinaison linéaire de fonctions de base qui ont des propriétés bien connues et qui sont relativement faciles à manipuler et à étudier.

La transformée de Fourier pour le signal discrèt $x(n)$ est définie comme

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

$x(n)$ peut être restitué par

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$$

En périodisant $x(n)$ tel que $x(n) = x(n+N)$, $x(n)$ peut s'exprimer en somme et non en intégrale. Cela donne la définition de la transformée de Fourier discrète (T.F.D.):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jkn2\pi/N}$$

et la transformée inverse (T.F.D.I.):

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{jkn2\pi/N}$$

où k est la fréquence discrete. $k = 0$ correspond à la composante continue du signal. T.F.D. et T.F.D.I. sont symétriques. Le calcul direct d'une T.F.D. est est de complexité $O(N^2)$ et peut être réduit par des méthodes efficaces en $O(N \log_2 N)$ [Cooley 65,Singleton 69,Gong 83].

Soit $W_N = e^{-j2\pi/N}$ le facteur de phase. Le formule de la transformée de Fourier devient alors

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}.$$

La plupart des techniques de transformée de Fourier rapide reposent sur les deux propriétés du facteur W_N suivantes [Cooley 65,Rabiner 79]:

symétrie: $W_N^k = -W_N^{k+N/2}$

périodicité: $W_N^k = -W_N^{N+k}$

Le spectre $X(\omega)$ du signal $x(n)$ est en général une fonction complexe:

$$X(\omega) = Re[X(\omega)] + jIm[X(\omega)].$$

qui peut être également représentée sous forme de module et d'argument:

$$X(\omega) = |X(\omega)|e^{jarg[X(\omega)]}$$

où $|X(\omega)|$ est appelé spectre d'amplitude ou spectre et $arg[X(\omega)]$ spectre de phase. $|X(\omega)|^2$ est souvent appelé spectre d'énergie.

6.3 Filtre linéaire

Soit un système S agissant sur signal d'entrée $x(n)$ (excitation) et produisant une réponse $y(n)$ à la sortie: $y(n) = S[x(n)]$.

- Si le système est linéaire, c'est à dire s'il obéit à la loi de superposition, i.e: $S[A\,x_1(n) + B\,x_2(n)] = A\,S[x_1(n)] + B\,S[x_2(n)]$,
- et invariant du temps, c'est à dire que les paramètres du système sont indépendants du temps, i.e: si $x(n) \rightarrow y(n)$ alors $x(n - n_0) \rightarrow y(n - n_0)$

alors le système est complétement défini par sa réponse à une implusion de Dirac $\delta(n)$, ou sa réponse impulsionnelle $h(n) = S[\delta(n)]$.

Un filtre est un système linéaire discret invariant. Dans la sortie $Y(\omega)$ du filtre $H(\omega)$, les composantes des diverses fréquences du signal d'entrée $X(\omega)$ ont été modifiées par l'action du filtre sur la répartition fréquentielle de l'excitation. On distingue deux grands types de

filtres. Si la réponse impulsionnelle est finie un certain temps après l'excitation, le filtre est un *filtre à réponse impulsionnelle finie (FIR)* sinon un *filtre à réponse impulsionnelle infinie (IIR)*.

La fonction de transfert $H(z)$ d'un filtre peut s'écrire sous la forme d'un rapport de deux polynômes en z:

$$H(z) = \frac{\sum_{q=0}^{M} b_q z^{-q}}{1 - \sum_{p=1}^{N} a_p z^{-p}} = \frac{Q(z)}{P(z)}.$$

Les pôles et les zéros sont respectivement les racines de $P(z)$ et de $Q(z)$. Si $H(z)$ a M zéros z_m et N pôles p_n alors $X(z)$ peut s'exprimer sous la forme:

$$X(z) = A\frac{\prod_{m=1}^{M}(z - z_m)}{\prod_{n=1}^{N}(z - z_n)}.$$

Trois techniques sont équivalentes pour la réalisation d'un filtre linéaire:

- La convolution.
- La transformée Fourier.

$$X(\omega) = \mathcal{F}[x(n)]$$
$$Y(\omega) = H(\omega) \times X(\omega)$$
$$y(n) = \mathcal{F}^{-1}[Y(\omega)].$$

- L'équation différentielle. En appliquant la transformée en z inverse sur la fonction de transfert d'un filtre $H(z)$, on obtient une équation différentielle qui lie, dans le domaine temporel, la sortie courante $y(n)$ avec les sorties précédentes, les entrées et les coefficients du filtre:

$$y(n) = \sum_{p=1}^{N} a_p y(n - p) + \sum_{q=0}^{M} b_q x(n - q).$$

6.4 Analyse homomorphique

Dans les systèmes linéaires (obéissant au principe de superposition), les composantes sont combinées de façon additive. Les composantes qui occupent différentes bandes de fréquences sont séparables par un filtrage linéaire. Mais comment isoler des composantes dans un système non-linéaire tel qu'un système

- **multiplicatif:** $x(n) = x_1(n)x_2(n)$ ou
- **convolutif:** $x(n) = x_1(n) * x_2(n)$?

La solution est de convertir, en utilisant des transformations non-linéaires bijectives soigneusement définies, des opérations non-additives en opérations additives [Oppenheim 75,Oppenheim 69a, Kunt 80].

système homomorphique multiplicatif

Pour les signaux multipliés, on peut utiliser l'opération logarithmique pour convertir la loi multiplicative en loi additive. Soit $x(n)$ un signal composé de $x_1(n)$ et $x_2(n)$ par une multiplication:

$$x(n) = a_1 x_1(n) \times a_2 x_2(n)$$

Appliquer l'opération non-linéaire logarithme:

$$\hat{x}(n) = ln[x(n)] = a_1 ln[x_1(n)] + a_2 ln[x_2(n)] = a_1 \hat{x}_1(n) + a_2 \hat{x}_2(n)$$

$\hat{x}(n)$ est la combinaison linéaire de $\hat{x}_1$ et $\hat{x}_2$. Selon la répartition des composantes dans $\hat{x}_1$ et $\hat{x}_2$, un filtrage linéaire peut être utilisé pour séparer l'un ou l'autre. Après le filtrage, on peut récupérer le signal isolé par une opération exponentielle:

$$x(n) = exp[\hat{x}(n)]$$

système homomorphique convolutif

Rappelons que la transformée de Fourier convertit la relation convolutive entre composantes en une relation multiplicative. Les signaux multipliés peuvent être séparés par un système homomorphique multiplicatif. A la fin du traitement homomorphique, on applique une transformée de Fourier inverse pour revenir au domaine temporel. Pour faciliter la conception du système linéaire dans le système multiplicatif, le premier est remplacé par un autre système, toujours linéaire, de trois composantes successives: une T.F. inverse, un filtre et une T.F.. Ce qui assure que le signal traité dans le filtre est un signal temporel. Ce signal est généralement appellé *cepstre*. Le processus de la construction d'un système homomorphique convolutif est le suivant [Oppenheim 75,Oppenheim 69b]:

Signal original: $x(n) = x_1(n) * x_2(n)$ (temps).

T.F.: Transformée de Fourier, le signal à traiter $x(n)$ dans le domaine temporel est passé dans le domaine fréquentiel $X(\omega)$. Une conversion de convolution à multiplication est effectuée.

$$X(\omega) = X_1(\omega) X_2(\omega).$$

ln: Logarithme, à la sortie les composantes du signal sont combinées de façon additive.

$$\hat{X}(\omega) = ln[X(\omega)] = ln[X_1(\omega)] + ln[X_2(\omega)] = \hat{X}_1(\omega) + \hat{X}_2(\omega).$$

T.F.I.: Transformée de Fourier inverse. Le signal est repassé dans le domaine temporel, mais il reste toujours additif.

$$\hat{x}(n) = \hat{x}_1(n) + \hat{x}_2(n) \quad (cepstre, temps)$$

L: Système linéaire, les traitements (filtrage en général) peuvent être effectués ici. Si $\hat{x}_1(n)$ et $\hat{x}_2(n)$ sont séparables, on peut éliminer $\hat{x}_2(n)$.

$$\hat{y}(n) = \hat{x}_1(n)$$

T.F.: Transformée de Fourier. Le signal est passé dans le domaine fréquentiel.

$$\hat{Y}(\omega) = \hat{X}_1(\omega).$$

Exp: Opération exponentielle, la loi de combinaison de composantes repasse en multiplication.

$$Y(\omega) = X_1(\omega)$$

T.F.I.: Transformée de Fourier inverse, le signal traité est retransformé du domaine spectral au domaine temporel.

$$y(n) = x_1(n).$$

Le calcul pour l'analyse homomorphique est lourd: plusieurs T.F. et opérations logarithmiques. Dans [Markel 71] est détaillé l'utilisation de la technique cepstrale sur des exemples.

6.5 Prédiction linéaire

Principe

Un signal transportant des messages n'est jamais un signal complètement aléatoire: il existe une corrélation entre les échantillons successifs. La technique de prédiction linéaire utilise cette corrélation pour réduire la quantité de données tout en conservant l'information transportée. L'idée de base est la suivante:

- $x(n)$ peut s'exprimer en la somme d'une combinaison linéaire de son passé $\{x(n-1), x(n-2), ...\}$ et d'une erreur.
- On règle les coefficients de cette combinaison pour que l'énergie de de la séquence d'erreurs $e(n)$ qui est la différence entre le signal réel et le signal prédit dans un intervalle soit minimale. Ces coefficients sont une représentation de la parole dans l'intervalle.
- Ce modèle est cohérent avec le modèle de production de la parole: source d'excitation (périodique ou aléatoire) + conduit vocal → parole.

Formulation de la prédiction linéaire

On suppose que la sortie d'un système linéaire vérifie [Markoul 73]

$$s(n) = \sum_{k=1}^{p} \alpha_k s(n-k) + N(n) \tag{2.1}$$

où $N(n)$ est une séquence de bruit blanc. Un prédicteur avec coefficients $\{a_k\}$ est défini comme le système dont la sortie est:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k)$$

L'erreur de prédiction s'écrit alors:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad (2.2)$$

Elle est la sortie du système:

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

Comparons 2.1 avec 2.2, si le signal obéit exactement à l'équation 2.1 et si $\alpha_k = a_k$, alors:

$$e(n) = N(n)$$

Nous avons:

$$H(z) = \frac{G}{A(z)}$$

Le filtre de l'erreur de prédiction $A(z)$ est donc le filtre inverse du système $H(z)$. L'énergie de l'erreur est:

$$E_n = \sum_m e_n^2(m) = \sum_m [s_n(m) - \hat{s}_n(m)]^2$$

En minimisant E_n par rapport à $\{ a_k \}$:

$$\frac{\partial E_n}{\partial a_i} = 0 \qquad (i = 1, 2, ..., p)$$

on obtient:

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} a_k \sum_m s_n(m-i)s_n(m-k) \qquad (1 \le i \le p).$$

posons:

$$\Phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k)$$

nous avons:

$$\sum_{k=1}^{p} a_k \Phi_n(i,k) = \Phi_n(i,0) \qquad (i = 1, 2, ..., p).$$

L'intervalle de la sommation de m n'a pas été précisé dans la formulation. En fait, deux méthodes se distinguent par la façon de définir cet intervalle:

1. La méthode d'autocorrélation. Le signal est multiplié par une fenêtre et devient:

   $$S_n(m) = 0 \qquad m \le 0 \text{ et } m \ge N$$

   l'énergie de l'erreur se calcule alors:

   $$E_n = \sum_{m=0}^{N-1+p} e_n^2(m).$$

   la forme matricielle de $\Phi_n(i,k)$ est dans ce cas une matrice du type Toeplitz.

2. La méthode de covariance. On limite l'intervalle à [0..N-1]:

   $$E_n(m) = \sum_{m=0}^{N-1} e_n^2(m)$$

   ici on trouve une matrice de covariance.

Il existe des solutions efficaces utilisant les propriétés mathématiques des matrices du type Toeplitz (Durbin) ou covariance (Cholesky) [Rabiner 78b]. En général, le nombre de multiplications est $N \times p$ (corrélation) + p^2 (solution de matrice). L'extraction des paramètres se fait entièrement dans le domaine temporel et est rapide.

Paramètres issus de la prédiction linéaire

Les coefficients de prédiction linéaire peuvent être utilisés pour calculer d'autres paramètres de la parole.

Spectre: Les coefficients a_k constituant un filtre, on peut obtenir la réponse fréquentielle du filtre qui donne une estimation du spectre de la parole dont on a éliminé l'influence de l'excitation.

Formants: Les fréquences des paires de pôles correspondent aux fréquences de résonances du conduit vocal. On factorise la fonction de transfert H(z) pour déterminer les pôles:

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p} (1 - z_k z^{-1})$$

Soit $z_k = e^{(\sigma_k + j\Omega_k)T}$, on peut obtenir la fréquence et la largeur de bande de chaque formant.

Erreur: L'erreur de prédiction est $e(n)$. Un pic apparaît au moment où le signal change brutalement. $e(n)$ peut être considérée comme le bruit $N(n)$.

Autres: Les coefficients du filtre peut être convertis en: coefficients du cepstre, réponse impulsionnelle du filtre $h(n)$, fonction de la surface du conduit vocal, etc.

6.6 D'autres paramètres

Nous avons également implanté le calcul d'autres paramètres du signal notamment:

- Fonction d'autocorrélation [Oppenheim 75]

  $$\phi(n) = \sum_{i=0}^{N} x(i)x(n+i)$$

  qui a les propriétés importantes que $\phi(n)$ a sa valeur maximum à $n = 0$ et que $\phi(n)$ est périodique si $x(n)$ est périodique.

- Energie moyenne:

$$E_e = \phi(0)/N$$

ou

$$E_a = \frac{1}{N}\sum_{n=0}^{N-1}|x(n)|$$

- Passages par zéro [Niederjohn 75]:

$$Z_n = \sum_{n=0}^{N-1}|sign[x(n)] - sign[x(n-1)]|$$

7 Interface avec l'utilisateur

7.1 Menus

introduction

L'interface avec l'utilisateur du système doit résoudre les problèmes suivants : indiquer en clair à l'utilisateur ce que peut faire le système à un instant donné, donner à l'utilisateur la possibilité de choisir une action proposée par le système, récupérer l'information fournie par l'utilisateur et maintenir le contact direct avec le système d'exploitation. La communication se fait par trois moyens: la souris avec le menu, la souris avec l'image du signal et le clavier alpha-numérique.

Un menu est une liste des couples description-action dont les descriptions sont affichées dans des cases sur l'écran représentant les options parmi lesquelles l'utilisateur peut faire son choix. La description est un texte qui décrit brièvement une action associée. L'action sera déclenchée lorsque le couple est sélectionné par l'utilisateur. Les actions peuvent faire appel à un autre menu. Les menus peuvent être imbriqués pour construire un grand reseau de couples, profond et compliqué [Brown 82]. Un système est contrôlé par menu si chaque demande de l'utilisateur est prédite par un ensemble de menu prédifinis par le système [Arthur 86]. C'est le cas de notre système. En sélectionnant une séquence de couples à travers des menus, l'utilisateur peut parcourir toutes les possibilités de traitement fournies par le système. En conséquence, seuls les paramètres numériques et parfois le nom d'une variable nécessitent l'entrée au clavier.

Le système de menus utilise l'environnement ou le contexte défini par la séquence de commandes données par utilisateur pour guider l'utilisation du système. A chaque instant où le système a terminé l'exécution d'une commande il propose un menu pour inviter l'utilisateur à effectuer un choix parmi un ensemble de commandes qui sont légales à cet instant. Le menu permet ainsi d'éviter en partie des séquences de commandes sémantiquement incorrectes, car seuls les choix valides peuvent apparaître sur le menu. L'ensemble menu plus souris constitue un moyen rapide et puissant pour le contrôle interactif du système. En plus, en sélectionnant dans le menu, l'utilisateur ne peut pas commetre des erreurs syntaxiques. La conception du système, en particulier la partie d'interface avec utilisateur, est largement simplifiée.

structuration du système de menus

Les menus sont organisés hiérarchiquement. L'utilisateur doit simplement effectuer deux ou trois étapes de sélection à l'aide de la souris dans l'arborescence du système de menus multi-niveaux pour accéder à plus d'une centaine de termes. En général chaque menu de commandes est présenté sur l'écran en même temps qu'un menu de paramètres. Ces paramètres sont globaux pour les opérations accessibles dans le menu de commandes et ont des valeurs implicites. Ils sont modifiables avec la souris si ces valeurs ne conviennent pas. L'appendice A montrent la hiérarchie d'une partie du système de menus.

7.2 Macro commandes

Une macro-commande est une suite de commandes, simples ou macro mémorisées pendant la phase de définition, exécutable comme une simple commande. Les macro-commandes sont utiles pour remplacer une suite de commandes et ainsi simplifier le contrôle du système. Nous avons inclus la possibilité de définir et d'exécuter automatiquement des macro-opérateurs pour pouvoir enchaîner les opérations proposées par le système de menus.

Le processus de définition d'une macro-commande est très simple: Après la sollicitation de la commande MACRO-ON dans un menu le système enregistre automatiquement toute information venant de la souris et du clavier. Le processus de définition de macro-commandes se termine par la commande MACRO-OFF. La définition d'une macro-commande peut être imbriquée. C'est à dire qu'à l'intérieur d'une définition de macro-commande une nouvelle définition de macro-commande peut avoir lieu.

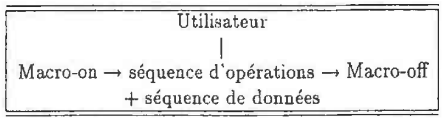
Le principe de fonctionnement des macro-commandes est de simuler le clavier et la souris. Le fonctionnement peut inclure toutes les possibilités du système, y compris macro-commandes elles mêmes, car l'information du contrôle du système est entièrement passée par les deux entrés. Nous avons implanté une couche logicielle à l'entré du système qui effectue l'acquisition des commandes et la relecture des commandes.

Création

Un macro-opérateur est défini par deux fichiers: le fichier de commandes et le fichier de données. Le fichier de commandes contient les positionnements de la souris qui dirige le système. Stockée sous forme de couples, l'information sur la sélection des cas dans les différents menus et sur la définition des fenêtres est mémorisée lorsque le système est en fonctionnement. Le fichier de paramètres contient tout ce que l'utilisateur tape au clavier, essentiellement des nombres et des chaînes de caractères qui ne peuvent être sélectionnés par un menu.

La suite de commandes peut être composée de façon arbitraire et l'utilisateur est responsable de la cohérence entre la macro-commande et l'environnement dans lequel elle sera exécutée.

Le processus de la définition d'une macro-commande peut se schémasiser par la figure suivante:



Réutilisation

La commande RUN-MACRO permet de lancer un macro-opérateur. Une vérification de l'environnement assure que l'état du système est le même que celui au moment où le macro-opérateur a été défini.

Le mécanisme de définition et d'exécution des macro-commandes est implanté de manière à autoriser la récursivité. Chaque fois qu'une macro-commande est en cours de définition, l'environnement de l'ancienne macro-commande est empilé et la séquence de contrôles d'entrée est enregistrée dans toutes les définitions de macro-commandes actives. Cela permet d'imbriquer différentes combinaisons entre définition et exécution de macro-opérateurs. Notamment on peut définir une autre macro-commande lorsqu'on est en train de définir une macro-commande. Afin de pouvoir exécuter une autre macro-commande lorsque une ou plusieurs macro-commandes sont en train de s'exécuter, les entrées de chaque macro-commande active sont sauvegardées.

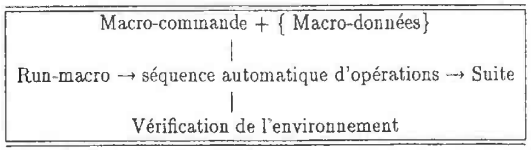
Si une macro-commande est lancée en lui donnant le fichier de commandes et le fichier de données, elle répète tout simplement la séquence d'actions définie au moment de la création. Cependant, les macro-commandes sont paramétrables au sens où l'on peut changer des objets et des paramètres de traitement. La paramètrisation consiste à modifier le fichier de données d'une macro-commande par l'une des deux méthodes suivantes:

1. On ne donne pas le fichier de données obtenu au moment de la définition, le système va demander au fur et à mesure les données au clavier nécessaires à l'exécution.

2. On utilise un éditeur de texte pour modifier directement le fichier de données contenant des paramètres tels que par exemple le nom du signal traité ou le point de vue de la fenêtre d'un buffer.

Pour assurer le déroulement correct d'une macro-commande, il faut qu'au moment de lancement d'une macro-commande, l'environnement du système ou l'ensemble de variables représentant l'état du système soit compatible avec l'environnement au moment de la définition de la macro-commande. Nous avons donc implanté un mécanisme de vérification d'environnement qui refuse l'exécution d'une macro si il y a un conflit avec l'environnement.

Le schéma suivant résume le processus de réutilisation d'une macro-commande:



8 Traitement des fichiers de signal

Nous avons regroupé tous les opérateurs qui prennent comme entrée un fichier de signal et qui donne comme résultat soit un fichier de signal soit une valeur dans le menu "traitement de fichiers". Ces opérateurs utilisent un itérateur commun qui, pour chaque point dans le fichier de signal d'entrée, calcule la valeur d'une fonction. Les opérateurs qui donnent un fichier de signal sont par exemple le filtrage et le calcul de la dérivée d'un fichier de signal. Le calcul de l'énergie moyenne et la valeur maximale sont des exemples d'opérateurs qui donnent un point en résultat.

8.1 Extraction de paramètres

Ce groupe d'opérations permet d'extraire un certains paramètres caractéristiques du signal, par exemple le calcul du contour de l'énergie d'un fichier de signal, le contour de la fréquence fondamentale, le contour du passage à zéro du signal, etc.

8.2 Modification du signal

Ce groupe d'opérations permet d'effectuer des modifications physiques sur le fichier de signal. Ces modifications comprennent l'addition, la soustraction de deux fichiers de signal, la multiplication d'un fichier de signal par une ligne avec une certaine pente, la normalisation d'un fichier à une valeur donnée, le calcul de la dérivée d'un fichier, l'adjonction du bruit Gaussien sur un fichier et le filtrage d'un fichier de signal par un filtre FIR ou IIR dont la caractéristique fréquentielle est donné par l'utilisateur.

8.3 Création du signal

Nous avons implanté les générateurs de constante, de bruit blanc, de sinus, des impulsions et de la parole à partir des coefficients de LPC. Ces signaux permettent de tester et d'évaluer des systèmes de traitement du signal.

9 Simulation des algorithmes

9.1 Introduction

Au cours de ces dernières années, les chercheurs en traitement du signal ont développé de nombreux algorithmes. Cependant, l'effort sur l'organisation et la programmation de ces algorithmes est largement insuffisant [Wu 83]. Cela oblige à reprogrammer certaines composantes de base pour commencer un travail propre, et a par conséquent ralenti les progrès dans ce domaine.

Dans un processus de développement d'algorithme de traitement du signal, le chercheur doit concrètiser l'idée par ses programmes, les tester dans des cas divers, évaluer les points forts et faibles de son algorithme, puis modifier la conception et enfin recommencer. Ce cycle

conception → implantation → test → évaluation limite la vitesse de conception si aucune aide automatique n'est disponible.

Notre objectif est de réduire le plus possible le coût de conception et d'implantation associé au développement de nouveaux algorithmes. A travers l'interface du système, l'utilisateur dispose d'un environnement agréable pour la manipulation et l'analyse de signaux. ASSIA fournit non seulement un ensemble complet d'opérateurs, qui peut de plus être augmenté à loisir, mais aussi la possibilité de construire des logiciels à partir des opérateurs existant dans le système. ASSIA libère l'utilisateur des détails de programmation et de la représentation de données; de plus il fournit un ensemble d'opérateurs qui sont facilement réutilisables pour construire de nouveaux opérateurs. Les opérateurs de signal et de graphisme peuvent être enchaînés.

La plupart des algorithmes de traitement du signal peuvent être formulés à l'aide de la composition de calculs matriciels ou de traitements successifs, ce qui permet leur décomposition en étapes de réalisation représentées par des opérateurs. Les opérateurs réalisant les différents traitements ont des propriétes communes [Bentz 85]:

- Ils peuvent être considérés comme un bloc de traitement ayant une entrée et une sortie;
- Les formats d'entrée et de sortie ont des types et des tailles de données similaires et peuvent être normalisés de façon à les rendre compatibles entre eux.

Cela fournit la possibilité de tester facilement un algorithme avant de chercher à réduire son temps de calcul. C'est une propriété très utile lors du développement d'algorithmes, car c'est souvent la qualité du traitement qui est recherchée dans un premier temps, la préoccupation sur la vitesse d'exécution n'ayant de sens que si cette qualité est satisfaisante.

Un opérateur réalisant une certaine opération peut se décomposer en une séquence de sous-opérateurs. Chaque opérateur compilé séparément possède des caractéristiques relativement simples et bien définies et un format commun d'échange d'informations avec l'extérieur. Cette notion a déjà été partiellement mise à profit pour diverses applications [Johnson 84]. On sait par exemple qu'une analyse cepstrale classique pour obtenir un spectre lissé peut se décomposer en: -acquisition des données brutes, -fenêtrage temporel, -transformée de Fourier, -calcul de log-norm pour chaque composante, -transformée de Fourier inverse, -fenêtrage cepstral et -transformée de Fourier. L'enchaînement d'un nouvel opérateur "Cepstre" peut être donc défini:

$$> Acquis > Window > Dft > Log\text{-}Norm > Idft > Lowtime > Dft$$

Disposant d'une batterie d'opérateurs de base, il devient ainsi possible de définir et de réaliser de nouveaux algorithmes de traitement du signal avec un gain de temps important. Cette organisation permet aussi aux utilisateurs non-informaticiens de construire aisément des opérateurs car on manipule des objets du type signal et des opérateurs et non des données non structurées telles que des suites de réels ou d'entiers.

La construction des opérateurs est simplifiée sous le système Unix par le Shell, un interprèteur de commandes, qui est un langage complet et structuré. Dans ASSIA, le Shell sert principalement à l'analyse syntaxique des commandes, la préparation des différents arguments de description de l'opération à effectuer, le lancement d'un ou plusieurs programmes

correspondant aux opérations demandées par l'utilisateur, la lecture et l'écriture sur l'entrée ou la sortie standard et la réalisation de nouveaux opérateurs à partir de ceux qui existent déjà dans le système.

Unix fournit un moyen très souple de communication entre les processus. Il réalise le schéma de transmission de l'information du producteur vers le consommateur par le mécanisme de "tube" (pipe). La redirection de l'entrée-sortie d'un programme permet facilement de diriger le flux de données entre les programmes. Cela permet à chaque opérateur dans une séquence de traitements de s'appliquer sur la sortie de l'opérateur précédent sans passer par des fichiers intermédiaires. Afin d'avoir un compromis entre la rapidité d'échange d'informations entre les différents processus, la compatibilité de format et la visualisation des données, diverses couches sont implantées dans le système: le tableau, la suite d'octets (structure commune pour la communication entre programmes, par Unix pipe notamment), la suite de nombres entiers (représentant l'univers) et la représentation en Ascii liant le système et l'utilisateur. Les relations entre ces couches sont illustrées par la figure 2.1.

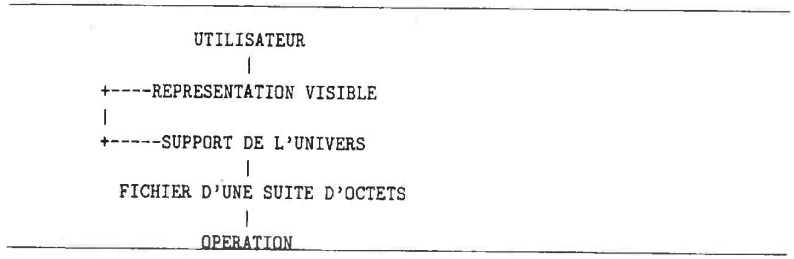


Figure 2.1: Les différentes couches de représentation des données du système

Un opérateur est une description de la manipulation sur le signal. Lorsque l'utilisateur applique l'opérateur "Spect" par exemple, il sait que le résultat est le spectre de l'objet signal donné mais il n'est pas nécessaire pour lui de savoir comment est définie la structure interne d'un signal ni comment est réalisée cette opération.

9.2 Programme

Variables

ASSIA dispose d'un ensemble de variables définies globalement et représentant l'environnement du système. Ces variables sont une source d'informations accessibles par le Shell et par les programmes. Elle sont modifiables par l'utilisateur à l'aide des opérateurs spéciaux. Une variable peut être un opérateur, un signal, ou une valeur scalaire. La longeur de la fenêtre d'analyse et le type de fenêtre utilisée sont par exemple deux variables dont on peut effectuer les affections: "window-length = 256" (variable scalaire) et "window = hamming" (variable d'opérateur).

Primitives

Les opérateurs sont implantés physiquement par des programmes. Ils sont traités par Unix comme des Unix processus. Il n'y a pas de limitation sur le langage de programmation, le système actuel étant écrit en C et en Pascal. La structuration stricte et les possibilités de diagnostic de Pascal facilitent le développement du système tandis que l'efficacité de *C* permet d'obtenir des exécutions rapides. L'utilisation de C permet aussi de disposer des primitives de base pour la gestion de l'écran "bit-map". Un programme d'opération contient essentiellement quatre parties separées:

- la partie physiquement liée au système, effectuant l'interprétation des commandes descriptives fournies comme des paramètres de programme,
- la partie assurant les entrées-sorties, généralement commune pour toutes les opérateurs,
- la partie fonctionnelle qui réalise l'opération demandée et
- la partie décrivant la syntaxe de commande et le mode d'emploi correspondant, faisant partie du système d'aide à l'utilisation.

Les "programmes d'opération" (P.O) sont définis récursivement par une syntaxe présentée sous forme de Backus-Naur sur la figure 2.2.

```
P.O ::= NULL | COMMANDE SHELL | OPERATEUR | PROCEDURE SHELL.
COMMANDE SHELL ::= grep | cat |... <expression>.
PROCEDURE SHELL ::= FONCTION(PROCEDURE SHELL |
          COMMANDE SHELL | OPERATEUR ).
OPERATEUR ::= P.O.P.1 | P.O.P.2 | .... | P.O.P.n <déscriptions>.
P.O.P.i ::= i-ième programme d'opération primitif.
```

Figure 2.2: Définition d'un programme d'opération

Nous distinguons cinq niveaux de programmation dans ASSIA. Ce sont: le noyau du système d'exploitation, les bibliothèques, les programmes d'opérations primitifs, les opérateurs de construction en procédure de Shell qui réalisent des opérations composées fréquemment utilisées, et l'utilisateur. En ce qui concerne les bibliothèques, nous disposons de la bibliothèque Unix et celle des primitifs de la gestion du "bit-map". Une bibliothèque propre au système ASSIA a été construite; elle contient trois sous-bibliothèques: le traitement et l'analyse de signal, le calcul vectoriel et matriciel et le graphique "bit-map". Différents types de signaux sont implantés sous une structure commune. Ils sont compatibles au sens de leur représentation physique. Cette réalisation est avantageuse car elle permet la coexistence dans le système de plusieurs représentations des données: virgule fixe, virgule flottante et signal-réel, signal-complexe. La dimension des données peut aussi être une variable. Une considération importante pour la représentation de données est de trouver un compromis entre la rapidité d'échange d'informations entre les différents processus et la compatibilité de format et la visualisation des données.

Structure de contrôle

Toutes les structures de contrôle existantes dans la Shell sont utilisables dans ASSIA. Nous avons défini en plus certaines structures de contrôle spécifiques au traitement du signal, en particulier des itérateurs sur des fichiers de signal dont nous donnons en exemple dans le paragraphe suivant.

Exemples de programmation

Quelques dizaines d'opérateurs sont implantés actuellement dans le système [Gong 85b]. A titre d'exemple, la figure 2.3 montre un opérateur construit à partir de plusieurs opérateurs et qui consiste à appliquer l'opérateur "formants" à tous les phonèmes "/a/" étiquetés dans les univers (fichiers de parole éventuellement) sous le répertoire courant (représentant un sous-ensemble d'une base de données de parole).

```
with [/a/] do [formants][expression] ::=
      for  univers  in  *
      do
      forall  $1 in  $univers do $3 $4 $5 $6 $7
      done
```

Figure 2.3: Exemple d'application d'un opérateur sur toutes les occurrences d'un phonème dans un ensemble de donneés de parole

L'opérateur "forall [arg1] in [arg2] do [operator] [expression]", qui est un itérateur sur un fichier de signal, applique le traitement "operator" sur toutes les occurences de l'événement "arg1" dans l'univers "arg2", ce qui est une opération très fréquente en reconnaissance acoustico-phonétique de la parole; citons par exemple la création de références, le test des résultats de classification, la réglage de seuils ou l'évaluation d'algorithmes.

Donnons un autre exemple sur la construction des nouveaux algorithmes. Il s'agit de réaliser un algorithme d'estimation spectrale à l'aide d'un modèle ARMA [Cadzow 80] (hparma) synthétisé à partir des opérateurs primitifs. Cet algorithme est décrit sur la figure 2.4.

Sur cette figure, "window" est une variable d'opérateur prédéfinie. "mat -c" indique que l'opérateur "mat" a travaillé sur l'objet transmis par "Unix pipe" (le même que "automat.m") tandis que l'objet créé "p.v" a pour sémantique les coefficients des pôles du modèle. En fait, l'opérateur "mat" lui-même se compose d'un certain nombre d'opérateurs. L'option "-c" signale l'intervention de l'opérateur "mat.cholesky", donnant la solution d'un système d'équation par la méthode de Cholesky. On peut visualiser le résultat en appliquant l'opérateur "u_s" dans la figure 2.5.

Cet opérateur convertit le signal de l'univers et qui a comme source de données l'entreé standard. y1 et y2 précisent la position sur l'écran, "as" est un mot-clé de la syntaxe et "mode" un mode graphique.

```
hparma [p][q] ::=
  $window_length|$window|mautoc $1 $2|tee automat.m|mat -c > p.v
  cat  p.v automat.m| zero > z.v
  cat  p.v z.v  |  dfthparma

mautoc [p][q] ::= matautoc $1  $2 >  autoc.m
                  autoc   $1      >  autoc.v
                  cat  autoc.m  autoc.v
```

Figure 2.4: Définition d'un estimateur spectral ARMA

```
u_s [block][sample][length]<univ|hparma [p][q]|image [y1][y2] as [mode]
```

Figure 2.5: Utilisation de l'opérateur graphique "u_s"

10 Conclusion

En introduisant la notion de type d'objet, défini en terme des différentes opérations applicables, nous avons réalisé avec un modèle conceptuel unique un système à la fois souple et général pour l'édition, l'analyse et le traitement de signaux ainsi que pour la simulation d'algorithmes de traitement. Le système fournit à l'utilisateur non seulement un moyen puissant pour le calcul et la visualisation, mais aussi la possibilité de construire de nouveax algorithmes à partir d'opérateurs primitifs existants, ce qui permet de suivre l'évolution des techniques dans le domaine de traitement du signal et/ou de la communication parlée.

Le système ASSIA, possède actuellement un grand nombre d'opérateurs pour l'analyse de la parole, mais il ne doit cependant pas être considéré simplement comme une collection de programmes, mais comme un noyau extensible et interactif qui fournit une base de travail pour la construction et le développement de nouveaux algorithmes, éventuellement dans d'autres domaines que la parole.

L'environnement du système Unix a non seulement facilité la réalisation de l'ASSIA mais aussi simplifié le processus d'implantation, de modification et d'évaluation des algorithmes de traitement. On dispose ainsi d'un langage de très haut niveau facilement accessible.

Chapitre 3

Un Modèle d'Estimation de Pitch

Un modèle est un système spécifié pour représenter certains aspects d'un phénomène complexe et pour ensuite les étudier. Le rôle essentiel d'un modèle est de reproduire le phénomène modélisé sous une forme simplifiée mais contenant toutes les caractéristiques intéressées. Tenant compte des propriétés du signal de parole, nous proposons un modèle paramétrique dans l'objectif d'estimer la periode de la parole. A partir du modèle, nous proposons une nouvelle formulation du problème de détection de pitch, où le signal de parole est modélisé par une séquence de fonctions spécifiées de façon dépendante du temps. Cette représentation autorise la variabilité en fonction du temps de la période et de l'amplitude de l'excitation du signal de la parole. L'assymétrie de la distribution de signal par rapport à l'axe temporel est aussi mise en utilisation dans la détection. En optimisant un critère énergétique, nous obtenons une fonction de ressemblance. L'estimation de la période est réalisée par maximisation de cette fonction. Pour un segment voisé, notre algorithme fournit également une estimation de la position et de l'amplitude du maximum pour chaque période. Une interprétation dans le domaine fréquentiel montre que notre approche est equivalente au processus d'appariement de structures harmoniques découvert récemment dans la perception de pitch. Nous présentons des expériences pour l'évaluation de l'algorithme pour la parole propre, bruitée et filtrée par une ligne téléphonique simulée. Les résultats indiquent que: - l'estimation est presque sans erreur dans le cas parole propre, - la présence de la première harmonique n'est pas indispensable, - l'estimation a une haute immunité au bruit et - le modèle de signal est suffisant pour suivre des variations rapides de pitch. Une étude de complexité montre que l'efficacité de notre méthode est au moins comparable aux plus efficaces méthodes de détermination de pitch basées sur la recherche de structures harmoniques.

1 Introduction au modèle

Un modèle est un système spécifié pour représenter certains aspects d'un phénomène complexe et pour ensuite les étudier. Les aspects sont caractérisés par des paramètres du modèle.

Un modèle doit être

- suffisamment général pour inclure toutes les variations du phénomème modélisé et

- suffisamment restreinte pour éliminer toutes les influences sur les aspects intéressés.

Un bon modèle tient compte convenablement des particularités du phémonène modélisé.

Lors du traitement d'un signal, on instancie un modèle – calculer un ensemble de paramètres en fonction du signal observé. Différents ensembles de paramètres modélisent des signaux différents. La quantité d'information sur le phénomène dans l'ensemble de paramètres résultants est inféreure à celle du signal original. Cette réduction d'information permet d'extraire le caractère essentiel du phénomène et de réduire le calcul ultérieur. Cependant, la perte d'infomation est irrécupérable dans les traitements qui suivent.

Selon la profondeur de la modélisation, nous pouvons distinguer deux types de modèles: les modèles fonctionnels et les modèles comportementaux.

1.1 Modèle comportemental

Il existe des phénomènes dont la description du mécanisme de production est difficile ou dont la réalisation artifitielle de ce mécanisme est compliquée. Les modèles comportementaux se contentent de décrire et de reproduire des aspects du phénomène observé auxquels on est intéressé et on peut les qualifier de modèles externes. On cherche a réinventer le processus de la production du phénomène. Le critère sur ces modèles est la fidèlité des fonctions de transfert excitation-réponse par rapport au mécanisme modélisé.

1.2 Modèle fonctionnel

Un modèle fonctionnel cherche à modéliser le processus réel de généralisation du phénomène étudié et on peut les qualifier de modèles internes. Autrement dit on cherche à copier la nature. La recherche d'un tel modèle, connue sous le nom de *bionique*, consiste à examiner le mécanisme de base de la production du phénomène. La modélisation consiste ensuite à simuler ce mécanisme. Si la description du mécanisme est correcte, le modèle peut exactement reproduire le phénomène. Tous les modèles fonctionels modélisent le mécanisme de production des phénomènes à un niveau d'abstraction limité.

2 Estimation de la fréquence fondamentale de la parole

La fréquence fondamentale de la parole est une information importante transportée par le signal de la parole. Une estimation correcte du contour de la fréquence fondamentale est indispensable pour tous les systèmes d'analyse-synthèse de la parole [Flanagan 72], de la rééducation des personnes sourdes assistée par l'ordinateur [Haton 79b], de l'identification de locuteurs [Atal 72] et des diagnostics cliniques de la voix [Laver 82]. Etant une composante majeure de l'information prosodique, la fréquence fondamentale est aussi une source de connaissance pour les systèmes de reconnaissance et de compréhension de la parole. [Lea 75] [Haton 84]. On a aussi démontré qu'il est possible de reconnaître les chiffres en anglais parlé en n'utilisant que l'information prosodique [Willems 72].

Cependant, bien que des algorithmes de différentes variétés aient été proposés [Rabiner 76] [Hess 83], l'estimation de ce paramètre est encore un problème loin d'être complètement et

systématiquement résolu. Différentes types de traitements tels que le lissage non-linéaire [Rabiner 75b], la programmation dynamique [Ney 83] et même la manipulation symbolique à base de connaissances – l'approche intelligence artificielle [Dove 83] – sont utilisés dans le système pour que le contour de fréquence fondamantale obtenu soit significatif.

La plupart de détecteurs de fréquence fondamentale sont incapables de donner des résultats dans des situations difficiles telle que environnement bruité, limitation de bande passante de la voie de transmission, transition brutale de la fréquence ou de l'amplitude, etc [Rabiner 76] [Hess 83] [Wise 76]. Ayant identifié ces difficultés, nous pensons que la recherche sur la détection de la fréquence fondamentale de parole doit se concentrer sur les aspects suivants:

- l'exploration des nouveaux modèles de signal qui utilisent plus explicitement la structure de périodicité du signal de la parole,
- l'inclusion du bruit dans le modèle utilisé pour l'estimation et l'introduction de certains effets d'évaluation de moyenne afin de diminuer le niveau de bruit effectif et d'utiliser entièrement l'information sur la fréquence fondamentale dans le signal,
- le développement des algorithmes qui sont formulés sur la structure harmonique du signal et qui explorent tout le spectre du signal disponible à un instant donné, et non seulement la première harmonique ou simplement les paramètres de l'onde du signal dans le domaine du temps qui sont trop sensibles à l'amplitude, à la distortion des phases spectrales et au bruit,
- la modélisation du signal de parole pour décrire les variations rapides de la fréquence et de l'enveloppe de la parole,
- la formulation du problème de l'estimation basée sur la connaissance d'aujourd'hui sur les modèles de perception de la fréquence fondamentale et l'intégration dans l'estimateur d'une simulation du système de codage du signal nerveux pour assurer que la performance et la distribution d'erreurs approche le comportement de l'oreille humaine,
- en même temps, maintenir une complexité de calcul acceptable vis-àvis des matériels actuels pour que les algorithmes puissent être implantés en temps réel dans la plupart des applications.

2.1 Travaux précédents

Pour un signal stationnaire de parole, la fréquence fondamentale peut être définie comme la perception de la fréquence fondamentale d'un gabari d'harmonies pures qui apparient optimalement la forme des composantes harmoniques successives du signal de la parole [Goldstein 73]. Nous appellons l'estimation de la fréquence fondamentale formulée en relation avec cette définition et qui examine la structure harmonique du signal *l'estimation de la fréquence fondamentale par l'appariement des structures harmoniques*, ou *EFFASH* en abréviation. Bien que le terme *EFFASH* n'ait pas été utilisé jusqu'au présent, La détermination de la fréquence fondamentale par *EFFASH* a attiré l'attention des chercheurs de différents points de vue. Plusieurs auteurs ont réalisé qu'un estimateur ayant une bonne performance doit se fonder sur la stratégie de recherche de la structure harmonique:

- La détermination de la fréquence fondamentale par la méthode cepstrale proposée par Noll [Noll 64] examine la périodicité dans le spectre logarithmique du signal de parole par l'application d'une autre transformée de Fourier puis par la recherche dans le cepstre obtenu d'un pic indiquant la période.

- Schroeder considère la fréquence fondamentale comme le plus grand commun diviseur des composants harmoniques obtenus par une transformée de Fourier sur le signal de parole [Schroeder 68] [Miller 70].

- Wise et ses collègues formulent, dans le domaine temporel, le problème comme l'estimation d'un signal périodique inconnu dans un bruit gaussien de densité connue et proposent une solution explicite de vraisemblance maximum [Wise 76]. Leur résultat peut être vu comme la maximisation de l'énergie totale de la fonction d'auto-corrélation multipliée par une série de Dirac ayant une période variable. En termes de *EFFASH*, ceci est équivalent à apparier le spectre du signal avec une série de Dirac spectrale.

- Afin d'assimiler les variations lentes en période à l'intérieur de chaque fenêtre d'analyse comme une source de bruit, Friedman [Friedman 77] introduit une fonction de pondération dans son modèle du signal. L'estimateur obtenu fonctionne directement sur la version pondérée du signal et non sur la fonction d'auto-corrélation.

- Directement fondé sur le fait que dans un spectre harmonique les pics successifs sont espacés à une distance égale proportionnelle à la fréquence fondamentale, l'algorithme de Seneff [Seneff 78] détermine la fréquence fondamentale à partir de l'écart entre les harmoniques dans la partie base du spectre.

- Dans la formulation de Martin [Martin 82], on examine la périodicité de la parole par l'évaluation de la corrélation entre son spectre et une série de fonctions de Dirac avec une amplitude décroissante. La meilleure corrélation qui donne l'estimation de la période est obtenue en faisant varier la période de la série de Dirac.

- La formulation de Paliwal et ses collègues [Paliwal 83] est basée sur le même principe. La différence majeure qui réduit l'influence des formants et qui augmente la fiabilité de la méthode est que la série de fonctions de Dirac est pondérée par le spectre du signal obtenu par une prédiction linéaire à tout-pôle.

- Motivés par le modèle de perception de pitch proposé par Goldstein [Goldstein 73] [Goldstein 78], Duifhuis et ses collègues [Duifhuis 82] [Sluyter 80] ont implanté un détecteur de la fréquence fondamentale constitué d'une analyse de l'amplitude spectrale du signal acoustique, d'une extraction de composants fréquentiels du spectre et d'un appariement de structures harmoniques qui est basé sur la technique de crible harmonique. Ce détecteur peut être considéré comme la réalisation la plus directe et la plus explicite de *EFFASH*.

Cependant, ces estimateurs sont fondés sur l'hypothèse que le signal de parole est stationnaire à l'intérieur de l'intervalle d'observation. A cause de cette hypothèse, leur performance est limitée car en réalité l'hypothèse n'est pas toujours vérifiée. Il est possible de réduire l'effet de la non-stationnarité en traitant le signal dans un intervalle d'observation encore

plus petit mais comme compromis, la précision sur la fréquence estimée, l'immunité au bruit et la fiabilité vont être inévitablement dégradées car, dans ce cas, moins d'information sur la périodicité est disponible. Par ailleur, ces estimateurs sont incapables de donner une estimation sur les positions des pics maximaux dans chaque période à cause de leur schéma de traitement en bloc.

Comparé aux autres types d'estimateur de la fréquence fondamendale, l'*EFFASH* fonctionne d'une manière qui approche, dans une certaine mesure, le modèle de perception de pitch de l'homme connu actuellement. La faculté de concentrer suffisament l'information sur la périodicité portée par le signal dans une fenêtre d'analyse (normalement 20-60 ms) et donc de donner une indication nette de période et la faculté d'éliminer l'influence de bruit blanc montrent l'avantage de l'*EFFASH*. Les algorithmes de *EFFASH* sont en général plus fiables dans des situations difficiles mentionnées antérieurement. Ils donnent de bons résultats même pour le signal de parole où la première harmonique est complètement absente. En plus, *EFFASH* est une approche relativement systématique et non heuristique.

2.2 Nouveautés

Dans ce chapitre, nous proposons d'abord un nouveau modèle de signal destiné à estimer la fréquence fondamentale de la parole. Nous décrivons ensuite notre approche de l'appariement de structures harmoniques basée sur ce modèle pour l'estimation de la fréquence fondamentale. L'estimation est formulée dans le domaine temporel et d'une manière non-stationnaire. Notre approche est différente des travaux précédents par les aspects suivants:

- Un modèle paramétrique d'onde temporelle du signal avec une période variable dans chaque période du signal est présumé pour chaque fenêtre d'analyse. Cette représentation du signal permet une meilleure utilisation de la structure périodique de la parole et surmonte certaines ambiguïtés provoquées par les formants à basse fréquence et haute énergie et à largeur de bande étroite,

- Le modèle que nous proposons est dépendant du temps dans le sens où les périodes variables et les variations dans l'énergie à court terme sont autorisées. Les erreurs causées par les deux facteurs sont ainsi diminuées,

- Une estimation de la longueur de période et de la position et de l'amplitude du pic maximum de chaque période dans une fenêtre d'analyse est fournie en même temps,

- L'assymétrie du signal de parole par rapport à l'axe du temps est prise en compte dans l'estimation. Ceci améliore la fiabilité de la méthode pour le signal produit par certains phonèmes nasalisés,

- La recherche de structures harmoniques est réalisée par un processus d'appariement de l'onde du signal dans le domaine temporel,

- Cette approche est efficace. Elle demande beaucoup moins de calcul que les algorithmes basés sur la transformée de Fourier rapide.

Dans la section 3, la modélisation paramétrique de la parole pour l'estimation de la fréquence fondamentale et la formulation mathématique de la méthode proposée sont développées. La section 4 décrit les points importants concernant la réalisation de notre algorithme. La méthode de réduction de calcul discutée dans cette section permet que la complexité de l'algorithme soit compatible avec les détecteurs efficaces tels que AMDF [Ross 74]. Une interprétation du comportement dans le domaine fréquentiel est donnée dans la section 5 qui révèle que la méthode fonctionne de la manière *EFFASH*. Notre algorithme a été évalué sur un corpus de parole contenant la parole enregistrée dans une salle calme et dans une salle de machine bruyante, la parole filtrée par une ligne téléphonique simulée et la parole fortement bruitée telle que le rapport de signal sur bruit est inférieur à 0dB. Ces résultats sont présentés et discutés dans la section 7.

3 Principe

3.1 Introduction

Les algorithmes basés sur l'*EFFASH* qui fonctionnent dans le domain temporel demande en général une quantité de calcul importante pour obtenir l'information spectrale. Puisque la structure harmonique dans le domaine fréquentiel et la périodicité dans le domaine temporel sont essentiellement équivalentes, l'appariement des structures harmoniques dans le domaine fréquentiel peut être alternativement réalisé dans le domaine temporel. Ce changement de domains permet de réduire le calcul tout en maintenant la performance. Une autre raison de formuler le problème de l'estimation de la fréquence fondamentale dans le domaine temporel est de pouvoir traiter correctement certains phénomènes revélés par des expériences de perception de pitch pour lesquelles il est difficile de construire un modèle dans le domain fréquentiel. Des exemples de ces phénomèmes sont la variation rapide dans la fréquence instantanée et la rectification de l'onde acoustique pendant une stimulation périodique acoustique et le mécanisme de l'adaptation d'amplitude en fonction du temps du système auditif [Cohen 82]. Ceci suggère qu'un estimateur de bonne qualité doit être non seulement capable de rechercher la structure harmonique pour un signal stationnaire mais aussi d'inclure une certaine sélection de polarité et un mécanisme pouvant traiter la non-stationnarité de la parole.

3.2 Formulation mathématique

Parmi tous les phénomènes concernant la non-stationnarité du signal de parole, deux ensembles de paramètres variables en fonction du temps, de la période et de l'amplitude de l'excitation, sont particulièrement intéressants dans le problème d'estimation de la fréquence fondamentale. Par conséquent, un modèle variable en fonction du temps pour cette estimation doit explicitement inclure ces paramètres. Il est bien connu que le signal de parole dans un invervalle [0,N-1] peut être modélisé comme la réponse d'un système linéaire $h(n)$ à une série d'excitations, plus une source de bruit additif représentant l'environnement bruité. Le système linéaire contient l'influence de la glotte, du conduit vocal et des conditions de transmission telles que la ligne téléphonique. La série d'excitation est une séquence d'impulsions

pour les segments voisés ou une séquence de bruit aléatoire pour les segments non voisés ou alors simultanément les deux pour les segments où une excitation mixte est présente. Du point de vue du traitement du signal, les segments à excitation mixte peuvent être considérés comme de la parole à excitation périodique bruitée. Afin de traiter les phénomènes plus généraux, nous étendons la séquence périodique au cas variable en fonction du temps par les deux aspects suivants:

- Premièrement, de la séquence purement périodique à une séquence périodique dépendante du temps pour décrire la pseudo-périodicité de la parole provoquée par la perturbation du système de la production de la parole et par les variations porteuses d'information linguistique,
- Deuxièment, de l'amplitude constante à l'amplitude variable en fonction du temps pour spécifier les variations du signal qui apparaissent au cours des transitions entre les phonèmes.

Nous écrivons la séquence comme

$$A_0(n)\,\delta_p(n) \tag{3.1}$$

où

$$A_0(n) \geq 0 \quad 0 \leq n < N \tag{3.2}$$

est la fonction de l'enveloppe d'excitation et

$$\delta_p(n) = \sum_{i=0}^{I} \delta(n - m_i) \quad \text{avec} \quad m_0 = 0,\ m_i < m_{i+1} \tag{3.3}$$

est la série Dirac pseudo-périodique où

$$\delta(n) \equiv \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{3.4}$$

La limite supérieure I satisfait $m_I < N$. En posant

$$p_i = m_i - m_{i-1} > 0 \quad 0 < i \leq I \tag{3.5}$$

comme la longueur de période entre deux impulsions successives, nous obtenons

$$\delta_p(n) = \sum_{i=0}^{I} \delta(n - \sum_{j=1}^{i} p_j) \tag{3.6}$$

Nous construisons le vecteur

$$\mathbf{p} = [p_1, p_2, \ldots p_I]^t \tag{3.7}$$

pour désigner les périodes p_i, $0 < i \leq I$. Les deux ensembles de paramètres $A_0(n)$ et $\mathbf{p}$ permettent de reproduire tous les types de signal d'excitation pour la parole voisée. Le signal de parole observé peut donc s'écrire comme

$$[A_0(n)\,\delta_p(n)] * h(n). \tag{3.8}$$

Où $*$ signifie la convolution, et $h(n)$ est la réponse impulsionnelle du conduit vocal.

L'exploitation directe de Eq-3.8 pour l'estimation de la fréquence fondamentale, c'est-à-dire pour fournir une estimation de l'ensemble de paramètres $\mathbf{p}$, peut entraîner des calculs compliqués car ni $h(n)$ ni $[A_0(n)\delta_p(n)]$ ne sont mesurables. Cependant ceci peut être évité en modélisant alternativement l'onde du signal de la parole voisée par une séquence de fonctions spécifiées $f(n)$. L'amplitude de cette séquence est modulée et sa période est variable en fonction du temps. L'ensemble de périodes de la séquence à estimer est $\mathbf{p}$. Soit

$$\Pi(f,n) = \delta_p(n - N_0) * f(n) \tag{3.9}$$

la séquence pseudo-périodique, où N_0 est le délai initial et $f(n)$ dépend de l'onde du signal de parole. Soit l'enveloppe de l'amplitude du signal observé $A(n)$, une fonction positive. Nous représentons le signal par

$$A_\Pi(f,n) = A(n)\,\Pi(f,n) = A(n)\,[\delta_p(n - N_0) * f(n)] \tag{3.10}$$

A(n) et f(n) seront étudiées en détail dans la section suivante. Cette modification ne change pas de façon significative les paramètres d'excitation tels que la période et l'enveloppe. La figure 3.1 donne un exemple de signaux synthétisés par Eq-3.8 et Eq-3.10 respectivement. La synthèse est réalisée en utilisant le LPC tout-pôle modèle fonction d'amplitude linéaire. Les coefficients de LPC sont extraits de corpus de parole réelle. Puisque l'objectif du modèle est d'estimer la période, la faible inconsistence introduite n'a pas d'influence sur le résultat final.

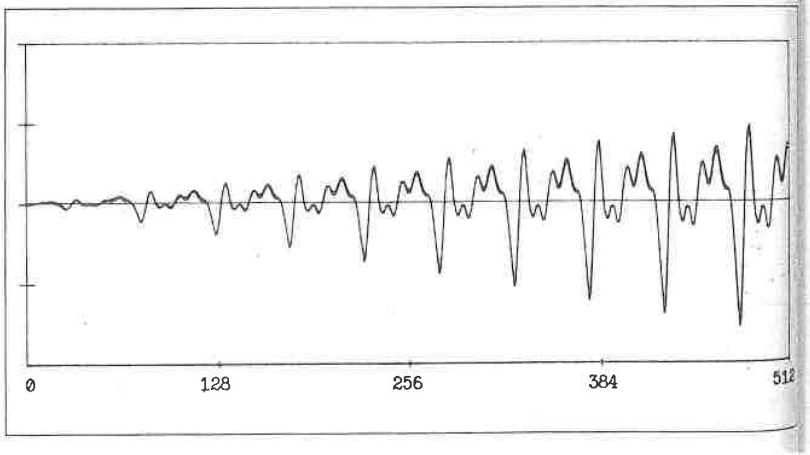


Figure 3.1: signaux de parole synthétisés avec une fonction d'amplitude linéaire.

La stratégie consiste à chercher une fonction R qui mesure le mieux la ressemblance entre le signal présumé $A_\Pi(f,n)$ et le signal de parole observé $S(n)$. Raisonablement, l'énergie

totale de l'erreur de l'appariement est utilisée comme critère de mesure à optimiser. Ce critère assure l'inclusion de la sensibilité à l'énergie de l'oreille humaine dans l'estimation. En plus, l'expression formulée est sous forme quadratique et est donc intéressante mathématiquement. La maximisation de la fonction de ressemblance en faisant varier les périodes inconnues $\mathbf{p}$ donne l'estimation de l'ensemble de périodes. Pour simplifier les notations, nous écrivons le signal présumé sous forme de vecteur dans les textes qui suivent, i.e:

$$\mathbf{A_\Pi} = \mathbf{A} \otimes \mathbf{\Pi}(f) \tag{3.11}$$

où $\mathbf{A}$ et $\mathbf{\Pi}(f)$ sont des vecteurs à N dimensions:

$$\mathbf{A} = [A(0),\ A(1),\ ...,\ A(N-1)]^t, \tag{3.12}$$

et $\mathbf{\Pi}(f) = [\Pi(f,0),\ \Pi(f,1),\ ...,\ \Pi(f,N-1)]^t$. N est la longueur de la fenêtre d'analyse. L'opérateur $\otimes$, appelé *multiplication élément par élément*, peut simplifier les peut simplifier les dérivations. La multiplication élément par élément de deux vecteurs à N dimensions $\mathbf{x}$ et $\mathbf{y}$ donne le vecteur $\mathbf{z}$ dont les composantes z_i satisfont $z_i = x_i\, y_i$ et s'écrit en $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$.

Nous construisons un vecteur $\mathbf{S_\Pi}$ à partir du signal observé S(n):

$$\mathbf{S_\Pi} = \mathbf{S} \otimes \mathbf{\Pi}(g) \tag{3.13}$$

où $\mathbf{S} = [S(0),\ S(1),\ ...,\ S(N-1)]^t$ et

$$g(T,n) \equiv \begin{cases} 1 & 0 \le n < T \\ 0 & \text{sinon} \end{cases} \tag{3.14}$$

est la fonction de fenêtre qui permet de sélectionner une portion de signal.

Nous introduisons une fonction de ressemblance R, influencée par le signal de parole observé,

la fonction de fenêtre, l'ensemble des périodes inconnues $\mathbf{p}$ et la position initiale du signal présumé N_0

$$R(\mathbf{S},\ g,\ \mathbf{p},\ N_0) \rightarrow \Re. \tag{3.15}$$

Cette fonction apparie les deux vecteurs $\mathbf{A_\Pi}$ et $\mathbf{S_\Pi}$ de la façon suivante:

$$\mathbf{S_\Pi} = R(\mathbf{S},\ g,\ \mathbf{p},\ N_0)\,\mathbf{A_\Pi} + \mathbf{e} \tag{3.16}$$

Le vecteur $\mathbf{e} = [e(0),\ e(1),\ ...\ e(N-1)]^t$ représente l'erreur d'appariement et le bruit du canal de transmission. Pour des raisons statistiques et expérimentales, les éléments de la séquence e(n) sont considérés comme ayant une moyenne nulle, indépendants et non-corrélés avec le signal

$$\mathbf{e} = \mathbf{S_\Pi} - R(\mathbf{S},\ g,\ \mathbf{p},\ N_0)\,\mathbf{A_\Pi}. \tag{3.17}$$

La perturbation dans la périodicité du signal de parole réel est inclue dans le modèle par l'affectation d'une largueur T à la fonction de fenêtre. Cette perturbation peut être aussi considérée comme une source de bruit supplémentaire et peut être éliminée dans chaque fenêtre d'analyse par le processus d'estimation. A cause de la non-stationnarité du signal de parole, la variance de {e(n)} est en général variable en fonction du temps. Pour que

chaque élément dans e influencé par la valeur de l'enveloppe de parole différente puisse avoir une contribution égale à la mesure d'énergie, nous pondérons chaque terme dans le produit interne du vecteur d'erreur. Nous introduisons une matrice $\mathbf{W}$ $N \times N$:

$$\mathbf{W} = [w_{i,j}] \tag{3.18}$$

$$\text{avec} \quad w_{i,j} = \begin{cases} \frac{1}{A(i)} & i = j \\ 0 & \text{sinon} \end{cases} \tag{3.19}$$

Pour corriger l'influence de la non-stationnarité de la variance du signal sur la séquence d'erreur, nous minimisons l'espérance

$$\mathcal{E}[\frac{e^2(n)}{A(n)}] \tag{3.20}$$

qui peut par équivalence s'exprimer sous forme matricielle

$$J(R) = \mathbf{e}^t \, \mathbf{W} \, \mathbf{e} \tag{3.21}$$

Les deux processus d'optimisation sont donnés par les formules suivantes:

1. Nous cherchons d'abord la forme de la fonction en minimisant J(R):

$$J_{min}(R) = \min_R [J(R)] \tag{3.22}$$

2. Nous maximisons ensuite le résultat R en ajustant $\mathbf{p}$ et N_0:

$$R_{max}(\mathbf{S}, g, \mathbf{p}, N_0) = \max_{N_0 \in D_{N0}, p_i \in D_p, C(\mathbf{p})=0} [R(\mathbf{S}, g, \mathbf{p}, N_0)] \tag{3.23}$$

où D_{N0} et D_p sont les intervalles à l'intérieur desquels N_0 et $\mathbf{p}$ sont recherchés. C est la contrainte sur les relations entre les composantes de $\mathbf{p}$ et sera détaillée dans la section suivante. Lorsqu'il est optimisé, $\mathbf{p}$ donne les périodes estimées.

La mise à zéro de la dérivée partielle de $J(R)$ par rapport à R, i.e.:

$$\frac{\partial}{\partial R} J(R) = [\frac{\partial}{\partial R}(R \, \mathbf{A_\Pi})]^t [\frac{\partial}{\partial (R \, \mathbf{A_\Pi})} J(R)] = 0 \tag{3.24}$$

donne l'estimateur de la fonction de ressemblance R. On peut démontrer, sous l'hypothèse précédente, que cet estimateur est non biaisé.

$$R(\mathbf{S}, g, \mathbf{p}, N_0) = [\mathbf{A_\Pi}^t \mathbf{W} \mathbf{S_\Pi}][\mathbf{A_\Pi}^t \mathbf{W} \mathbf{A_\Pi}]^{-1}. \tag{3.25}$$

L'estimation est précise du point de vue de la perception car l'optimisation porte directement sur l'énergie de l'erreur résiduelle du vecteur d'erreur. Une solution efficace pour le calcul de $\mathbf{p}$ est présentée dans la section suivante.

4 Considérations sur l'implantation

Pour simplifier la notation dans les discussions qui suivent, des zéros sont ajoutés implicitement a la fin d'une séquence si on l'utilise en dehors de l'intervalle où elle est définie.

4.1 Détermination de la function d'excitation $f(n)$

Pour spécifier la fonction de ressemblance, la fonction d'excitation $f(n)$ dans Eq-3.9 doit être déterminée. Comme nous l'avons indiqué, cette fonction est relative à l'onde du signal de parole et par conséquent elle doit être représentative pour tous les signaux de parole ayant une excitation périodique ou mixte. Idéalement, $f(n)$ devrait être remplacée de façon continue pendant la transition de phonème en phonème. Cependant le fait que la fonction doit être facilement générée et manipulée nous empêche de satisfaire complétement cette exigence. Dans la version actuelle de l'algorithme, une fonction de fenêtre ayant le même paramètre que celle dans Eq-3.13 a été choisie. Dans ce cas $A_\Pi(f, n)$ de Eq-3.10 est une fonction de fenêtre pseudo-périodique dont l'amplitude est modulée. Cette technique diminue une grande quantité de calcul et n'introduit qu'une baisse des performances négligeable. La forme de $f(n)$ semble peu sensible aux paroles différentes car

- à l'intérieur d'une largueur T de la fonction de fenêtre sélectionnée, la différence entre les impulsions du signal présumé et l'onde de parole réelle est peu importante lorsque le meilleur appariement est réalisé et
- pour l'estimation de période, ce sont la position et la portion à haute énergie et non la forme précise dans chaque période de la parole qui sont intéressantes.

En substituant à $f(n)$ dans Eq-3.9 par $g(n)$ et en appliquant les propriétés de la multiplication élément par élément $\mathbf{\Pi}(g) \otimes \mathbf{\Pi}(g) = \mathbf{\Pi}(g)$, nous obtenons:

$$R(\mathbf{S}, g, \mathbf{p}, N_0) = [\mathbf{S}^t \mathbf{\Pi}(g)][\mathbf{A}^t \mathbf{\Pi}(g)]^{-1}. \tag{3.26}$$

4.2 Approximation de l'enveloppe $A(n)$

Par enveloppe de parole, nous entendons la séquence de lignes droites qui connectent deux maxima de périodes succesives. Il est difficile d'extraire $A(n)$ du signal de parole sans connaître la longueur de chaque période. L'extraction de l'enveloppe utilisant un modèle complexe peut entraîner des difficultés lors de la détermination de paramètres et peut être ainsi lourde en calcul. Cependant, pour la longueur de fenêtre d'analyse couramment utilisée, la variation dans l'enveloppe de l'amplitude du signal de parole peut être toujours considérée comme monotone grâce à la constante de temps du système de production de la parole. Pour estimer la période, cette enveloppe peut donc être approchée de façon satisfaisante par deux lignes même dans le cas de transitions rapides de certains sons. L'une de des lignes peut éventuellement être réduite à un point. La séquence $A(n)$ est représentée alors par

$$A(n) = aU(n) + b \sum_{i=0}^{n} g(T_w, i - T_0) \quad 0 \leq n < N \tag{3.27}$$

avec

$$U(x) \equiv \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3.28}$$

Les constantes T_0 et T_w vérifiant $0 \leq T_0 < N$ et $0 \leq T_w < N$ sont respectivement la position initiale et la largueur de la fonction de fenêtre Eq-3.14. Avec a et b, ces paramètres sont déterminés lorsque le signal de parole est disponible, en utilisant une méthode similaire à celle décrite dans [Dubnowski 76]. Dans la figure 3.7 (page 79), nous donnons une enveloppe typique du signal de parole.

4.3 Sélection de polarité du signal

Une possibilité pour tenir compte de l'effet de rectification [Cohen 82] dans l'estimation consiste à considérer l'onde de parole comme un signal polarisé avec assymétrie par rapport à l'axe du temps. L'objectif de la sélection de polarité est de produire un signal dans lequel les impulsions périodiques paraissent plus évidentes pour les traitements ultérieurs. Puisque la séquence de la fonction de fenêtre utilisée pour apparier le signal de parole est non-négative, dans l'objectif de l'estimation des périodes, le pic principal dans chaque période peut être plus distinct d'un côté de l'axe du temps que l'autre, à cause de l'asymétrie du signal de parole. Un renversement de la polarité du signal observé, peut donc éventuellement réduire l'ambiguïté dans la détermination de la période et donner un meilleur appariement pendant le processus d'estimation. Ce renversement est particulièrement intéressant pour les signaux produits par certains phonèmes nasalisés où l'asymmétrie est importante. Dans la suite, le signal de parole sera supposé ne pas contenir la composante continue. Il est évident que cette condition est facile à remplir. L'idée de base de la détermination de polarité est de chercher grossièrement dans le signal de parole le côté de l'impulsion d'excitation. Nous utilisons les mesures d'asymmétrie suivantes:

- Les impulsions d'excitation se situent plus probablement du où l'espérance du maximum est plus grande. Si on définit le rapport

$$R_a = \frac{\mathcal{E}[MAX(-S(n))]}{\mathcal{E}[MAX(S(n))]} \quad 0 \leq n < N \tag{3.29}$$

  alors la partie positive de l'onde de parole doit être appariée avec le signal présumé si $R_a \leq 1$, sinon on apparie la partie négative.

- Les observations sur l'onde de la parole et les analyses simples suggèrent que les impulsions d'excitation se situent plus probablement sur le côté où l'espérance du nombre d'échantillons du signal est plus petite. Nous définissons le rapport

$$R_d = \frac{\mathcal{E}[\sum_{i=0}^{N-1} U(S(i))]}{\mathcal{E}[\sum_{i=0}^{N-1} U(-S(i))]} \tag{3.30}$$

  et nous décidons que si $R_d \leq 1$ alors la partie positive doit être appariée sinon la partie négative.

Ces deux critères sont combinés à une fonction pour la détermination de polarité P(S):

$$P(S) = R_a \times R_d. \tag{3.31}$$

Si R_a ou R_d est égal à 1, le signal est symétrique par rapport à l'axe du temps sous ces critères. L'onde de la parole doit être inversée avant d'effectuer l'appariement si $P(S) > 1$. L'analyse théorique et l'expérience montrent que la polarité ainsi déterminée est insensible au bruit même si le rapport S/N est très petit.

4.4 Estimation du délai initial

Le délai initial N_0 de la séquence $\Pi(f, n)$ dans Eq-3.9 est le nombre d'échantillons qui donne l'accord entre la position de la k-ème fonction de fenêtre et le pic maximal de la k-ème période du signal de parole dans la fenêtre d'analyse considérée. Ce délai assure le processus d'appariement. L'estimation de ce délai doit être efficace. Sur une grande quantité de spécimens de parole voisée, nous constatons que le maximum de chaque période est le plus stable et informatif. Le maximum à l'intérieur d'une fenêtre extrait d'une manière statistique indique la position N_m du maximum du plus grand pic de la k-ème période, bien que le k ne soit pas nécessairement égal à 1. N_0 peut être calculé par

$$N_0 = N_m - \sum_{j=1}^{k} p_j \quad 0 \leq k < I \tag{3.32}$$

avec la relation $C(\mathbf{p}) = 0$. L'estimation de N_m est réalisée par la construction d'un vecteur

$$\mathbf{V} = [V(0), V(1), \ldots V(N-1)]^t$$

à partir du signal $\mathbf{S}$ et puis par la recherche de l'indice de l'élément le plus grand:

$$\mathbf{V} = \mathbf{M} \times \mathbf{S}$$

où

$$\mathbf{M} = [m_{i,j}]$$

avec

$$m_{i,j} = \frac{1}{M_0}[U(i-j) - U(i-j-M_0)]. \tag{3.33}$$

$0 < M_0 < N$ est une constante. En effet, la séquence $V(n)$ est une version de moyenne mobile de $S(n)$. Le choix correct de M_0 permet à $V(n)$ de suivre la courbe de $S(n)$ avec une variance atténuée dans un intervalle de l'ordre de M_0.

Nous constatons que l'estimation est fiable dans les cas d'excitation mixte et dans les cas de parole bruitée. Cette fiabilité se justifie par le fait que le rapport S/N est grossièrement proportionnel à l'amplitude du signal dans un intervalle relativement petit et donc est élevé au voisinage des maxima locaux du signal.

4.5 Décision voisé non-voisé

Une décision parfaite de silence/non-voisé/voisé ne peut être obtenue que par utilisation de l'information sur la nature du bruit environnant, le résultat du processus de reconnaissance de phonème et le contexte du discours. Notre objectif principal étant de tester la stratégie d'estimation de la fréquence fondamentale, certains paramètres dérivés de l'algorithme lui-même sont utilisés pour faire la décision. Nous montrons dans la section 7 que pour la parole non-bruitée, ils donnent des résultats satisfaisants.

Le silence est identifié simplement en mesurant l'énergie moyenne normalisée à la longueur de la fenêtre d'analyse et le rapport de l'énergie à basse fréquence sur l'énergie à haute fréquence. L'énergie moyenne $E_a(s)$ est donnée par

$$E_a(S) = \frac{1}{L}\sum_{i=0}^{L} |S(h \times i)|^2 \tag{3.34}$$

où L est le plus grand entier approchant N/h. Au lieu d'utiliser tous les échantillons

$$S(0), S(1), ..., S(N-1),$$

nous utilisons

$$S(0), S(h), ..., S(h \times L)$$

afin d'accélérer le calcul. Avec une valeur de h convenable (4 à 8 pour une fréquence d'échantillonnage 10kHz), comparé à $h = 1$, peu d'écart est introduit dans le résultat car les échantillons utilisés correspondent encore à un sous-espace statistiquement représentatif de l'espace d'origine grâce à leur grand nombre et car il existe des perturbations inhérentes entre chaque période.

En parole réelle, surtout en parole continue, il n'existe pas en fait des critères universels permettant de séparer les sons voisés des sons non-voisés et vice versa. Ceci est en grande partie parce que

- la parole est une version lissée de phonèmes articulés isolément et
- certains phonèmes peuvent avoir des réalisations continuellement variables.

Une considération importante lors de la décision voisé/non-voisé est que la décision finale doit approcher celle faite par l'expert humain qui examine le signal de parole. Plusieurs paramètres dérivés de la parole peuvent servir individuellement comme indication pour cette décision mais aucun n'est capable de fonctionner de façon satisfaisante dans toutes les situations. Pour éviter des erreurs provoquées par une utilisation séparée de ces paramètres, plusieurs paramètres obtenus pendant le processus d'estimation sont combinés pour prendre la décision.

Parmi ces paramètres, celui qui est le plus important est le maximum de la fonction de ressemblance qui est atteint lorsque le meilleur appariement entre le signal de parole et le signal pseudo-périodique présumé est obtenu. Cette valeur donne le plus grand dégré de ressemblance sous les contraintes données et peut donc être considérée comme une mesure de périodicité discriminante. Elle est aussi une bonne indication de la fiabilité lorsque la fenêtre d'analyse est estimée voisée.

Les impulsions glotales contiennent des harmoniques riches. Par conséquent, pour les signaux produits par l'excitation périodique, le maximum de chaque période V_m est beaucoup plus grand que la moyenne des valeurs absolues V_a dans la période. Le rapport des deux valeurs V_a/V_m est donc utilisé comme un facteur de décision. Ce rapport est grand lorsque le signal dans la fenêtre d'analyse est un segment voisé. Nous discutons l'estimation du maximum de chaque période moyennée dans une fenêtre au paragraphe 4.7.

L'examen des spectres de parole montre que pour l'excitation périodique ou mixte, l'énergie du signal est répartie dans les basses fréquences. Ceci suggère que le pourcentage de l'énergie des composantes à basse fréquence comparé à l'énergie totale peut être significatif pour dintinguer des segments voisés. Ce paramètre est un sous-produit du processus d'estimation que nous discutons dans 4.8.

4.6 Vérification de la périodicité

Le processus d'appariement rejette en pratique les échantillons du signal pour lesquels la fonction $f(n)$ est nulle, si la fonction de fenêtre est utilisée pour la construction de la séquence pseudo-périodique présumée. Grâce à cette propriété, le temps de calcul est énormément réduit mais la méthode présente une tendance légère à classer un segment non-périodique comme périodique pour certains signaux, au demaurant rarement rencontrés en parole. Cela est due au fait qu'une partie du signal peut être masquée par la fonction de fenêtre. Afin de surmonter ce problème, nous adoptons une vérification de périodicité. Dans cet objectif, une fonction dite AMDF, variable en fonction du temps (time-dependent AMDF (TDAMDF)), est évaluée aux points donnés par l'ensemble de périodes p. Cette TDAMDF est définie comme

$$TDAMDF(\mathbf{p}) = \frac{1}{T_m + 1}\sum_{n=0}^{T_m} |S(n) - S(n + P(n))| \tag{3.35}$$

où T_m est le plus grand entier qui satisfait

$$T_m + P(T_m) < N. \tag{3.36}$$

Pour éviter des calculs complexes et avoir une précision suffisante, P(n) est définie comme

$$P(n) = \sum_{i=0}^{I-1} [p_{i+1} g(p_{i+1}, n - \sum_{j=1}^{i} p_j)] \tag{3.37}$$

La périodicité est rejetée si $TDAMDF(\mathbf{p})$ dépasse un seuil prédéterminé par expérience.

4.7 Evaluation de la fonction de ressemblance

L'introduction de la séquence de fonctions de fenêtre permet de simplifier le calcul de la fonction R. En utilisant la relation Eq-3.9, l'expression de la fonction R dans Eq-3.26 peut se réécrire en

$$R = \frac{\sum_{n=0}^{N-1} [s(n)(g(T,n) * \delta_p(n - N_0))]}{\sum_{n=0}^{N-1} [A(n)(g(T,n) * \delta_p(n - N_0))]} \tag{3.38}$$

Avec la définition de $\delta_p(n)$ de l'Eq-3.6, le numérateur se réécrit en

$$R_n = \sum_{n=0}^{N-1}[S(n)\sum_{i=0}^{I} g(T, n - N_0 - \sum_{j=1}^{i} p_j)] \tag{3.39}$$

ou encore

$$R_n = \sum_{n=0}^{N-1}[\sum_{i=0}^{I}(S(n)\, g(T, n - N_0 - \sum_{j=1}^{i} p_j))] \tag{3.40}$$

De la définition de $g(T, n)$ dans Eq-3.14, nous avons

$$R_n = \sum_{i=0}^{I}\sum_{n=0}^{T-1} S(n + N_0 + \sum_{j=0}^{i} p_j) \tag{3.41}$$

Puisque la somme sur la variable muette n est toujours dans l'intervalle $[0, T-1]$ pour tout i et pour l'ensemble de périodes d'essai $\mathbf{p}$, nous définissons une séquence auxiliaire $Y(n)$ dérivée de $S(n)$ afin d'éviter des calculs repétitifs pour chaque ensemble de périodes différent $\mathbf{p}$

$$Y(n) = \sum_{j=0}^{T-1} S(n + j) \quad avec\ 0 \leq n < N \tag{3.42}$$

R_n devient alors

$$R_n = \sum_{i=0}^{I} Y(N_0 + \sum_{j=0}^{i} p_j) \tag{3.43}$$

ou encore

$$R_n = \sum_{n=0}^{N-1}[Y(n)\, \delta_p(n - N_0)] \tag{3.44}$$

De la même manière, le dénominateur dans R peut se simplifier à la même forme, la somme

$$\sum_{j=0}^{T-1} A(n + j)$$

peut être approchée par

$$T \times A(n + \frac{T}{2})$$

avec une précision suffisante, car l'enveloppe varie beaucoup plus lentement que le signal de parole lui même. En conséquence, la fonction R est réduite en

$$R = \frac{\sum_{n=0}^{N-1}[Y(n)\delta_p(n - N_0)]}{T\sum_{n=0}^{N-1}[A(n + \frac{T}{2})\delta_p(n - N_0)]} \tag{3.45}$$

Le résultat de la multiplication d'une séquence par une série de fonctions de Dirac donne juste les échantillons de la séquence pour l'indice desquels la série de Dirac n'est pas nulle. L'évaluation de Eq-3.45 ne demande donc pas de multiplication et le nombre d'additions nécessaire est aussi considérablement réduit.

La vitesse de variation de période du pitch qui apporte de l'information est limitée par la constante de temps du système glotal. Toujours pour réduire le calcul, ce fait est exploité dans notre implantation par la restriction des relations entre chaque deux périodes successives. Les statistiques sur les signaux de parole montrent que, à l'intérieur d'une fenêtre de longueur normale, la variation de période du pitch peut être considérée comme une constante et la valeur absolue de cette variation est en général environ 10% de la longueur de la période précédente. Nous formulons donc les contraintes sur $\mathbf{p}$:

$$C(\mathbf{p}) = p_i - p_i - 1 - D = 0 \quad 0 < i \leq I \tag{3.46}$$

avec $|D| \leq D_{max}$, D_{max} étant la variation maximale autorisée.

En résumé, dans la version implantée de notre méthode, l'évaluation de la fonction de ressemblance est réalisée par la recherche exhaustive du maximum de R donnée dans Eq-3.45 dans un espace défini par

$$P_{min} \leq p_i \leq P_{max} \text{ et } |D| \leq K \times p_i \quad 0 < i \leq I. \tag{3.47}$$

où K, la variation relative entre deux périodes successives détectable, est une constante. $K \times p_i$ est choisi en général inférieur à 10% de p_i ($0 \leq K \leq 0.1$) et P_{min} et P_{max} sont respectivement la plus grande et la plus petite période détectable sous la fréquence d'échantillonnage donnée. Une certaine pondération est incorporée dans le processus de la recherche du maximum. La procédure ne demande pas d'autre stockage à l'exception de quelques buffers du type entier.

La séquence $Y(n)$ contient des composantes de basses fréquences de $S(n)$ et possède la répartition spectrale désirée. Le rapport d'énergie pour la décision voisé/non-voisé décrit dans 4.5 peut se calculer par

$$E_n = 1 - \frac{E_a(Y)}{E_a(S)} \tag{3.48}$$

4.8 Position et amplitude des impulsions

Notre processus d'estimation présente la particularité, c'est qu'après le processus d'appariement, la position initiale N_0 et les intervalles de temps entre les impulsions successives donnés par $\mathbf{p}$ sont complètement connus. La position et l'amplitude de la séquence pseudo-périodique présumée donnent une approximation de la séquence d'excitation de la parole originale. Dans la plupart des applications telles que l'analyse de parole synchrone à la période du pitch, c'est la position relative et non la forme exacte des impulsions qui est utilisée. Ces paramètres peuvent être utilisés pour

- la reproduction automatique de la parole avec une vitesse d'articulation variable tout en conservant la période du pitch de la parole originale et
- l'analyse de la parole synchrone à la fréquence fondamentale.

En ce qui concerne l'amplitude moyenne à court terme dans une fenêtre d'analyse, les formules suivantes donnent une approximation

$$AMP = \frac{1}{I \times T}\sum_{i=0}^{I} Y(N_0 + \sum_{j=0}^{i} p_j) \tag{3.49}$$

ou également

$$AMP = \frac{1}{I \times T} \sum_{n=0}^{N-1} [\delta_p(n - N_0)\, Y(n)] \tag{3.50}$$

où $Y(n)$ est détaillé dans 4.7. AMP peut servir comme une mesure supplémentaire utile pour la décision voisé/non-voisé, la discrimination voyelle-consonne et l'extraction de propriétés du signal de parole.

4.9 Fonction de pondération sur $R(n)$

Quand le nombre de périodes dans une fenêtre est grand, $R(n)$ peut ne pas atteindre sa valeur maximale du fait de contraintes sur la relation entre les p_i. Ceci est particulièrement important lorsque la fréquence fondamentale est élevée ou la longeur de fenêtre grande. Nous avons utilisé une ligne droite dont la pente est spécifiée pour réaliser la pondération.

4.10 Précision de l'estimation

Etant données les périodes p obtenues par la maximisation de la fonction R, la fréquence fondamentale estimée est calculée par

$$F_i = \frac{F_e}{p_i} \tag{3.51}$$

où F_e est la fréquence d'échantillonnage . Une incrémentation Δp_i de p_i provoque l'incrémentation ΔF_i de F_i :

$$|\Delta F_i| = \frac{F_i^2}{F_e} |\Delta p_i| \tag{3.52}$$

Si aucune interpolation n'est utilisée la précision de l'estimation de p_i peut atteindre un intervalle d'échantillonnage du signal. La précision de l'estimation de fréquence correspondante est donc

$$|\Delta F_i| = \frac{F_i^2}{F_e} \tag{3.53}$$

Par exemple, la résolution théorique de F_i à 100Hz sous une fréquence d'échantillonnage de 10kHz est à peu près 1Hz, équivalente à celle du système auditif de l'homme [Terhardt 82].

5 Comportement fréquentiel

La formulation de l'estimation de la fréquence fondamentale proposée peut être expliquée de différents points de vue du traitement du signal dont le plus important, qui revèle les propriétés de base de *EFFASH*, est l'interprétation du comportement dans le domaine fréquenciel. Nous limitons notre discussion dans un cas simplifié (K=0) et seul le signal stationnaire est considéré.

Dans l'expression de la fonction R dans l'Eq-3.38, le rôle du dénominateur peut être assimilé à l'introduction d'un effet de normalisation de manière que R soit indépendant du

nombre des impulsions synthétisées lorsque le signal est stationnaire. Par conséquent notre attention peut être focalisée sur le numérateur.

Puisque K=0, les p_i, $0 < i \leq I$ sont égaux à l'intérieur d'une fenêtre d'analyse. Soit $p_i = p$, $0 < i \leq I$. La caractérisque de $g(T,n) * \delta_p(n - N_0)$ dans le domaine fréquentiel est

$$\mathcal{F}[g(T,n) * \delta_p(n - N_0)] = |\sin(\omega T/2)/(2\sin(\omega p/2)\sin(\omega/2))| e^{j\theta(\omega,p,N_0,T)} \tag{3.54}$$

Un signal de parole stationnaire à l'excitation périodique avec une période P peut être exprimé par la convolution

$$S(n) = \delta(n \bmod P) * h(n) \tag{3.55}$$

où $h(n)$ est le système de transmission décrit dans la section 3. La transformée de Fourier du signal de parole $S(n)$ est donc donnée par

$$\mathcal{F}[S(n)] = \mathcal{F}[\delta(n \bmod P)]\, H(j\omega) = |H(j\omega)/sin(\omega P/2)| e^{j\psi(\omega)} \tag{3.56}$$

En appliquant le théorème de Parseval [Oppenheim 75], nous avons

$$\sum_{n=0}^{N-1} [S(n)[g(n) * \delta((n - N_0) \bmod p)]] \tag{3.57}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\mathcal{F}[S(n)]| |\frac{sin(\omega T/2)}{sin(\omega/2) sin(\omega p/2))}| e^{-j\phi(\omega)}\, d\omega \tag{3.58}$$

avec $\phi = \theta - \psi$. La phase $\phi(\omega, p, P, T)$ est une fonction impaire de ω car $S(n)$ et $\Pi(g,n)$ sont toutes les deux des séquences réelles [Oppenheim 75]. La partie droite de Eq-3.58 se réduit donc en

$$\frac{1}{\pi} \int_0^{\pi} |\mathcal{F}[S(n)]| |\sin(\omega T/2)/(\sin(\omega/2))| \frac{1}{|sin(\omega p/2)|} \cos(\phi)\, d\omega \tag{3.59}$$

L'Eq-3.56 montre qu'une série de maxima locaux ayant pour période

$$\omega = 2k\pi/P \tag{3.60}$$

apparaissent dans le spectre du signal lorsque le signal est périodique avec la période P. Cependant, dans l'Eq-3.59, si

$$\omega = 2k\pi/p, \tag{3.61}$$

la fonction dont on calcule l'intégration tend vers l'infini aux points $\omega = 2k\pi/p$ d'une manière semblable à un peigne à cause du troisième facteur. Sous cette condition, l'intégrale qui représente l'énergie totale prend une valeur très grande. Puisque le processus d'estimation ajuste continuellement la période d'essai p, un maximum de l'Eq-3.59 sera atteint lorsque tous les maxima locaux de la série harmonique dans le signal présumé apparient correctement ceux du spectre du signal d'entrée. Autrement dit, le spectre, sous forme d'un peigne, de la séquence d'impulsions présumée approxime de façon optimale le spectre du signal de la parole. L'apparition de valeurs infinies est seulement dans le cas idéal et en réalité les pics du peigne sont déaccentués par un effet de convolution. Cet effet est produit lorsque le signal de parole est tronqué par $g(N,n)$ à cause du choix de la longueur de fenêtre d'analyse. La période p_o qui domine le processus d'appariement des harmoniques est affectée comme la

période du pitch quand le maximum de l'intégrale indiquant l'énergie maximale est obtenu. L'estimation correcte du délai initial N_0 donne une valeur convenable de $cos(\phi)$ pour assurer le plus grand maximum possible.

Le facteur $\sin(\omega T/2)/\sin(\omega/2)$ dans l'Eq-3.59 introduit par l'application de la fonction de fenêtre $g(T,n)$ modifie le spectre de la séquence de fenêtres de manière que le spectre résultant ait une amplitude décroissante lorsque ω s'approche la demi-fréquence d'échantillonnage . C'est cette propriété qui permet que la recherche de structures harmoniques s'effectue principalement dans les basses fréquences du spectre du signal de parole. La sélection soigneuse de la largueur de la fonction de fenêtre T fournit une structure spectrale de peigne qui comprend les premiers 6-8 maxima du spectre du signal. Ce nombre est approprié

- à l'assurance d'une estimation fiable car seules les harmoniques de faibles ordres (inférieurs à 10) sont effectives au transport de l'information sur le pitch [Goldstein 78] [Sluyter 82], et
- à l'atténuation du bruit dans la bande de fréquences plus élevée donc à l'élimination des erreurs provoquées par la perturbation du bruit blanc ou par l'excitation aléatoire de glotte.

En parole réelle, la non-stationnarité en période élargit les pics du spectre alors que la non-stationnarité en amplitude, telle que modélisée précédemment, introduit un effet de convolution similaire à celui introduit par la troncature. Par conséquent, les discussions précédentes peuvent être généralisées dans le cas du signal non stationnaire.

6 Algorithmes

L'algorithme général de la détermination de la fréquence fondamentale se résume de la façon suivante:

```
F0Estimation
    CréationEnveloppe;
    MatchingSignal;
    VérificationAMDF;
    CorrectionPériode;
    MoyennePeriode;
    DécisionV-UV-SIL;
```

Nous donnons l'algorithme de recherche de la meilleure correspondance entre le signal synthétisé et le signal de test pour lequel on estime la fréquence fondamentale. La fonction MatchingSignal détermine la période i qui donne la plus petite erreur de la différence entre deux vecteur Sig et Env.


```
Sig = le signal de test (le surface): tableau d'entiers
Env = l'enveloppe du signal de test: tableau d'entiers
δ = incrémentation maximale de période autorisée: entier
Popt = la période optimale estimée: [Nbas,Nhaut]
Vopt = la meilleure vraisemblance de l'estimation: [0,1.0]
Imax = l'indice pour lequel le signal de test éteint son maximum: entier
N = la longueur de la fénêtre d'analyse: entier
δr = la meilleure incrémentation estimée: réel
Nbas = la plus petite période calculable: constante
Nhaut = la plus grande période calculable: constante
Rdescent = la pente de pénalisation de basses fréquences: [0,1.0] constante
Fact = l'échelle pour que le calcul se fasse en entier: constante
```

```
MatchingSignal(Sig,Env,Imax,N,δ) → (Popt,Vopt,δr)
    Popt = Nbas;
    DesMove = Fact;
    δr = 0;
    Vopt = 0;
    Descent = Rdescent × Fact / (Nhaut-Nbas+1);
    Pour i de Nbas à Nhaut Faire
        (Ri,Di) = PériodeOpt(Sig,Env,i,i/δ,Imax,N,DesMove);
        Si Ri > Vopt
            Alors Vopt = Ri; δr = Di; Popt = i;
        FinSi;
        DesMove -= Descent;
    FinPour;
```

La fonction PériodeOpt cherche la meilleure incrémentation d'une période dans une fenêtre d'analyse. Elle retourne un couple composé de la meilleure vraisemblance et de l'incrementation.

```
i = la période d'essai
Dr = le maximum de l'incrémentation pour la période i
PériodeOpt(Sig,Env,i,Dr,Imax,N,DesMove) → (Ri,Di)        Rpj = 0; dj = 0;
    Pour j de -Dr à Dr Faire
        SurfP = Sig[Imax]; SurfT = Env[Imax];
        MoveP = i; Move = Imax + MoveP;
        TantQue Move < N Faire
            SurfP += Sig[Move]; SurfT += Env[Move];
            MoveP += j; Move += Movep;
        FinTantQue;
        MoveP = i - j; Move = Imax - MoveP;
        TantQue Move > 0 Faire
            SurfP += Sig[Move]; SurfT += Env[Move];
            MoveP -= j; Move -= Movep;
        FinTantQue;
        Rp = SurfP × DesMove / SurfT;
        Si Rp ≥ Rpj
           Alors Rpj = Rp; dj = j;
        FinSi;
    FinPour;
    (Rpj,dj);
```

7 Expérimentation

Notre algorithme a été programmé en Pascal pour évaluer ses performances. Dans toutes les expériences, la fréquence d'échantillonnage est 10kHz. Le rapport du signal sur bruit SNR utilisé est défini comme le rapport de l'énergie moyenne du signal sur la puissance du bruit:

$$SNR = 10 \times \log_{10} \frac{P(S)}{P(N)} \tag{3.62}$$

où $S(n)$ est la séquence du signal, $N(n)$ la séquence du bruit et

$$P(X) = \frac{1}{L} \sum_{n=0}^{L-1} X^2(n) \tag{3.63}$$

l'énergie moyenne du signal. Le bruit utilisé dans les expériences, sauf indication spéciale, est un bruit pseudo-blanc réparti uniformément dans l'intervalle $[-T_n, T_n]$. Le schéma du traitement par fenêtre de notre méthode implique l'observation du signal sur un intervalle de temps suffisament grand pour fournir l'information suffisante sur la périodicité et aussi suffisament petit pour que le modèle soit valable. Une longueur de fenêtre de 38ms est utilisée avec une superposition de 28ms entre les fenêtres successives, sauf précision contraire. Aucune tentative n'est faite pour interpoler les valeurs estimées. La largueur de la fonction de fenêtre T a été fixée à 8 échantillons pour tous les tests. Aucune fenêtre de pré-pondération du signal n'est utilisée. Tous les contours de valeurs de pitch sont tracés simplement par impression

des valeurs estimées indépendamment. Aucun algorithme de lissage ni connaissance de haut niveau (par exemple l'information contextuelle) ne sont utilisés pour corriger des erreurs. Dans les contours de pitch présentés, les zéros désignent le silence et les valeurs de 20Hz indiquent un segment non voisé.

Les expériences sont conduites successivement avec deux valeurs de l'incrémentation de période K: (a) K=0 et (b) K=0.1, en utilisant le même algorithme. Nous avons constaté que:

- La période de pitch estimée sur une fenêtre d'analyse est presque toujours la même pour le cas (a) et le cas(b). Ceci peut être expliqué par la propriété suivante. Le processus d'appariement est une optimisation globale sur l'intervalle d'observation entier et termine toujours le placement de chaque impulsion de la fonction de fenêtre à proximité du maximum de chaque période pour que la fonction R soit maximisée sous les contraintes données. Cette propriété moyenne automatiquement les périodes lorsque le signal de parole est périodique de façon dépendante du temps quand l'incrémentation K est zéro. Il est clair cependant que dans le cas K=0 l'estimation de la position du pic maximal de chaque période n'indique pas précisément la position réelle concernée;
- La fonction R a une valeur plus grande pour le cas (b) que pour le cas (a). Comme nous avons décrit dans la section 4.7, quand $K \neq 0$, l'optimisation est effectuée dans un espace à deux dimensions. Puisque un dégré de liberté est disponible, il n'est pas surprenant qu'un meilleur appariement puisse être obtenu. La différence de R pour les deux cas est d'autant plus grande que la variation de la valeur de pitch est plus rapide.
- Par conséquent, dans le cas $K \neq 0$, si les périodes de pitch estimées sont moyennées sur la fenêtre d'observation, peu de différence significative peut être observée dans les contours pour (a) et (b).

Grâce à ces observations, il n'est pas nécessaire de lister et discuter toutes les expériences pour les deux cas. Dans les paragraphes qui suivent, seuls les résultats pour K=0 seront présentés. Certains points particuliers concernant le cas $K \neq 0$ seront discutés dans 7.2.

7.1 Expérience sur des signaux synthésés

La première partie de nos expériences a pour objectif de montrer, par des simulations idéalisées, les propriétés de base de notre méthode d'estimation.

Vérification du Modèle

Notre méthode peut être considérée comme une réalisation de *EFFASH* dans le domaine du temps. Il est donc naturel que la méthode soit entièrement capable de fonctionner dans des situations où la première harmonique de la fréquence fondamentale est complètement absente. La méthode est aussi capable de traiter l'enveloppe à variation dépendante du temps du signal. Afin de vérifier ces propriétés, un signal donné par

$$S(n) = 3n(\sin((2\pi \times 400)n) + \sin((2\pi \times 500)n))$$

est synthétisé et correspond aux deux premières harmoniques successives (la quatrième et la cinquième) d'un signal à la fréquence perceptible de 100Hz et à une enveloppe linéairement croissante. La séquence de fonctions de fenêtre a correctement apparié avec les maxima de chaque période du signal. Il y a aucune ambiguïté dans la recherche du maximum de R et la fréquence estimée est 100.00Hz.

Immunité des Bruits

Afin de tester les performances dans un environnement très bruité, un bruit avec T_n = 1000 a été ajouté à un signal sinusoïdal de 200.00Hz. L'amplitude du signal était 200. Le SNR obtenu est -12.2 dB. Sous ce SNR il est difficile de dire, en regardant la courbe, si le signal resultant est périodique. Une décision "voisé" était forcée pour donner une valeur de période car dans ce cas la décision n'est plus significative. Il est à noter que ce signal pourrait être jugé comme bruit par l'oreille humaine. Néanmoins, même dans ce cas, une période de 196.1Hz a été estimée. Cette valeur correspond à une déviation d'un échantillon dans l'estimation de la fonction R. Des résultats semblables ont été obtenus en utilisant un bruit blanc gaussien avec le même SNR. L'augmentation de la largueur de la fonction de fenêtre T peut aider à diminuer l'influence des bruits et donc à améliorer le comportement dans des applications où le bruit d'ambiance est important. Notre méthode a une bonne immunité au bruit blanc parce que le bruit est largement attenué par l'évaluation des doubles sommes des Eq-3.42 et Eq-3.45. En plus, seuls les pics du signal au voisinage des quels le SNR est relativement élevé sont pris en considération lorsque le meilleur appariement est réalisé. Comme exemple, pour T=8 à 10kHz de fréquence d'échantillonnage et pour un signal sinusoïdal pur à f=100Hz, l'amélioration de SNR est environ 3dB après l'appariement. Puisque, en général, le signal voisé contient une grande quantité d'harmoniques, une amélioration plus importante peut être espérée dans des applications réelles. Cette propriété dans l'environnement bruité est rarement rencontrée dans les travaux existants traitant de l'estimation de la fréquence fondamentale.

Effet du Bruit sur la Précision

Pour étudier l'influence du niveau de bruit sur la précision de la fréquence estimée, un signal sinusoïdal de 150Hz a été utilisé comme signal d'origine. Son amplitude était 400 et la longueur du fichier de signal était de 60 blocs (1 bloc = 128 points). 8 fichiers étaient créés pour obtenir des SNR différents. La largueur de fenêtre d'analyse était 450 échantillons. La superposition entre deux fenêtres successives était de 363 échantillons. Il y avait ainsi 88 estimations pour chaque fichier de test. La moyenne et la déviation standard ont été ensuite calculées à partir de ces estimations pour chaque fichier. Le résultat présenté dans la Tableau 3.1 illustre la bonne précision fréquentielle de notre algorithme dans des situations bruitées. On peut observer notamment que la moyenne est presque toujours la même que la fréquence vraie du signal original et que la déviation est proche de la résolution théorique de la méthode. Puisque dans la production de la parole normale l'excitation glottale est toujours perturbée et la limite supérieure de cette perturbation est d'environ 3% [Lieberman 63] et puisque la résolution de la fréquence du son de l'être humain est d'environ 3Hz à 100Hz [Goldstein 78], ces déviations sont entièrement tolérables pour des applications normales.

Signal	SNR(dB)	Tn	m(Hz)	σ	σ/m (%)	$Gross_{er}$
S15B40	40	5	150.1	1.46	0.97	0
S15B20	20	49	150.2	1.48	0.99	0
S15B10	10	154	150.6	1.94	1.29	0
S15B05	5	276	150.3	2.26	1.50	0
S15B03	3	347	150.4	2.68	1.78	0
S15B00	0	490	149.9	2.99	1.99	0
S15BN3	-3	692	150.6	3.14	2.08	0
S15BN6	-6	978	150.2	3.66	2.43	0

Table 3.1: Relation entre le niveau du bruit et la précision de la détection de la fréquence (sin f=150Hz)

Note: m: valeur moyenne, σ : écart type, $Gross_{er}$: nombre des estimations pour lesquelles l'inégalité $|F_t - F_\epsilon|/F_t > 0.05$ est vérifiée. (F_t : vraie fréquence, F_ϵ : fréquence estimée)

La relation 3.53 montre que l'augmentation de la fréquence d'échantillonnage peut améliorer la précision, en demandant plus de calcul. Une interpolation convenable peut être aussi utile.

7.2 Expériences sur la parole réelle

Cette partie des expériences utilise le signal de parole réel. Les évaluations sont basées sur 10 minutes de parole collectée dans plusieurs corpus de parole. 30 loculeurs sont intervenus. Nous avons d'abord effectué une évaluation détaillée de l'algorithme sur un corpus de petite taille contenant 16.3 secondes de parole constituant 8 phrases françaises phonétiquement equilibrées prononcées par 2 locuteurs mâles. Les expériences étaient conduites successivement pour le signal de parole propre, la parole filtrée par une ligne téléphonique simulée, et la parole hautement bruitée. Les résultats sont présentés respectivement dans les paragraphes qui suivent, et résumés dans la Tableau 3.2.

Nous avons ensuite testé l'algorithme dans un contexte multi-locuteurs. Nous commenterons les résultats qui sont présentés dans les Tableaux 3.4 3.5 3.6.

Dans toutes les tableaux que nous donnons par la suite, "Taille" est la taille du fichier de parole mesurée en nombre de blocs contenant 128 points de signal échantilloné, N_ϵ est le nombre d'estimations dans chaque contour de F_0, "PHRASE" référencie le fichier de parole, "$F0_{er}$" est le nombre d'erreurs de l'estimation et "$V-UV-S_{er}$ " est le nombre d'erreurs dans les décisions de Voisé/Non-voisé/Silence. Sauf définie autrement ($Gross_{er}$ dans le Tableau 3.1), une erreur de F_0 est comptée chaque fois qu'un segment de parole est estimé comme voisé et que la fréquence vraie F_t et la fréquence estimée F_ϵ satisfont

$$|F_t - F_\epsilon|/F_t > 0.1.$$

L'enregistrement du corpus de petite taille était réalisé dans une salle de console avec un

Phrase	Taille	N_e	$F0_{er}/V - UV - S_{er}$					
			propre	filtrée (0.3-2kHz)	bruitée			
					uniforme		Gaussian	
					4.2dB	-1.7dB	5.5dB	0.dB
TIR	143	183	0/0	0/2	0	0	5	8
LEC	204	256	0/0	0/7	2	3	7	11
LET	131	167	0/0	3/1	0	0	4	6
CHA	170	217	0/0	0/9	1	2	2	6
ILS	144	184	0/0	3/6	2	0	5	2
UNE	188	240	0/0	1/4	2	6	7	11
JEM	151	193	0/0	3/7	1	2	2	4
AIN	142	181	0/0	2/2	1	1	5	6
Total	1273	1621	0/0	12/38	9	14	37	54
Erreur %			0/0	0.74/2.3	0.56	0.86	2.3	3.3

Table 3.2: Résults sous conditions de parole: propre, filterée et bruitée (uniforme et Gaussian)

Note: N_e est défini antérieurement. Pour la parole bruitée seul $F0_{er}$ est donné.

niveau de bruit d'ambiance normal. Le signal analogique était filtré par un filtre pass-bande de 80-4700Hz et digitalisé avec 10bits de dynamique. Le SNR résultant de la parole "propre" est inférieur à 30dB. La parole digitalisée utilisée pour le test multilocuteurs était enregistrée à travers un système de transmission de son de haute qualité.

Il est difficile de définir en accord avec la perception humaine les erreurs d'estimation des contours de la fréquence fondamentale de parole réelle. L'évaluation est encore plus compliquée lorsqu'il s'agit de faire une décision voisé/non-voisé qui est quelquefois par nature ambigüe et ne peut pas être definie de façon consistante dans certaines évolutions de combinaisons de phonèmes. Dans notre expérience, une comparaison de chaque résultat d'estimation, la valeur de F_0 et la décision, avec la périodicité de l'onde associé dans le domaine temporel a été effectuée sur un terminal graphique en utilisant notre éditeur du signal ASSIA [Gong 85b] présenté dans le chapitre 2. C'est un travail fastidieux mais c'est une solution plus raisonnable.

Tous les paramètres et seuils à l'exception de la décison de silence dans l'expérience sur la parole enregistrée dans une salle de consoles avait été ajustés soigneusement seulement pour la prole propre avant que les expériences soient commencées.

Parole propre

Nous donnons les contours de fréquences estimés pour deux phrses (TIR et LEC) correspondant à la parole propre dans Fig.3.2 (a) et Fig.3.3 (a).

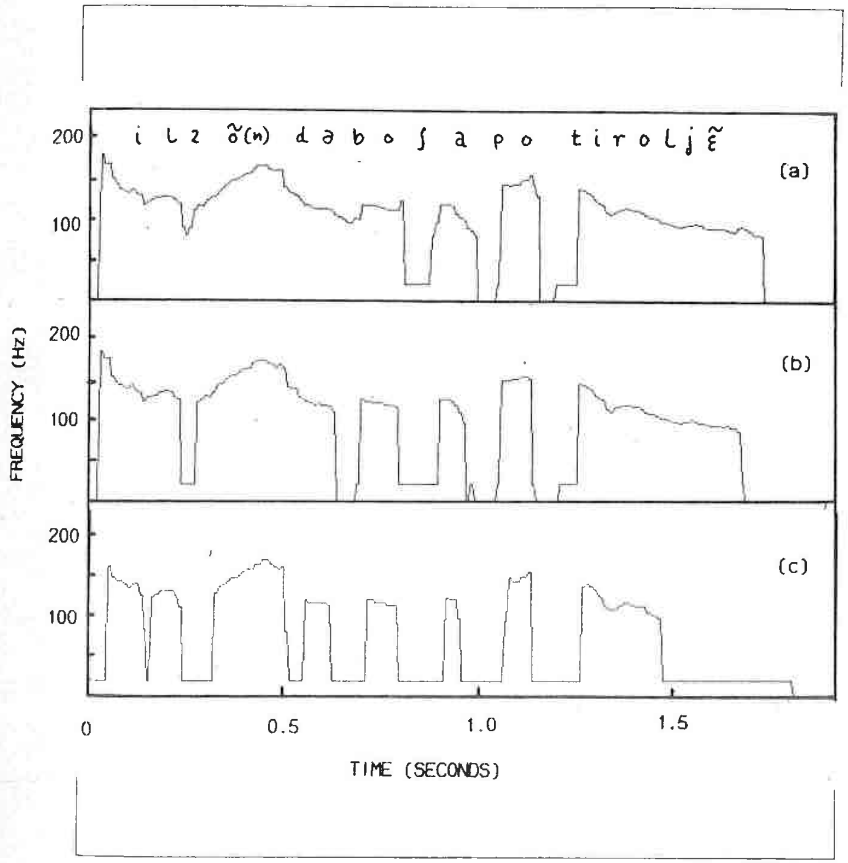


Figure 3.2: Contours de F_0 de la parole (a) propre (SNR $\simeq$ 30dB), (b) filtrée (0.3-2kHz) et (c) bruitée (SNR $\simeq$ -1.7dB) extraite de la phrase "TIR" (K=0)

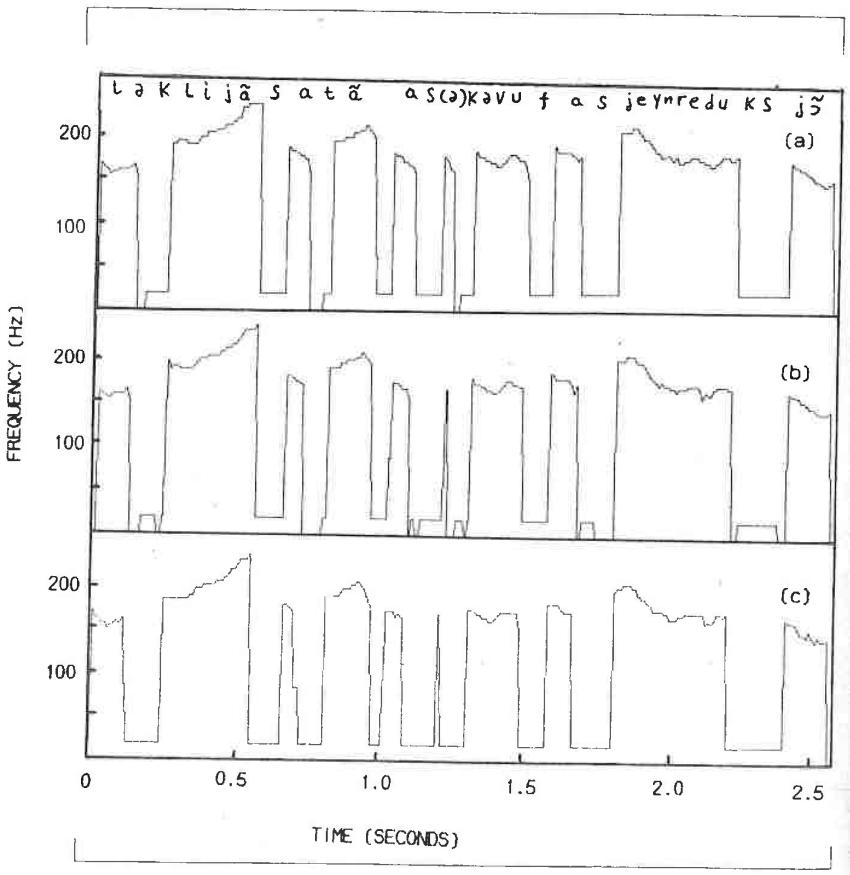


Figure 3.3: Contours de F_0 de la parole (a) propre (SNR $\simeq$ 30dB), (b) filtrée (0.3-2kHz) et (c) bruitée (SNR $\simeq$ -1.7dB) extraite de la phrase "LEC". (K=0)

Les estimations semblent sans erreur en ce qui concerne les contours fréquentiels. Il est à noter que la méthode montre une capacité d'adaptation de l'enveloppe du signal et peut suivre de façon satisfaisante les changements rapids sur l'énergie moyenne à courte terme pour une grande variété de phonèmes. L'un des problèmes les plus difficiles rencontrés dans l'estimation de la fréquence fondamentale est donc éliminé par notre algorithme. Il est remarquable qu'il n'y ait pas de points isolés estimés comme voisés dans les zones non-voisées ou de silence ni estimés comme non-voisés dans les zones voisées. Ceci revèle que les décisions successives sont assez consistente, – une propriété très recherchée dans la décision. Les contours de valeurs de la fonction R de l'Eq-3.45 et du rapport d'énergie E_n définie dans l'Eq-3.48 dérivés du processus de l'estimation pour la phrase "LEC" sont tracés sur les Fig.3.4 (a) et 3.4 (b).

En les comparant avec les contours de fréquences correspondants on peut conclure que ce sont de bons indicateurs de la décision voisé/non-voisé. Les transitions brutales dans ces courbes correspondants aux différentes combinaisons de transition de signal de parole entre voisé, nonvoisé et silence indiquent que le seuilliage n'est pas difficile et par conséquent la décision voisée-nonvoisée peut être obtenue assez fiablement pour les signaux des différents sons.

Les phrases du corpus de test comprennent des combinaisons différentes de phonèmes et les signaux correspondants contiennent des segments pour lesquels l'estimation de la fréquence fondamentale est difficile pour les détecteurs classiques:

- le commencement brutal des voyelles (/pa/ , /sa/, /po/, etc.),
- l'excitation irrègulière (/tr/, /pr/, /li/, etc.),
- les plosives voisés ayant une énergie faible (/b/, /d/, etc.),
- l'onde hautement asymétriques (/n/, /m/, etc.),
- les phonèmes ayant le premier formant fort de telle manière que l'amplitude du 3-ème ou 4-ème ou à la fois 3-ème et 4-ème harmonique(s) est très grande (/o/, /a/, etc.)
- l'excitation mixte dans les fricatives voisées (/z/, /v/, etc.).

Dans certaines situations délicates, l'estimation fourniée pourrait être jugée comme incorrecte par l'examination globale du signal de parole et du contour de fréquence. Des traces de quelques ondes qui illustrent le processus d'estimation de façon plus détaillée sont donc nécessaires. La Fig.3.5 (a) et (b) donnent quelques exemples. La Fig.3.5 (a) concernne deux réalisations du phonème français /r/ dans le mot "surprise" (JEM) et le contour de fréquences estimées.

Ce phonème peut avoir des réalisations variables dans de contextes différents et est quelques fois irrégulier et instable. Suivant le contexte, il peut être voisé ou nonvoisé ou provoquer des changement brutaux de la fréquence fontamentale de la parole. Cependant, il a été traité correctement dans notre test. La Fig.3.5 (b) montre une autre réalisation de /r/ extraite du mot "chambre" (ILS). Le détecteur a pu suivre la variation rapide causée par l'irrégularité sévère de l'excitation même dans le cas où K=0. Il est à noter aussi que le

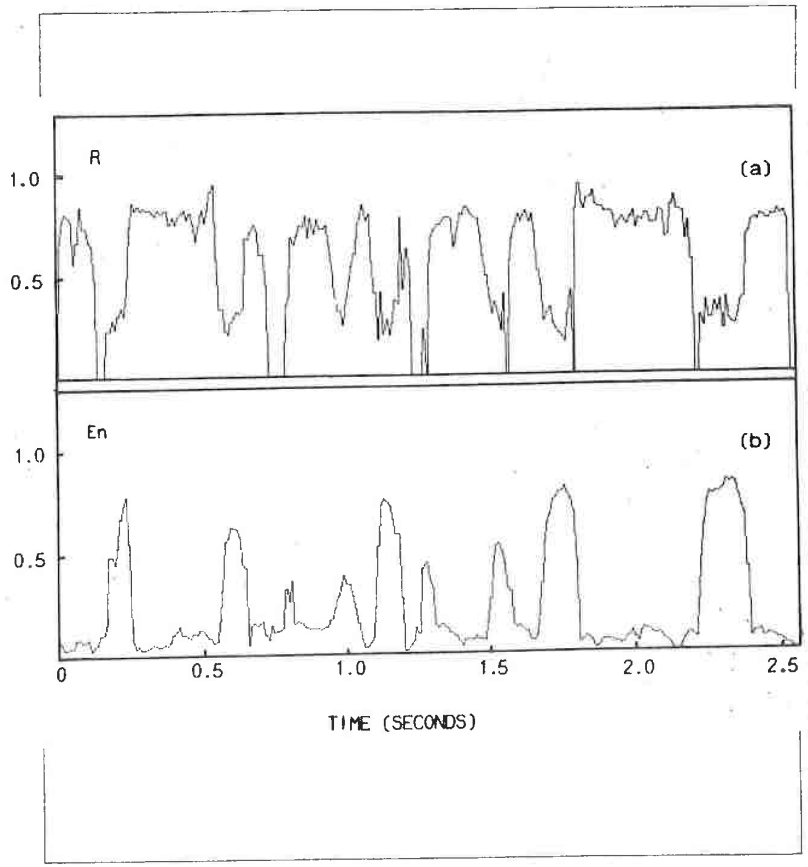


Figure 3.4: Contours (a) de la fonction de ressemblance R (K=0) et (b) du rapport d'énergie E_n de la phrase "LEC".

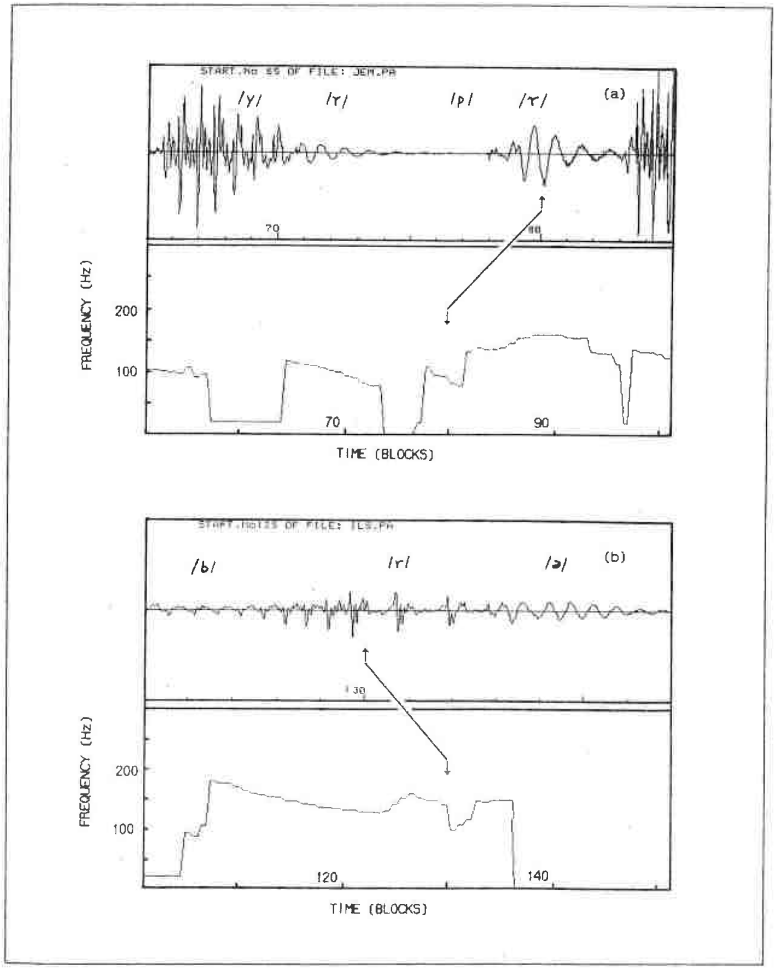


Figure 3.5: Le signal produit par le phonème /r/ et le contour F_0 associé dans le contexte de (a) "surprise" (/syrpriz(∂)/) de la phrase "JEM" et (b) "chambre" (/$\int$a$\tilde{o}$br(∂)/) de la phrase "ILS". (K=0)

schéma de traitement par l'optimisation dans une fenêtre fournit un bon compromis global dans chaque fenêtre. La Fig.3.6 montre la transition du signal de parole d'une voyelle /o/ dans le contexte "beaux". les $A(n)$ et $A_\Pi(g,n)$ correspondents et une courbe à une dimension (K=0) de la fonction R.

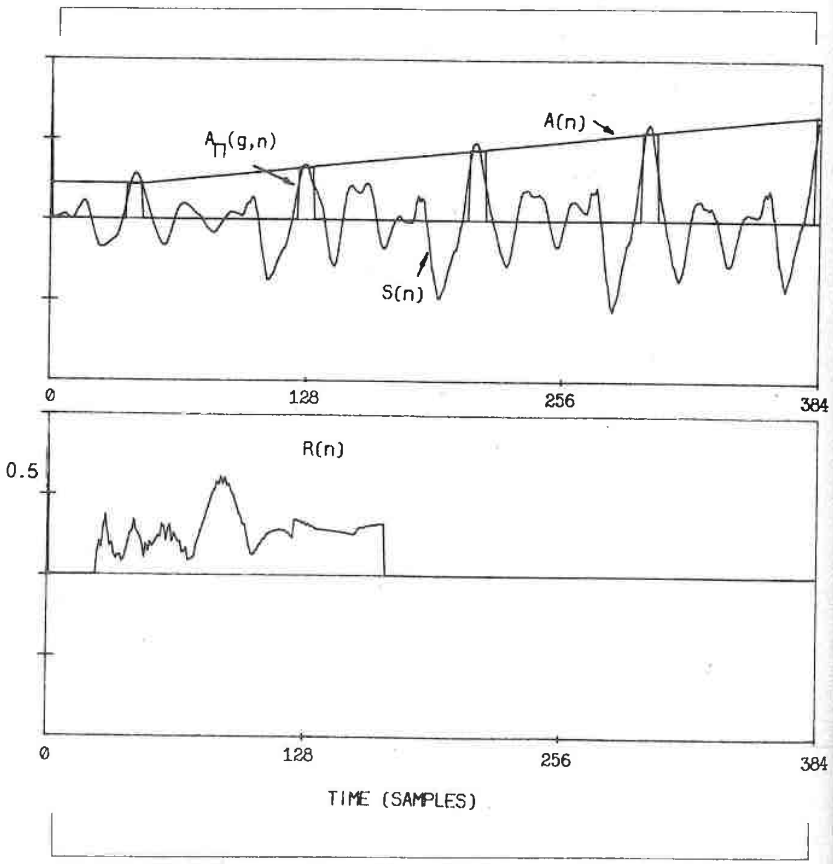


Figure 3.6: Le signal de parole $S(n)$, la fonction d'enveloppe $A(n)$, la fonction de fenêtre $A_\Pi(g,n)$ (en haut) et fonction de ressemblance (en bas) evaluée à K=0 pour le signal de la voyelle /o/ dans le contexte "beaux" (/bo/) de la phrase "TIR".

Phrase	Taille	N_c	$F0_{er}/V-UV-S_{er}$
SAN200	200	256	0/1
SAN400	200	256	2/3
SAN600	200	256	0/1
SAN124	124	158	0/2
Total	724	926	2/7
Erreur %			0.22/0.76

Table 3.3: Résultats pour la parole enregistrée dans une salle d'ordinateurs (rapport signal/bruit $\simeq$ 20dB).

Parole filtrée par ligne téléphonique simulée

Le même matériau de parole a été filtré par un filtre digital de bande passante 0.3-2kHz et une attenuation de 40dB en dehors de la bande pour simuler la caractéristique de transmission d'une ligne téléphonique. La fréquence de coupure supérieure du filtre est inférieure d'environ 1.4kHz de celle de la ligne téléphonique standard. Le signal était ensuite normalisé de manière que l'étendue dynamique reste toujours dans l'intervalle [-512,512]. On peut alors constater que les phonèmes tels que /i/ ont été attenués énormément à cause de la basse fréquence de leur premier formant et la haute fréquence des autres formants. Quand à la périodicité, pour certains phonèmes elle n'est plus si visible que dans les signaux originaux. Les Fig.3.2 (b) et 3.3 (b) donnent les résultats obtenus des mêmes phrases que celles pour les Fig.3.2 (a) et 3.3 (a). Comme on peut le constater dans la Tableau 3.2, la baisse de performance en estimation de F_0 est très faible. La plupart des erreurs de décision sont provoquées par la classification de non-voisé comme silence. Par exemple, certaines segments fricatifs comme /f/ or /s/ dont l'énergie est concentrée en hautes fréquences sont classés comme silence. Ceci est due d'abord à la faible énergie après le filtage et aussi à la procédure de détection de silence utilisée. Nous avons constaté qu'un réajustement leger des seuils de décision peut équilibrer les erreurs de décision et donc donner de meilleurs résultats en présence de la répartition du spectre du signal modifiée. Une autre observation est que la trace de la fonction R ne montre qu'une petite différence entre les signaux avant et après le filtrage. Nous concluons que l'algorithme est capable de fonctionner dans des conditions de transmission variées.

Parole enregistrée dans une salle d'ordinateur

Pour cette partie de l'expérience, le signal a été échantillonné dans une salle de console et le SNR mesuré était d'environ 20dB. Pour avoir une décision correcte sur le silence, le seuil de silence a été réajusté de manière adéquate. Au total, la parole analysée est de 9.3 secondes. Dans les contours obtenus il n'y a pas d'erreurs apparentes. Les résultats sont résumés dans le Tableau 3.3.

Ces résultats sont comparables à ceux obtenus pour de la parole propre, mise à part une

légère dégradation de décision. Les erreurs de la décision sont principalement les erreurs de classification de silence en non-voisé. En présence de la fluctuation d'énergie du bruit, étant donné que nous utilisons une simple décision basée sur l'énergie, ces erreurs sont bien entendu inévitables.

Parole bruitée

Pour la parole légèrement bruitée telle que celle utilisée dans la sous-section précédente, le processus d'estimation semble non influencé. Afin d'examiner la performance dans le cas de la parole réelle perturbée par le bruit aléatoire, le signal de parole propre utilisé précédemment a été mélangé à un bruit blanc uniformément réparti et un bruit blanc gaussien. Ceci donne les SNRs suivants:

- uniforme: $SNR = 4.2dB(T_n = 100)$ et $SNR = -1.7dB(T_n = 200)$.
- gaussien: $SNR = 5.5dB(\sigma = 50)$ et $SNR = 0.0dB(\sigma = 95)$.

Puisqu'il est impossible, même à l'œil, de distinguer le bruit des segments de silence et les segments voisés ayant une faible énergie, les statistiques sur la décision ne sont pas données. Les contours de fréquences fondamentales pour les phrases TIR et LEC dans le cas SNR= -1.7dB sont montrés dans la Fig.3.2 (c) et la Fig.3.3 (c) respectivement. Presque toutes les valeur de F_0 estimées sont correctes, comme on peut le constater dans la Tableau 3.1. Encore une fois, ce résultat montre que la stratégie d'optimisation globale dans une fenêtre entière est efficace pour éliminer le bruit aléatoitre. Il n'est pas surprenant de voir que les silences et certains segments voisés de faible énergie sont annoncés comme non-voisés. De telles erreurs ne sont pas facilement remédiables sans utiliser des techniques de reconnaissances de formes qui utilisent l'information sur la nature de bruit de fond.

Multi-locuteurs

L'estimateur a été ensuite testé sur un plus grand corpus dans un contexte multi-locuteurs. Les résultats sont présentés dans les Tableaux 3.4 3.5 et 3.6.

Le corpus contient essentiellement

- 10 phrases françaises (differentes des celles utilisées dans les tests précédents) équilibrées phonétiquement (Tableau 3.4),
- des mots isolés (Tableau 3.5) et
- le corpus de parole du GRECO [Carre 84] (Tableau 3.6),

et est enregistré pour des locuteurs masculins et féminins, pour une durée d'environ 10 minutes. La largeur de la fenêtre d'analyse adoptée est de 450 points. L'étendue des valeurs de F_0 mesurées est de 65Hz à 400Hz. Dans les tableaux, F0m est la moyenne des périodes de pitch et la valeur entre parenthèses est la déviation standard de la période.

Locuteur	Taille	N_e	Temps(s)	F0m	$F0_{er}$ / $V-UV-S_{er}$ (%)
FRCH	1814	2321	23.22	150.51(18.19)	0.04/0.04
FRCH	1896	2426	24.27	148.61(18.10)	0.04/0.00
JMPI	1899	2433	24.31	91.91(14.95)	0.21/0.49
JPHA	2052	2626	26.27	100.05(16.81)	0.04/0.42
DIMA	1718	2200	21.99	122.90(17.89)	0.00/0.23
JPDA	1693	2168	21.67	130.18(20.46)	0.14/0.05
YIGO	5120	6554	65.54	117.69(25.17)	0.24/0.34
ANBO	2000	2560	25.60	223.81(50.87)	0.08/0.20
Global	18192	23288	232.86		0.12/0.24

Table 3.4: Résultats pour 7 autres locuteurs prononçant des phrases phonétiquement équilibrées.

Locuteur	Taille	N_e	Temps(s)	F0m	$F0_{er}$ / $V-UV-S_{er}$ (%)
BA	795	1018	10.18	218.79(20.68)	0.39/0.49
CM	1081	1384	13.84	246.96(63.4)	0.58/0.51
DC	732	937	9.37	136.1(21.31)	0.64/0.64
DF	669	856	8.56	171.29(35.21)	0.58/1.05
DMC	1103	1411	14.12	283.56(52.38)	0.21/0.28
FC	730	935	9.34	171.77(23.16)	0.32/0.21
FL	990	1268	12.67	114.01(25.71)	0.24/0.32
FLM	1905	2438	24.38	128.33(12.94)	0.49/0.25
FM	868	1111	11.11	127.07(31.58)	0.90/0.54
GM	855	1095	10.94	146.66(32.27)	0.46/0.55
JCJ	600	768	7.68	152.96(25.91)	0.65/0.26
JDM	584	747	7.48	119.73(12.14)	0.13/0.27
JLB	689	882	8.82	118.58(16.89)	0.57/0.34
JPD	333	426	4.26	176.6(21.73)	0.23/0.23
JPZ	1932	2473	24.73	132.03(23.87)	0.69/0.40
LMC	920	1177	11.78	231.33(43.26)	0.00/0.17
MB	789	1010	10.10	127.94(41.44)	1.19/0.40
OM	823	1053	10.53	225.61(41.28)	0.19/0.19
PH	800	1024	10.24	160.16(25.8)	0.49/0.29
PP	796	1019	10.19	144.61(26.68)	0.29/0.39
Global	17994	23032	230.32		0.48/0.38

Table 3.5: Résultats pour 20 locuteurs prononçant des mots isolés et des chiffres.

Locuteur	Taille	N_e	Temps(s)	F0m	$F0_{er}$ / $V-UV-S_{er}$ (%)
B.P	3157	4041	40.41	131.27(34.53)	0.45/0.12
B.G	3466	4436	44.36	110.22(24.92)	0.14/0.20
Global	6623	8477	84.77		0.28/0.17

Table 3.6: Résultats pour la base de données du GRECO constituant des fables françaises.

Influence de l'Increment K

Des différences de comportement entre K=0 et K=0.1 apparaissent lorsque les phrases de test contiennent des segments où la période de parole est visiblement dépendante du temps. La Fig.3.7 montre l'estimation de période pour le signal de parole du phonème /i/ dans le mot "ils" (TIR) avec K=0 (a) et K=0.1 (b). Il est visible à travers les impulsions synthétisées que les positions du pic maximum estimé dans chaque période ne sont plus significatives dans la cas K=0 même si la moyenee des périodes est correcte, alors que dans le cas K=0.1, comme nous l'avons espéré, chaque période p_i et chaque position estimée sont déterminées sans ambiguïté. Dans cette figure, F_0 estimée est 169.49 pour $K = 0$ et 172.41 pour K=0.1.

Indications des pics d'excitation

Par minimisation de l'erreur d'appariement pondérée entre un signal synthétisé et le signal de parole dans toute la fenêtre d'analyse, notre algorithme présente une meilleure résistance aux cas pathologiques tels ques les excitations secondaires. En effet, les pics du signal de la parole pouvant provoquer une valeur double dans la détection de F_0 sont différents des vrais pics qui indiquent la période sur l'amplitude ou la largeur de l'implusion. Ces différences réduit la qualité de l'appariement (la fonction R qui mesure la qualité est indépendante du nombre d'implusions synthétisées). Nous donnons deux exemples de l'estimation pour ce type de parole sur la Fig.3.8 où les lignes verticales positionnées par notre programme indiquent les évènements des excitations.

La localisation des pics est stable en présence du bruit. Nous présentons deux exemples sur la Fig.3.9. Les pics étroits positifs ou négatifs indiquent les positions localisées.

8 Conclusion

Nous avons présenté un modèle de parole variable en fonction du temps destiné à l'estimation de la fréquence fondamentale de la parole. A partir de ce modèle, nous avons conçu un detecteur de la fréquence fondamentale dans le domaine temporel qui se réfer à la stratégie de l'*EFFASH* (l'estimation de la fréquence fondamentale par l'appariement des structures harmoniques). Nous avons discuté certains aspects importants de son implantation. Cette nouvelle méthode effectue un appariement de structures harmoniques du domaine fréquentiel par l'optimisation double d'un critère d'énergie dans le domaine du temps. Certaines ambi-

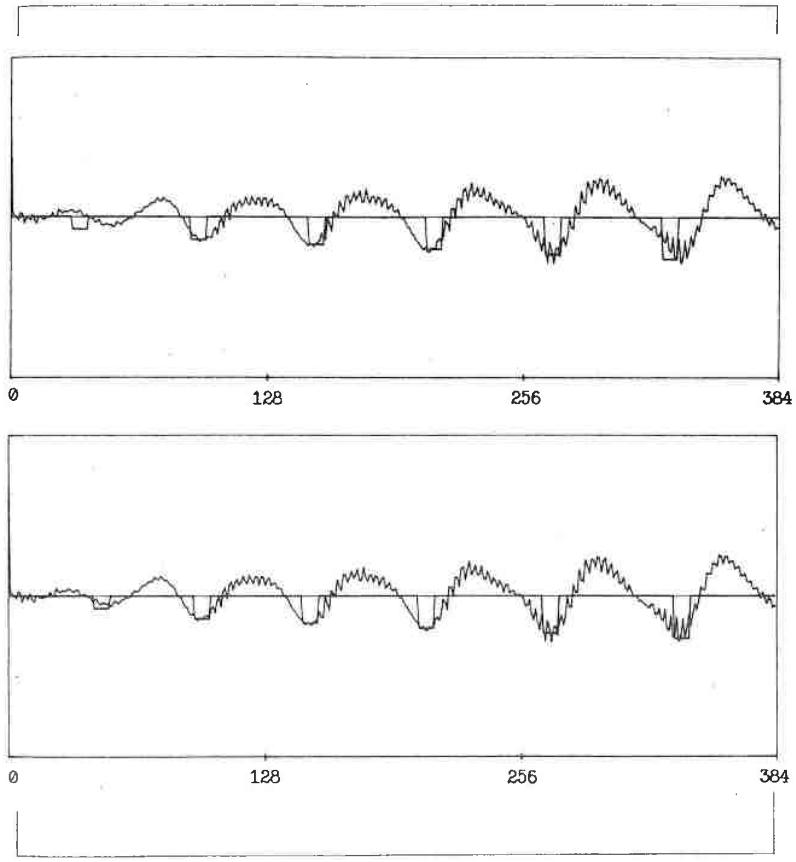


Figure 3.7: Estimation de périodes pour un signal de parole variable en fonction du temps (/i/ dans "ils" (/il/)) avec K=0 (en haut) et K=0.1 (en bas).

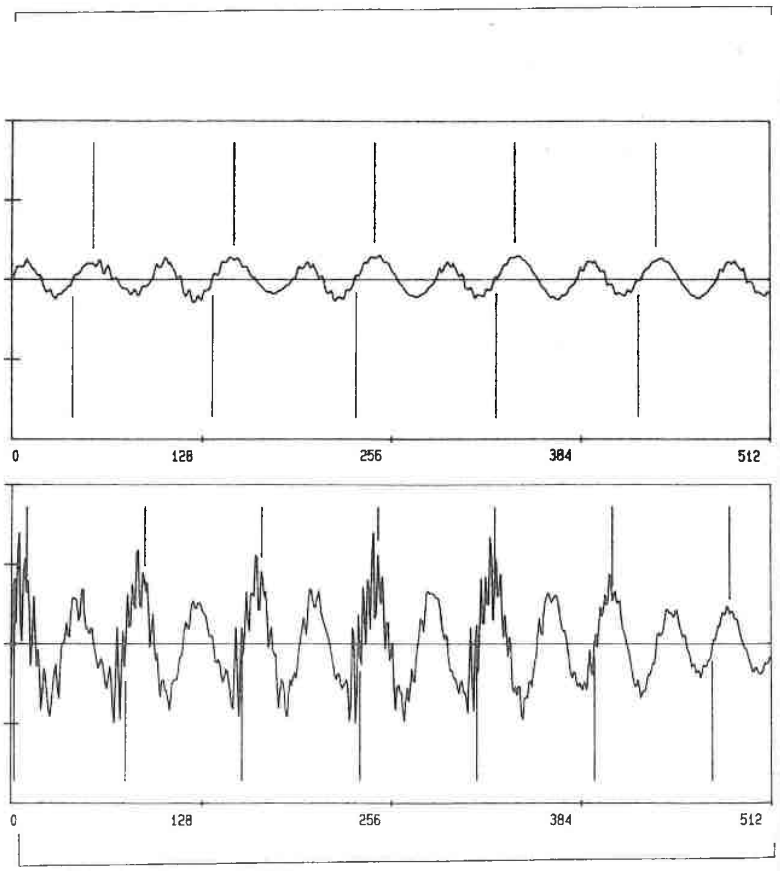


Figure 3.8: Deux exemple de localisation des évènements d'excitation du signal de la parole dont la détection est difficile

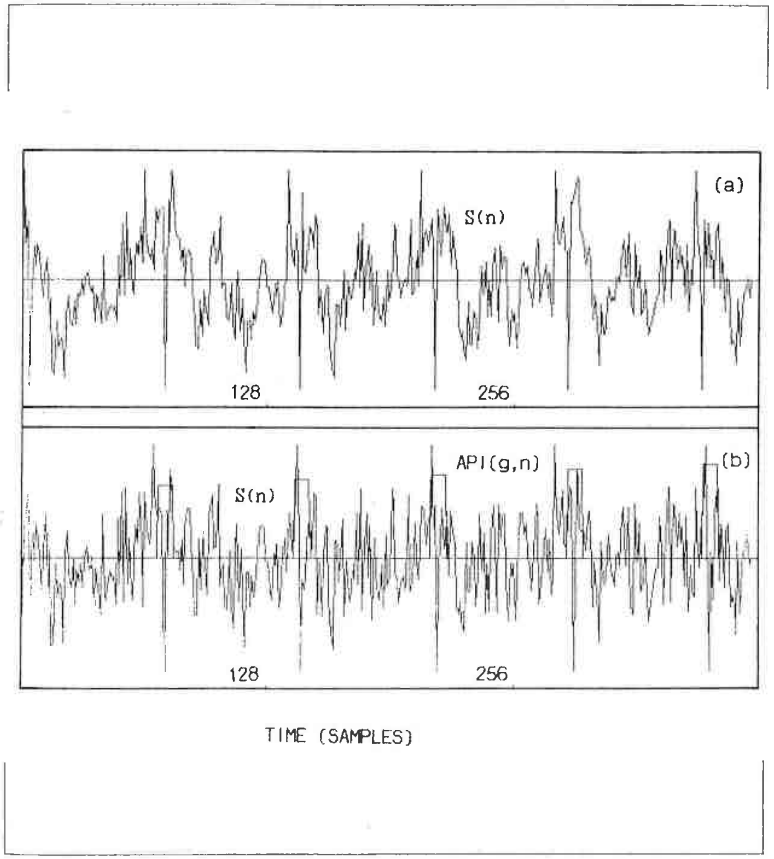


Figure 3.9: Estimation des position des impulsions d'excitation dans la parole bruitée pour le signal du phonème /o/ dans le contexte "chapeau" (/ʃapo/) de la phrase TIR avec (a) $T_n = 200$ et (b) $T_n = 500$. K=0.

guitiés dans la détection de la période pendant l'appariement sont éliminées en représentant le signal de parole par une séquence de fonctions pseudo-périodiques. La formulation dépendante du temps à la fois sur la période et l'amplitude du signal de parole et sur la réduction de la non-stationnairité de la séquence d'erreurs fournit des résultats fiables pendant les transitions des phonèmes. L'inconvénient du schéma du traitement par bloc est donc surmonté. L'estimation statistique assure une bonne détection de période dans le cas où le signal de parole est très bruité. Les paramètres pour la décision voisé/non-voisé dérivés de la méthode offrent une bonne indication pour une grande variété de son. Le traitement de signaux avec une onde asymétrique est amélioré par la sélection de polarité. Les périodes obtenues sont vérifiées par le calcul d'une fonction AMDF dépendante du temps. L'utilisation d'une séquence de fonctions fenêtre et l'opération complète dans le domaine du temps rend l'algorithme efficace en temps de calcul. Comme sous-produit, une approximation des positions et des amplitudes du point ayant la valeur maximale dans chaque période est aussi obtenue.

Les expériences sur 10 minutes de parole pour 30 locuteurs montrent que le modèle est satisfaisant dans différentes situations réelles. La linéarisation des variations de période et d'amplitude est valide et suffisante. Pour les applications normales, une valeur nulle de l'incrémentation maximum détectable des périodes successives dans une fenêtre donne des estimations satisfaisantes. Dans le cas où plus d'information sur chaque période dans une fenêtre est exigée avec précision, la valeur K=0.1 peut être utilisé. Les résultats obtenus montrent que la méthode est rapide, fiable et donne des valeurs de fréquence précises. Pour les signaux de parole propres, filtrés et bruités ces résultats sont meilleurs que ceux obtenus avec d'autres méthodes qui sont parfois plus couteuses en temps de calculs. Dans notre test, nous constatons que les statistiques ne varient plus en fonction du nombre de phrases traitées, nous concluons donc que les résultats sont fiables. L'approche proposée est donc prometteuse et peut être développée d'avantage pour les applications dans des différents domaines du traitement de la parole.

Le détecteur a été utilisé pour des travaux de reconnaissance de la parole [Gong 86a] et d'analyse-synthèse de parole à partir du signal enregistré dans la communication orale *pilote d'avion-tour de contrôle* qui est fortement bruité et sévèrement limité en bande passante. En ce qui concerne la recherche future sur notre algorithme, on peut envisager l'exploration des paramètres tels que la position et l'amplitude du maximum de chaque période, l'étude de l'influence de la largeur de la fonction de fenêtre T, l'effet de différents pré-traitements sur la performance et la comparaison avec différents algorithmes de détection de la fréquence fondamentale.

Partie II

CONVERSION DE SIGNAL-SYMBOLE SOUS INCERTITUDE

Chapitre 4

Classification Floue des Objets Réels

La conversion signal-symbole consiste en l'association d'un symbole à une portion de signal. A cause de l'incertitude, cette association est indéterministe. A un instant donné, une zone dans le signal peut vérifier partiellement des propriétés de plusieurs symboles. Autrement dit le signal appartient à plusieurs classes de symboles. Nous modélisons ce phénomène par une classification floue où un signal est associé simultanément à plusieurs classes. Il est généralement reconnu que l'accès à un dictionnaire ayant comme entrée un signal et comme sortie un ou plusieurs symboles demande en général un nombre de comparaisons proportionnel à la taille du vocabulaire M, car il est difficile d'ordonner des signaux, représentés par des vecteurs dans un espace infini, dans le but de les structurer. Des techniques ont été proposées pour réduire ce nombre jusqu'à $\log_2 M$ mais la probabilité d'associer un signal à son plus proche voisin dans le dictionnaire est faible. Nous proposons un algorithme de classification floue pour l'accès rapide qui permet de réduire ce nombre propotionnel à $\log_2 M$ et de faire un compromis entre le temps et la qualité de recherche. Par des tests avec des vecteurs aléatoires, nous obtenons dans 96% de cas le plus proche voisin.

1 Introduction

L'homme est incapable de raisonner sous forme d'information numérique car cette forme ne permet pas d'exprimer des notions, des relations et des structures. Le raisonnement nécessite l'identification des situations, – de catégoriser le signal observé en un ensemble d'expressions symboliques. Cette identification constitue la première étape dans le processus d'interprétation du signal.

Pour que le signal transmette de l'information il faut identifier, à partir de sa variation continue, un ensemble fini d'élements discrets. La conversion signal-symbole est un processus d'association des points de l'ensemble infini de vecteurs V (domaine numérique) aux points de l'ensemble fini de symboles S (domaine symbolique). Cette association oblige une quantification de l'espace de représentation ce qui entraîne une perte d'information.

Le problème de la conversion signal-symbole consiste à trouver une fonction

$$f : V \rightarrow S$$

qui est

- complète: pour tout vecteur correspondant au symbole s f donne s et
- discriminante: pour tout vecteur ne correspondant pas au symbole s, f ne donne pas s.

On peut distinguer deux types de conversion signal-symbole:

- Dans le premier type, les symboles convertis n'ont pas un sens physique, à part le fait que chacun d'entre eux représente une repartition des signaux. A priori on ne connaît pas le nombre de classes ni les symboles. Le seul type d'information utilisé pour réaliser la conversion est la répartition statistique d'un ensemble de signaux échantillonnés. Ce type de classification est également appelée classification non supervisée ou clustering. La quantification vectorielle en est un exemple: chaque code (un symbole) ne porte que l'information sur la répartition dans l'espace de représentation d'un signal. Dans ce chapitre nous decrivons une méthode de conversion de ce type.
- Dans le deuxième type de conversion, chaque symbole représente un sens conceptuel dans un langage de communication. Ce type de classification est aussi connu sous le nom de classification supervisée. Dans ce cas l'information sur la répartition statistique des échantillons n'est plus suffisante pour la conversion. D'autres types de connaissances sur la relation signal-symbole sont nécessaires. Nous décrivons en détail ce type de conversion dans le chapitre 5.

Nous montrons dans ce chapitre d'abord la nécessité d'introduire des classes floues pour la représentation de l'information. Nous présentons ensuite une méthode de classification floue et binaire. Nous appliquons enfin cette méthode à la recherche rapide du plus proche voisin d'un vecteur de test dans l'espace R^n.

2 Clustering

Le clustering, ou classification non supervisée, est une technique de conversion signal-symbole spéciale au sens que les symboles associés aux signaux de test correspondent à la répartition statistique des signaux et n'ont pas d'autres significations physiques.

Il existe plusieurs définitions d'un cluster dont nous proposons celle-ci [Everitt 74]:

> Les clusters sont des regions connectées dans un espace à N-dimensions contenant une densité de points relativement importante.

Le clustering par coalescence, est une technique d'analyse de données qui organise des formes en groupes ou clusters de telle façon que les formes dans un groupe sont plus semblables entre

elles que les formes de groupes différents, selon un critère spécifique [Chen 87]. La technique de clustering explore des données sans utiliser la connaissance à priori sur la répartition de ces données.

Beaucoups d'algorithmes de clustering ont été proposés tels que K-means, Isodata, et des méthodes de clustering hiérarchique. Les livres de Anderberg [Anderberg 73], Duda et Hart [Duda 73], [Everitt 74] et les les articles de synthèse [Chien 76,Dubes 81] donnent un paranoma du domaine.

Pour classifier un objet réel on utilise un certain ensemble de propriétés mesurées sur l'objet. En réalité, un objet peut posséder quelques propriétés qu'un autre objet possède également. Cet objet peut avoir donc un degré d'appartenance à une autre classe d'objets. Par conséquent, la classification d'un objet réel a intrinsèquement la propriété de flou.

3 Une Méthode de classification floue

3.1 Classification floue

On considère le centre C_1 d'une classe de points et un point p. S'il existe aucune autre classe on peut associer p à la classe centrée en C_1 sans aucun doute. Supposons maintenant que l'on place un autre nuage de points centrés à C_2 de façon que la distance du point p en C_2 soit inférieure à la distance de p à C_1. Si on considère la classification de p, on va remettre en cause la classification faite sans la présence de C_2 et associer p à la classe centrée en C_2 et non plus en C_1. En fait, un point dans l'espace de représentation *appartient à toutes les classes* avec un degré d'appartenance dans chacune. La distance du point aux centres des classes peut par exemple indiquer le degré d'appartenance. La classification du point possède donc intrinsèquement le caractère flou.

La classification floue est un processus où les classes obtenues à chaque niveau de classification ne s'excluent pas mutuellement. Autrement dit, un élément de l'ensemble d'échantillons peut appartenir simultanément à plusieurs classes comme le montre la figure 4.1.

Cette idée, formulée mathématiquement par Zadeh [Zadeh 77,Zadeh 78], sous la théorie des ensembles flous, montre la différence fondamentale entre la classification en classes nettes et la classification floue [Selim 84]. La notion de flou est introduite dans le clustering [Ruspini 70,Backer 78,Bezdek 81,Dunn 73,Ruspini 73] et est exploitée de plus en plus en reconnaissance des formes [Zadeh 77,Ruspini 73,Bezdek 81,Pramanik 86,Bezdeck 86] et plus récemment en segmentation d'image [Trivedi 86] et reconnaissaance de parole [Tseng 87].

L'objectif de notre travail sur la classification floue est de constuire un arbre flou à partir d'un ensemble de formes initiales. Cet arbre permet d'associer ultérieurement une forme de test à un symbole dont la forme est la plus proche possible de celle de test avec un temps de recherche court et une précision suffisante.

3.2 Algorithmes

Nous définissons d'abord les objets manipulés par les algorithmes. Nous adoptons la même notation que celle utilisée par le langage C [Kernighan 78].

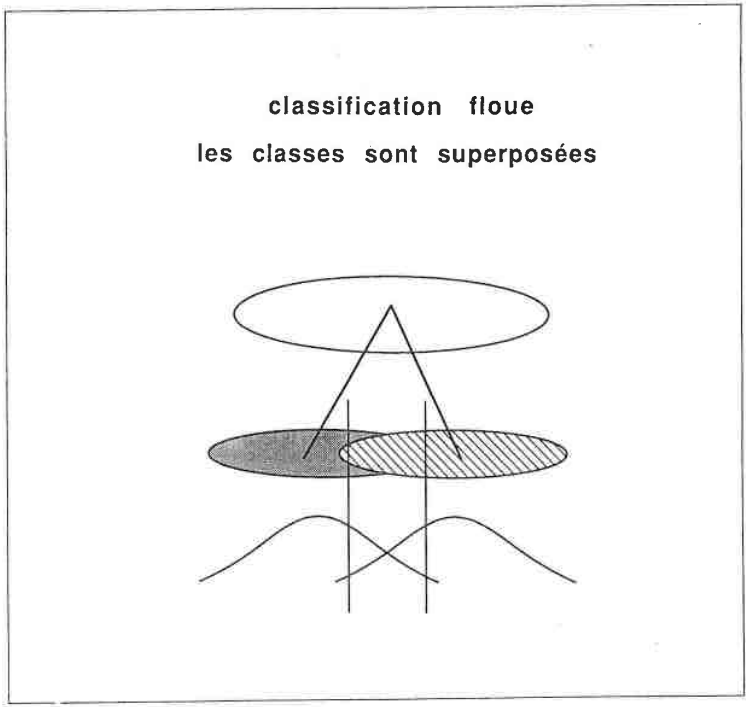


Figure 4.1: Chaque point dans l'espace appartient à toutes les classes avec un degré d'appartenance différent

- La structure d'une classe de points:

```
typedef struct clust {
        char *nom;
        float *centre;
        int taille;
        struct clust *clDroite, *clGauche;
        struct clust *suivant;
} clusttp;
```

- La structure d'un arbre flou:

```
typedef struct fuzzytree {
        char *nom;
        float *centre;
        clusttp *membre;
        struct fuzzytree *TrDroite, *TrGauche;
} fttp;
```

- La structure d'une paire de classes:

```
typedef struct {
        float dist;
        clusttp *clGauche, *clDroite;
} pairtp;
```

La construction de l'arbre de classification s'effectue en divisant récursivement l'ensemble d'échantillons initial en deux classes qui ne soient pas en exclusion mutuelle. La division se termine lorsque l'une des deux classes est vide. A chaque phase, la division en deux classes réalisée par la fonction "Séparer2Classes" est effectuée de la façon suivante:

- On classe l'ensemble de points en deux classes exclusives par un algorithme classique [Jain 86,Jambu 78,Tou 74], tel que le K-Means [Diday 76], Linde-Buzo [Linde 80] ou la classification hiérarchique [Duda 73]: $E_i = E_{i,1} \bigcup E_{i,2}$, tq $E_{i,1} \bigcup E_{i,2} = \emptyset$;

- On ajoute dans chacune des classes résultantes les points de l'autre classe dont la différence des degrés d'appartence aux deux classes est faible:
  $C_{i,1} = E_{i,1} + \{e \in E_{i,2},\ \text{tq}\ \mu(E_{i,1}, e) - \mu(E_{i,2}, e) < Seuil\}$
  $C_{i,2} = E_{i,2} + \{e \in E_{i,1},\ \text{tq}\ \mu(E_{i,2}, e) - \mu(E_{i,1}, e) < Seuil\}$

```
liste = la liste des points à classer en classe floue : clusttp
diam = la distance relative au plan de séparation des classes : [0,0.5)
centre = le centre de gravité de l'ensemble des points dans liste: vecteur
nœud : fttp
filss : pairtp
racine : clusttp
```

```
classe-floue(liste, diam, centre) → fttp
    Si vide(liste)
    Alors Null
    Sinon
        racine = Séparer2Classes(liste);
        nœud = CréerNœud(fttp);
        TrGauche(nœud) = TrGauche(nœud) = Null;
        centre(nœud) = centre;
        membre(nœud) = liste;
        Si (clGauche(racine)) et (clDroite(racine))
        Alors
            filss = Diviser(liste,diam,
                    centre(clDroite(racine)),centre(clGauche(racine)));
            TrGauche(nœud) =
                classe-floue(clGauche(filss),diam,centre(clDroite(racine)));
            TrDroite(nœud) =
                classe-floue(clDroite(filss),diam,centre(clGauche(racine)));
        Sinon
            nœud;
        Finsi
    Finsi
```

La fonction "Diviser" divise un ensemble de points en deux classes non exclusives en prenant comme paramètres deux centres de nuages de points obtenus par "Séparer2Classes". La fonction retourne le couple de classes.

```
lpt = liste de points à classer : clusttp
diam = la distance relative au plan de la séparation des classes : [0,0.5)
ctr1, ctr2: les deux centres de nuage de points.
p, p1 : pairtp
```

```
Diviser(lpt, diam, ctr1, ctr2) → pairtp
    p = clean-separate(lpt,ctr1,ctr2,Null,Null);
    dref = diam × fdist(oc1,oc2);
    p1 = CréerNœud(pairtp);
    clGauche(p1) = add-fuzzy(clGauche(p),clDroite(p),dref,oc1,oc2);
    clDroite(p2) = add-fuzzy(clDroite(p),clGauche(p),dref,oc1,oc2);
```

La fonction "clean-separate" utilise des mesures classiques pour séparer une liste en deux classes distinctes, les nouveaux centres étant donnés.


```
lpt = la liste de points à classer : clusttp
oc1, oc2 = les centres des nuages
ll, lr, c1, c2 : clusttp
p : pairtp
```

```
clean-separate(lpt,oc1,oc2,ll,lr) → pairtp
    Si Non(lpt)
    Alors
        p = Crée(pairtp); clGauche(p) = ll; clDroite(p) = lr; p;
    Sinon
        Si (dist-ratio(centre(lpt), oc1, oc2) < 1.0)
            Alors
                c1 = Crée(clusttp); c1 = lpt; suivant(c1) = ll;
                clean-separate(suivant(lpt),oc1,oc2,c1,lr);
            Sinon
                c2 = Crée(clusttp); c2 = lpt; suivant(c2) = lr;
                clean-separate(suivant(lpt),oc1,oc2,ll,c2);
        Finsi;
    Finsi
```

Nous ajoutons dans les deux classes ainsi créées les points de l'autre classe selon un critère sur la distance relative $D(p_1, p_2, x)$, détaillé dans 3.3, qu'un point possède à l'écart du plan de séparation des deux classes. La fonction "add-fuzzy" réalise cette opération.

```
augl = la liste à laquelle on ajoute : clusttp
decl = la liste qui fournit des points de test : clusttp
dref = la distance au plan de séparation des classes : réel
oc1, oc2 = les deux centres de nuages
cls :clusttp
```

```
add-fuzzy(augl,decl,dref,oc1,oc2) → clusttp
    Si Non(decl)
    Alors augl
    Sinon
        Si D(oc1,oc2,centre(decl)) < dref
        Alors
            cls = Crée(clusttp):
            cls = decl
            suivant(cls) = add-fuzzy(augl,suivant(decl),dref,oc1,oc2);
            cls;
        Sinon
            add-fuzzy(augl,suivant(decl),dref,oc1,oc2);
        Finsi;
    Finsi;
```

3.3 Plan de séparation

Nous décrivons maintenant le plan de séparation des deux classes.

Soient p_1 et p_2 deux vecteurs dans R^n représentant deux centres de nuages de points. On définit la distance euclidienne entre deux points x, y dans l'espace R^n comme

$$d(x,y) = \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2}$$

Tous les points x se situant à égale distance de p_1 et de p_2 forment un hyperplan T, correspondant au plan de séparation des deux classes de points centrés respectivement en p_1 et en p_2.

$$T : \forall x \in R^n \;\; d(p_1,x) = d(p_2,x).$$

Soit x un point dans l'espace, la distance $D(p_1,p_2,x)$ de x au plan T est donnée par la formule suivante:

$$D(p_1,p_2,x) = \frac{|\sum_{i=0}^{N-1}(p_{2,i} - p_{1,i})x_i - h|}{\sqrt{\sum_{i=0}^{N-1}(p_{2,i} - p_{1,i})^2}}$$

où

$$h = \frac{1}{2}(\sum_{i=0}^{N-1} x_{2,i}^2 - \sum_{i=0}^{N-1} x_{1,i}^2)$$

$D(p_1,p_2,x)$ est utilisé pour prendre la décision lors de la classification floue. La figure 4.2 montre deux nuages de points et le plan de séparation associé.

D'autres distances vérifiant la symétrie et l'inégalité triangulaire peuvent être également utilisées pour définir le plan de séparation, à condition qu'on peut obtenir une formule analytique de D.

4 Accès au dictionnaire signal-symbole

Soitent Ω_1 et Ω_2 deux ensembles bornés. Un dictionnaire est un ensemble de couples $< p_i \in \Omega_1, q_i \in \Omega_2 >$ dont chacun est accessible par l'extérieur à travers le mot-clé p_i, réalisant une application de Ω_1 à Ω_2. Le dictionnaire est utilisé comme une table de correspondance. L'ensemble Ω_1 dont les éléments dans les couples permettent de retrouver un couple est appellé *l'entrée* du dictionnaire. Nous distinguons deux types de dictionnaire:

- le dictionnaire à entrée symbole et
- le dictionnaire à entrée signal.

4.1 Dictionnaire symbole

Dans un dictionnaire à entrée symbole, la définition de correspondance est exacte. Le résultat d'un accès d'un couple est obtenu par des tests d'égalité et il n'y a que deux cas possibles:

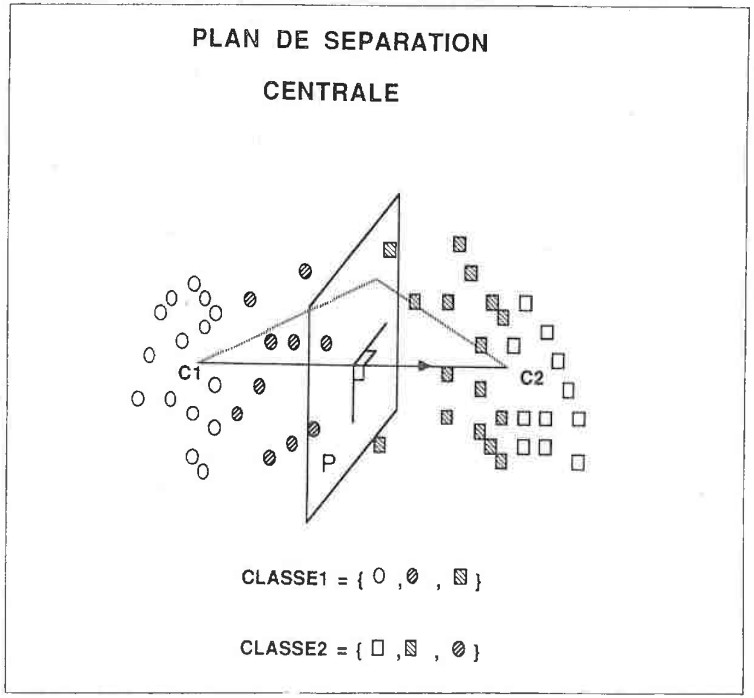


Figure 4.2: Deux classes exclusives et le plan de séparation des classes associées

1. si un couple contient le symbole d'entrée alors le couple est retouné comme résultat,

2. si le symbole d'entrée n'est pas inclu dans le dictionnaire alors erreur est rendue.

Grâce à l'exactitude de la correspondance, on peut définir un ordre sur les symboles dans l'ensemble d'entrée Ω_1 et organiser le dictionnaire sous forme d'un arbre K-aire. On peut ainsi réduire la complexité de l'algorithme d'accès au dictionnaire jusqu'à $log_K M$ où M est le nombre de couples contenus dans le dictionnaire.

4.2 Dictionnaire signal

Un dictionnaire à entrée signal est une table de conversion signal-symbole. Cette table décrit la correspondance entre les centres des nuages dans le domaine du signal et les symboles associés. En fait, chaque symbole dans le dictionnaire est associé à un ou plusieurs vecteurs de référence. Un tel dictionnaire est inexact parce que le nombre de signaux, ou le nombre de points dans l'espace de représentation, est illimité alors que Ω_1 est borné. Quelque soit la taille du dictionnaire, la probabilité de trouver exactement un signal de test dans le dictionnaire est toujours égale à zéro. C'est pourquoi la notion du plus proche voisin est introduite pour effectuer l'accès au dictionnaire.

Soit

$$\Omega_1 = \phi = \{v_1, v_2, ..., v_M\}$$

l'entrée du dictionnaire où chaque v_i est un vecteur. Le plus proche voisin du vecteur x dont on cherche le symbole associé est

$$v_m \in \phi \text{ tel que } d(x, v_m) = \min_{v_i \in \phi} d(x, v_i)$$

où $d(x, y)$ mesure la distance métrique entre les points x et y.

L'inexactitude d'un dictionnaire signal n'autorise pas de définir un ordre sur les éléments dans ϕ permettant de trouver le plus proche voisin d'un point inconnu par l'exploitation d'un arbre K-aire. Une méthode directe est la comparaison exhaustive. Lors de la conversion, on donne un vecteur de test et une distance métrique. La recherche du plus proche voisin du vecteur peut se faire par comparaison successive du vecteur à tous les vecteurs dans ϕ. Le plus proche voisin est le vecteur dans ϕ qui réalise la distance minimale au vecteur de test.

Lorsque la taille du dictionnaire est importante, telle que dans le codage de la parole par la quantification vectorielle [Buzo 80,Miclet 84] où le nombre de références peut aller de quelques centaines jusqu'à quelques milliers, la comparaison exhaustive d'un vecteur de test aux éléments dans Ω_1 devient prohibitive.

4.3 Méthodes d'accès rapide existantes

Trois approches ont été proposées pour améliorer le temps d'accès dans un dictionnaire signal-symbole:

- On construit dynamiquement (au moment de la recherche) un graphe de dépendance sur la relation entre le vecteur de test et les vecteurs comparés pour guider la recherche

en évitant les comparaisons inutiles [Jambu 78]. La méthode nécessite des calculs supplémentaires pour prendre des décisions et l'efficacité de l'algorithme décroit avec la dimension de l'espace de représentation.

- On hiérarchise le dictionnaire en un arbre K-aire selon les vecteurs. Lors de la conversion l'arborescence est utilisée pour la recherche du plus proche voisin. Pour un arbre K-aire équilibré, le nombre d'accès est de $O(\log_K M)$. La structure hiérarchique est créée de façon à minimiser l'erreur des décisions au moment de la conversion [Gersho 82,Gray 84,Miclet 83,Linde 80]. Au moment de la hiérarchisation du dictionnaire, l'information sur la répartition de chaque nuage est utilisée. Les méthodes issues de cette approche ne garantissent pas le plus proche voisin. En effet, le gain de temps est obtenu en sacrifiant la précision. En général, il est difficile d'évaluer l'amélioration de vitesse d'accès et la distorsion introduite de ces algorithmes autrement qu'expérimentalement [Miclet 84]. Par exemple, le travail de Rochette et ses collègues sur l'accès rapide d'un dictionnaire de 1024 mots est de structurer le dictionnaire en un arbre binaire équilibré basé sur des hyperplans et des centroïdes. Selon les auteurs [Rochette 87], leur algorithme requiert 77 calculs de distances pour obtenir un représentant qui est à plus de 70% le plus proche voisin du vecteur de test et à plus de 90% l'un des 3 plus proches.

- Pour trouver exactement le plus proche voisin dans un dictionnaire hiérarchique, certains chercheurs effectuent la recherche avec retour-arrière. A chaque pas de la recherche, on teste la validation de quelques règles définies sur la distance au plus proche voisin déjà rencontré et sur des distances locales [Fukunaca 75,Feustel 82,KamgarParsi 85, Lockwood 85]. Dans ce cas le temps d'accès maximum nécessaire est beaucoup plus que l'ordre logarithmique à cause des calculs suplémentaires. Lockwood a testé plusieurs ensembles de règles dans le cadre de la reconnaissance des mots isolés par programmation dynamique. Selon ses résultats, le rapport entre le nombre de comparaisons effectuées par la recherche hiérarchique et le nombre de comparaisons nécessaires par la recherche successive est de 37% à 64%. Plus ce rapport est réduit, plus l'erreur de reconnaissance est importante[Lockwood 86].

4.4 Accès rapide par classes floues

Le défaut de la deuxième approche est indiqué dans la figure 4.3. Cette figure schématise une étape de classification d'un point de test. On voit que, selon le critère de distance, la classification obtenue est fausse. Une fois la sous-branche de l'arbre mal choisie, il devient impossible de trouver le plus proche voisin dans le dictionnaire.

Notre méthode constitue une autre approche. Nous utilisons la classification floue pour organiser le dictionnaire sous une hiérarchie floue et pour diminuer la possibilité d'un telle erreur. Lors de la construction de classes hiérarchiques, les points dont la différence de degrés d'appartenance aux classes est petite sont inscrits simultanément dans les deux classes descendantes. Cette hiérarchie permet de réduire de façon progressive l'incertitude des décisions prises au cours de la conversion signal-symbole d'un point.

Grâce à l'arbre binaire de classification floue, la fonction de recherche du plus proche

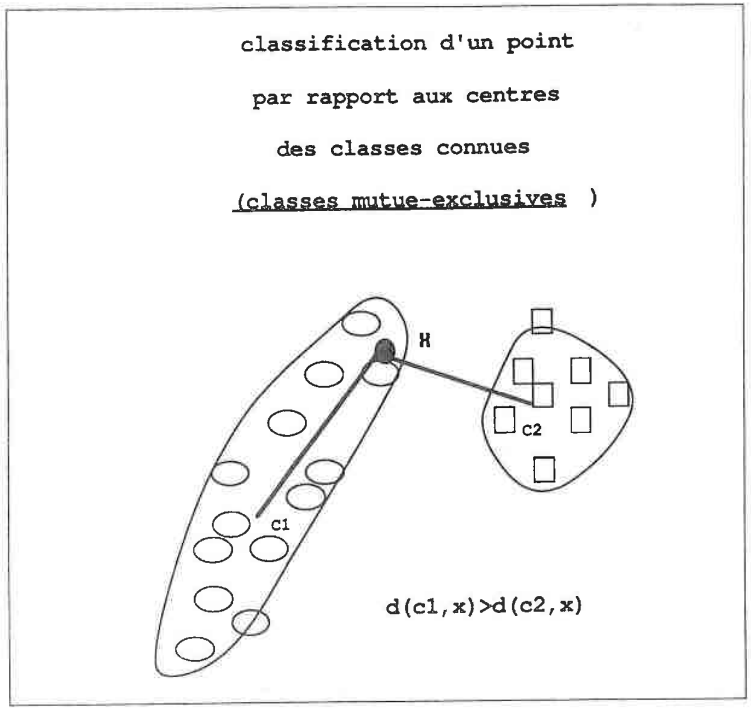


Figure 4.3: Exemple de classes s'excluant mutuellement. Selon le critère de distance, le point de test ne pourra jamais retrouver son plus proche voisin

voisin d'un point donné est simple, comme l'algorithme suivant le montre.

```
dico = l'arbre binaire de la classification floue : clusttp
vec = le point dont on cherche le plus proche voisin

AccèsFlou(dico,vec) → vecteur
    Si non(clDroite(dico) et clDroite(dico))
    Alors dico
    Sinon
        Si
            Dist(centre(clDroite(dico)),vec) <
            Dist(centre(clGauche(dico)),vec)
        Alors AccèsFlou(clDroite(dico),vec)
        Sinon AccèsFlou(clGauche(dico),vec)
        Finsi
    Finsi
```

4.5 Stockages supplémentaires

A chaque noeud du dictionnaire est stocké le centre d'un nuage de points. Sous les conditions

1. l'arbre est équilibre et
2. les classes ne sont pas superposées,

un dictionnaire de taille M nécessite $2 \times M - 1$ stockages des centres. En réalité ces conditions ne sont pas vérifiées et le nombre de stockages est supérieur à $2 \times M$.

5 Expérimentation

Nous avons testé notre méthode de classification floue en utilisant des données artificielles à base de nombres aléatoires. Les vecteurs d'apprentissage et de test sont des vecteurs aléatoires, générés à partir d'un générateur des nombres aléatoires. La génération consiste à remplir les composantes des vecteurs par des nombres aléatoires successivement calculés par la formule de récurence suivante:

$$N_n = N_{min} + \frac{r_n \times (N_{max} - N_{min})}{M}$$

où

$$r_n = (A \times r_{n-1} + C) \ modulo \ M$$

et $A = 7 \times 13 + 1$, $C = 2731$ (un nombre premier) et $M = 7^4 \times 13$ sont des constants, $r_0 = 1237$.

Les tests consistent à comparer le résultat de l'accès flou avec l'accès linéaire classique (par comparaison exhaustive) et de calculer le pourcentage des points trouvés qui sont exactement des voisins les plus proches. Les résultats obtenus dans la simulation sont présentés par les figures 4.4 et 4.5. Dans ces figures,

- ppv_t désigne le plus proche voisin *trouvé*,
- d, le diamètre qui mesure le degré de flou, est la distance relative,
- *temps*est le temps de construction de l'arbre flou plus celui de la reconnaissance,
- *niveau* est la profondeur de l'arbre donc le nombre maximum de comparaison pour convertir un vecteur en symbole par la recherche de son ppv_t,
- *type* représente l'algorithme utilisé pour classer les points en deux classes exclusives, K signifie K-means et H signifie hiérarchique,
- M_σ est la divergence entre le ppv_t et le vrai plus proche voisin ppv_r moyennée sur l'ensemble de points de tests, définie comme:

$$M_\sigma = \frac{1}{card(T)} \sum_{v \in T} \frac{dist(v, ppv_t) - dist(v, ppv_r)}{dist(v, ppv_r)} \times \%$$

où ppv_t et ppv_r sont respectivement le plus proche voisin trouvé et celui en réalité du vecteur v.

d	temps	niveau	ppv_t(%)	M_σ	type
0.000	2	8	149/200 (74.5)	11.72	K
0.100	2	9	152/200 (76.0)	11.22	K
0.200	2	8	160/200 (80.0)	7.34	K
0.300	2	10	165/200 (82.5)	4.88	K
0.400	3	11	174/200 (87.0)	2.60	K
0.499	5	15	180/200 (90.0)	2.99	K
0.000	3	10	130/200 (65.0)	25.74	H
0.100	3	11	145/200 (72.5)	11.79	H
0.200	3	12	156/200 (78.0)	8.22	H
0.300	4	13	173/200 (86.5)	3.87	H
0.400	4	16	180/200 (90.0)	1.68	H
0.499	8	18	192/200 (96.0)	0.41	H

Figure 4.4: résultat de la recherche rapide à base de classification floue en fonction de la distance relative $0 \leq d < 0.5$, le nombre de points à l'apprentissage est 25

Dans la figure 4.4 nous montrons la relation entre la distance relative et le pourcentage de plus proches voisins trouvés ppv_t sur l'ensemble des points de test. Nous constatons que

- Sans classification floue (d=0) le pourcentage ppv_t n'est qu'environ 65% à 74% suivant la méthode de classification exclusive utilisée. En introduisant certain degré de flou des classes ce pourcentage peut être amélioré jusqu'à 96% (d=0.499),

- Le ppv_t augmente lorsqu'on augmente la distance relative autorisée, de 74.5% à 90% pour le type K et de 65% à 96% pour le type H,
- La divergence diminue rapidement quand on augmente d, pour une variation de d de 0 à 0.499, M_σ de 11.72% à 2.99% pour le type K et de 25.74% à 0.41% pour le type H.

La figure 4.5 montre que le pourcentage des ppv_t obtenus est stable en fonction du cardinal de l'ensemble de test. Nous constatons que le ppv_t pour le type H est stabilisé à environ 96%.

nombre	temps	niveau	ppv_t(%)	M_σ	type
100	2	15	92/100 (92.0)	2.88	K
150	4	15	134/150 (89.3)	3.41	K
200	5	15	180/200 (90.0)	2.99	K
250	7	15	222/250 (88.8)	2.68	K
300	11	15	265/300 (88.3)	2.65	K
400	11	15	350/400 (87.5)	2.73	K
100	4	18	96/100 (96.0)	0.56	H
150	6	18	144/150 (96.0)	0.51	H
200	9	18	192/200 (96.0)	0.41	H
250	11	18	237/250 (94.8)	0.56	H
300	14	18	286/300 (95.3)	0.49	H
400	32	18	379/400 (94.7)	0.48	H

Figure 4.5: statistiques des résultats de classification floue en fonction du nombre de points de test, la distance d est maintenue à 0.499 et le le nombre de points à l'apprentissage est 25

Dans tous les tests le type H a donné de meilleurs résultats.

6 Application en conversion signal-symbole

Le principe de notre méthode est que, en présence d'incertitude, les décisions prises à différentes étapes de reconnaissance ne doivent pas s'exclure mutuellement afin de diminuer le risque de faire un mauvais choix. En introduisant un certain degré de flou, on diminue cette possibilité.

6.1 Quantification vectorielle

L'application la plus directe est la la recherche du plus proche voisin d'un vecteur dans un dictionnaire de vecteurs en quantification vectorielle. Dans cette application, la taille du dictionnaire est de l'ordre de quelques centaines à quelques milliers. L'accès rapide est donc souhaitable [Linde 80,Miclet 83,Gray 84].

6.2 Traits acoustiques en phonèmes

Il est généralement reconnu qu'un grand vocabulaire en reconnaissance de la parole doit être organisé en niveaux multiples [Adda 87]. Cette organisation doit permettre l'accès d'un mot en utilisant progressivement des traits acoustiques grossiers. Par exemple, un vocabulaire de 20000 mots peut être représenté par six critères de catégories phonétiques grossières en partitionnant hiérachiquement le lexique en classes de mots ayant les mêmes catégories phonétiques [Shipman 82]. Le danger au moment de la classification est que une faute de classification à un niveau donné ne pourra jamais être récupérée dans les niveaux ultérieurs. La méthode que nous présentons dans ce chapitre utilise la classification floue et donc peut être exploitée pour diminuer ce danger.

7 Conclusion

Nous avons présenté un algorithme de construction des classes floues qui permet d'effectuer un compromis entre le temps et la précision de l'accès d'un dictionnaire signal-symbole.

L'idée esentielle de la méthode est, lors de l'apprentissage, d'autoriser certains points dans l'espace de représentation d'appartenir simultanément à plusieurs classes afin d'éviter les erreurs irrécupérables au moment de la classification hiérarchique. Une fois le dictionnaire construit, la recherche du plus proche voisin d'un vecteur de test consiste à effectuer simplement un parcours d'un arbre binaire, sans nécessiter la construction de structure intermédiaire. Le temps d'accès est logarithmique par rapport à la taille du dictionnaire. Il est possible de trouver un compromis entre la précision et le temps d'accès en faisant variant une distance relative.

Par simulation, nous avons obtenu dans 96% de cas le plus proche voisin. En application réelle, les points sont répartis en nuages plus naturels, une précision plus élevée peut donc être espérée.

La performance de l'algorithme est encore améliorable sur les points suivants:

- recherche d'un algorithme adapté de séparation en deux classes nettes;
- recherche d'un autre critère de génération des classes floues.

La technique de classification floue présentée dans ce chapitre est une technique de reconnaissance statistique des formes et peut être utilisée directement dans d'autre applications.

Chapitre 5

Correction de la Déformation Contextuelle

La déformation des formes introduite par l'influence du contexte est difficile à modéliser par un modèle explicite et ne peut pas être résolue par des méthodes statistiques locales. L'identification de la présence d'un symbole dans le signal dépend des symboles voisins et d'un ensemble de descriptions en termes de mesures physiques sur le signal. Cette dépendance peut en général être décrite par des observations sous forme de connaissances floues et imprécises. Nous utilisons l'approche système à base de connaissances pour interpréter les résultats de la préclassification de symboles. Notre système accepte des prédicats flous paramétrables pour représenter le sens des descriptions linguistiques dans les règles. L'incertitude des données et des déductions est modélisée par des coefficients de certitude et par un raisonnement inexact. Ce système est utilisé pour interpréter les courbes des tons du chinois parlé, - un problème typique de reconnaissance de formes perturbées par le contexte. Le système fournit un outil d'aide à l'acquisition de connaissances sur la déformation contextuelle.

1 Introduction

1.1 Limites des méthodes locales

Nous avons défini deux types de classifications dans le chapitre précédent. On peut réaliser des classifications du premier type – la classification non-supervisée – de façon satisfaisante par des méthodes statistiques, car après tout ce type de classification sert à découvrir la répartition statistique de points dans un espace de représentation. Par contre, la classification du deuxième type – la classification supervisée –, qui consiste à associer un concept à un vecteur, est beaucoup plus difficile. Le comportement du système est déterminé d'une part par une fonction d'objectif [Duda 73,Foglein 86] et d'autre part par l'espace de représentation des objets et la mesure de distance entre les objets. Les fonctions d'objectif utilisées sont en général loin d'être suffisante pour modéliser convenablement les connaissances a priori sur la conversion. En ce qui concerne l'espace de représentation, on a constaté que quelque soient les paramètres choisis et la distance utilisée, les erreurs provoquées par l'influence

contextuelle ne peuvent pas être supprimées car la décision est prise de façon entièrement locale et l'information contextuelle n'a pas participé à la décision.

1.2 Travaux antérieurs

L'importance de l'utilisation de l'information contextuelle dans la conversion signal-symbole est reconnue depuis longtemps [Chen 73]. La manière la plus claire et la plus agréable de tenir compte du contexte est evidemment de définir une grammaire. La reconnaissance de formes par utilisation de la syntaxe tient compte de la relation entre les événements dans les formes. Une forme est decrite par une phrase – combinaison de sous-formes – générées par une grammaire qui peut donner toutes les combinaisons légales des formes primitives. Un analyseur syntaxique est utilisé comme procédure de reconnaissance. Afin de traiter l'incertitude de la conversion du signal en symboles primitifs, provoquée par la distorsion du signal, des langages stochastiques et des analyseurs capables de corriger des erreurs ont été développés [Fu 77,Fu 86]. Cependant,

- le phénomène de dépendance contextuelle est une propriété intrinsèque du signal et non une sorte d'erreur, produite par plusieurs sources couplées entre elles et la modélisation par une grammaire est inefficace et inconvenable;
- dans la plupart des cas, la connaissance sur le problème n'est pas suffisamment formalisée pour en déduire une telle grammaire et il est même impossible d'obtenir une grammaire sur la déformation contextuelle.

Dans les techniques markovienne, l'espace de formes est représenté par un ensemble fini d'états. On suppose que l'influence contextuelle à un état donné ne dépend que de l'état précédent; c'est à dire que le phénomène de l'influence contextuelle est modélisable par les probabilités de transition d'état de premier ordre [Levinson 83a]. Cette supposition a permis d'obtenir des solutions du problème très intéressantes en reconnaissance de symboles dans la parole [Tappert 77,Bahl 83] et en reconnaissance des caractères [Kordi 87]. Le résultat de la reconnaissance est nettement meilleur par rapport à celui obtenu sans contexte. Cette technique modélise le contexte par une méthode stochastique sans déduire explicitement des règles contextuelles. Lorsque le vocabulaire des symboles à reconnaître est important, le nombre de combinaisons de contextes devient élevé, le besoin en place mémoire, dû à l'explosion combinatoire, devient exorbitant, et l'apprentissage des modèles représentatifs devient impossible. D'autre part, cette méthode suppose que le phénomène en question obéit à une syntaxe (la structure topologique de la chaîne markovienne) connue [Thomason 86] et imposée, sinon l'inférence des probabilités de transition est impossible. Cette contrainte est trop forte en traitement de la déformation contextuelle.

Le travail de Kohonen [Kohonen 86] [Kohonen 87] sur la correction de l'influence contextuelle constitue une autre voie intéressante. L'essentiel de la méthode est de construire une application de chaînes de symboles erronées à cause du contexte vers des chaînes correctes. Cette application est représentée sous forme de règles de correspondance qui décrivent le contexte d'application et qui sont obtenues par une technique automatique. La méthode est utilisée en reconnaissance de la parole. Pour des vocabulaires de 5000-9000 mots en

mode monolocuteur il a fallu générer 10000-20000 règles. En effet, cette approche consiste à mémoriser toutes les chaînes fausses et leurs versions correctes. Elle revient à mémoriser les caractéristiques du convertisseur signal-symbole sous forme de probabilités de transition. Par rapport au modèle de Markov, on n'impose plus une structre topologique mais le stockage est plus redondant. Puisque la méthode ne cherche pas à modéliser les lois cachées de l'effet contextuel, le comportement du système sera détérioré si la variabilité de la parole est plus importante. En fait, pour le même vocabulaire mais de nouvelles phrases, 70% d'erreurs sont corrigées et pour un texte entièrement nouveau 56% d'erreurs sont corrigées.

Une autre approche au problème est celle de l'intelligence artificielle. On a constaté que, après une phase d'observation et d'examen des phénomènes de déformation contextuelle, l'homme est capable de détecter et de corriger la plupart des erreurs issues de la conversion signal-symbole. Cette capacité humaine est similaire à celle dans la lecture de spectrogrammes de parole [Zue 82,Fohr 86,Carbonell 84,Carbonell 86,Carbonell 85], la reconnaissance de la parole à l'aide des indices acoustiques [Demichelis 83,DeMori 83], ou l'interprétation de signaux de sonar. L'expertise peut être formulée sous forme de règles de production et donnée à une machine capable d'inférencier pour effectuer la correction automatique. Dans cette approche, la qualité des règles est directement déterminée par l'homme qui fournit la connaissance et peut être améliorée au cours de l'expérience.

Dans le but de la conversion signal-symbole, le contexte a été également représenté par la dépendance voisine des décisions [Welch 71], par le modèle stochastique spatial [Fu 80], par des règles de décision en termes de probabilités de transition [Kittler 85]. Le livre de Suen et DeMori présente également d'autres méthodes d'utilisation du contexte [Suen 82].

1.3 Notre travail

L'approche par des règles de production permet d'utiliser, de façon naturelle, la connaissance humaine sur le problème de la conversion signal-symbole et constitue donc une voie prometteuse. Cependant, les problèmes essentiels suivants dans un système de correction de la déformation contextuelle sont encore à résoudre:

- la représentation adaptée des connaissances humaines sur le problème;
- l'utilisation de ces connaissances imprécises, parfois erronées, et le raisonnement inexact;
- la définition et l'utilisation de fonctions qui mesurent les quantités (information numérique) et de traits (information symbolique) nécessaires pour un raisonnement fiable;
- la stratégie d'utilisation des règles de production.

Nous discutons ces problèmes dans ce chapitre.

2 Déformations contextuelles

2.1 Interprétation des formes déformées

Nous définissons d'abord le problème de la déformation contextuelle.
Soient

- $\mathcal{S}$ l'ensemble des symboles connus a priori;
- sig_n $0 \leq n < M$ le signal observé, produit par la séquence de symboles

$$\ldots, s_{k-2}, s_{k-1}, s_k, s_{k+1}, s_{k+2}, \cdots$$

où $\forall i \in \mathcal{N}\ s_i \in \mathcal{S}$;

- $\mathcal{R}$ l'espace de représentation paramétrique;
- $\mathcal{T}$ l'ensemble des contextes, formé par toutes les séquences de symboles dans $\mathcal{S}$;
- $P : \mathcal{S} \times \mathcal{T} \rightarrow \mathcal{R}$, la fonction de projection d'un symbole en présence d'un contexte;
- C: $\mathcal{R} \rightarrow \mathcal{S}$, la fonction de conversion signal-symbole.

Le problème de l'interprétation des formes consiste à déterminer une partition dans le domaine $\mathcal{N}$ tel que:

$$sig_n = \sum_{i=1}^{N} f_{p_{i-1},p_i-1} \text{ avec } p_0 = 0 \text{ et } p_N = M$$

où

- N est le nombre de formes dans le signal et est inconnu en général;
- $f_{p_{i-1},p_i-1} \in \mathcal{R}$ est une forme et doit vérifier

$$(\forall i \in [1, N]) : [(\exists s_j \in \mathcal{S}) : C(f_{p_{i-1},p_i-1}) = s_j]$$

de façon à ce que l'ensemble des connaissances a priori sur la conversion soit cohérent.

2.2 Superposition et dispersion

Superposition

A cause de l'influence contextuelle, la projection n'est plus bijective, il y a notamment le phénomène de superposition et de dispersion. On dit que la projection est superposée si

$$\exists((I \in \mathcal{N}) \wedge (I > 1)) :$$
$$(\exists s_i \in \mathcal{S}, i = 1, \ldots I) \wedge ((\exists \mathcal{R}_0 \subset \mathcal{R} \wedge (\exists t_i \in \mathcal{T}) : P(s_i, t_i) \in \mathcal{R}_0, i = 1, \ldots, I)$$

C'est le cas où plusieurs symboles sont projetés sous la même classe des formes du signal. Ce recouvrement est inséparable en conversion.

Dispersion

On dit que la projection est dispersée si

$$\exists((I \in \mathcal{N}) \wedge (I > 1)) :$$
$$(\exists \mathcal{R}_i \subset \mathcal{R}, i = 1, \ldots, I) \wedge (\forall l \neq m\ \mathcal{R}_l \cap \mathcal{R}_m = \emptyset) \wedge$$
$$((\exists s \in \mathcal{S}) \wedge (\exists t_i \in \mathcal{T}) : P(s, t_i) \in \mathcal{R}_i, i = 1, \ldots, I)$$

C'est le ces où un symbole est projeté sur plusieurs classes de formes du signal.

Conclusion

- La superposition entraîne que la conversion C correcte et utilisant uniquement l'information locale est impossible.
- La dispersion justifie l'utilisation de références multiples.
- Les frontières entre formes sont parfois confondues à cause de l'incertitude.

2.3 Solution: système à base de connaissances

La difficulté de la correction de la déformation contextuelle se caractérise par les deux points suivants:

- Le phénomène est produit par le mécanisme de production du signal au bas niveau d'abstraction. En général, à la reception, il n'y a pas de contraintes suffisantes des niveaux supérieurs, telle que la structure syntaxique du signal, qu'on peut utiliser pour retrouver les symboles initiaux.
- Les relations, les lois auxquelles le phénomène de la déformation contextuelle obéit ne sont pas encore formulées formellement et clairement pour qu'on puisse les utiliser en interprétation.

L'insuffisance de répétition de contexte est souvent un obstacle majeur aux modèles probabilistes car l'estimation d'un tel modèle nécessite une grande quantité d'observations identifiées. En plus, le nombre de combinaisons de formes tenant compte du contexte est tellement grand en pratique que la mémorisation directe de toutes les données nécessaires à la correction est impossible.

Nous considérons le problème de la correction de la déformation contextuelle des formes comme un problème d'interprétation où le sens d'un événement dépend non seulement de l'événement lui-même mais aussi de ses voisins. La solution que nous proposons est de diviser le problème de la conversion en deux étapes:

- Dans un premier temps nous opérons une classification supervisée des signaux. Ce traitement donne des symboles intermédiaires dans l'ensemble $\mathcal{S}_1$. Cette étape réalise une réduction importante de données et fournit un vocabulaire $\mathcal{S}_1$ qui permet de décrire les connaissances sur la déformation.
- Dans un deuxième temps nous utilisons
  - la séquence de symboles dans $\mathcal{S}_1$ obtenue par l'étape précédente,
  - les propriétés supplémentaires associées à chaque symbole et mesurées sur le signal,
  - d'une part les relations entre les symboles dans $\mathcal{S}_1$ et les propriétés, d'autre part les relations entre les propriétés. Ces relations sont obtenues par l'examen de la correspondance entre les élements dans $\mathcal{S}_1$ et les éléments dans $\mathcal{S}$.

Ces relations constituent une base de connaissances et sont utilisées par un système à base de connaissances. L'objectif du système est donc de déterminer une séquence de symboles dans $\mathcal{S}_1$ de façon à ce que l'ensemble des connaissances soit le plus cohérent possible.

Notre système comprend deux parties:

- la partie traitement du signal qui fournit des données à interpréter, elle réalise notamment
  - une préclassification du signal et
  - le calcul des propriétés extraites du signal utilisées dans les descriptions de la déformation contextuelle, lorsque l'interprétation nécessite de l'information supplémentaire.
- la partie manipulation symbolique, un système d'interprétation à base de connaissances, qui donne la séquence de symboles où la déformation contexuelle est corrigée.

3 Représentation des connaissances

La représentation des connaissances est la description de connaissances sous forme d'un langage adapté à la provenance, la structure, le stockage et l'exploitation de ces connaissances.

Le premier problème à résoudre lors de la conception d'un système à base de connaissances consiste à choisir un mode de représentation approprié. Nous allons brièrement présenter certaines méthodes classiques de représentation et discuter leur adéquation au problème de conversion signal-symbole sous incertitude. A chaque représentation de connaissances est associée une méthode d'utilisation sans laquelle la représentation n'a aucun sens. Nous présentons donc également le mécanisme d'exploitation des connaissances représentées.

3.1 Réseaux sémantiques

Le réseau sémantique modélise des entités du monde réel par des nœuds et les relations entre ces entités par des arcs étiquetés et associés à une direction [Findler 79,Woods 72].

Le réseau sémantique exprime explicitement les différentes relations entre les objets par des pointeurs et il suffit de faire un accès par pointeur pour retrouver ces relations des ces objets. Il est particulièrement facile de faire des retours-arrière grâce à ces pointeurs souvent placés dans les deux sens d'un arc liant deux objets. Cette représentation fournit la possibilité de construire des bases de connaissances complexes mais la suppression ou l'adjonction de connaissances pendant la phase du développement du système demande cependant, l'examen et la redéfinition de la base entière à cause de l'interconnection entre les entités. Les réseaux sémantiques sont utilisés dans la compréhension du langage naturel et dans la compréhension de la parole [Pierrel 81] et de l'image [Bunke 84].

3.2 Prototypes

La représentation par prototypes *(frames)* [Aikins 83,Winston 84] et les représentations "orientée objets" sont intéressantes pour représenter des connaissances ayant un degré de structuration élevé ou supposant une dépendance hiérarchique entre les objets. L'idée essentielle de la représentation est de structurer et de mettre ensemble toute information sur un objet à représenter.

Ce type de représentation permet aux objets d'hériter des propriétés des classes d'objets qui sont conceptuellement plus abstraites. L'utilisation des connaissances consiste à instancier les prototypes, en fonction des données présentées. Dans des systèmes basés sur prototypes, les connaissances inférentielles ne sont pas efficacement représentées. Ces connaissances sont plutôt inclues à l'aide d'autres modes de représentation tels que règles de production ou procédures. Ces représentations ont été utilisées avec succès dans des domaines où les connaissances sont très bien structurées par leur nature tels qu'en reconnaissance d'image, vision par ordinateur [Kim 84] ou systèmes experts.

3.3 Règles de production

Les règles de production [Barr 82] sont largement utilisées dans les systèmes experts. Chaque règle dans la base de connaissances constitue un fragment de connaissance dont la validation ne dépend pas d'autres règles. Cette représentation a la propriété que si toutes les règles sont localement ou individuellement correctes alors l'ensemble de la base de connaissance est, en général, globalement correct.

L'accent de la représentation est mis sur les couples situation-action. La conversion des connaissances d'observation vers un formalisme exploitable par la machine est relativement directe. En plus, les règles de production fournissent de manière simple une façon d'expliquer partiellement le comportement du système.

Les connaissances décrites par des règles peuvent être représentées facilement par un graphe ET-OU. Il est démontré que les graphes ET-OU ont une simple correspondance à des grammaires de type contexte libre [Hall 73]. Donc toutes les grammaires de type contexte libre sont adaptées à la représentation par règles.

Les avantages d'organiser des connaissances sous forme de règles de production sont la modularité au niveau de chaque règle et l'homogénité de la représentation sous une syntaxe uniforme. Pour des grands système, les avantages du système à règles de production peuvent

devenir des défauts [Aikins 83] En effet, la manière de la représentation des connaissances est uniforme et monotone. Ce défaut oblige l'utilisateur à exprimer les connaissances de différents natures et de différents niveaux avec la même syntaxe et cache les fonctions spécifiques de ces connaissances.

Un inconvénient des règles de production est le manque de visibilité des structures statiques (ou des relations) entre les objets décrits par l'ensemble de règles. Les règles étant indépendantes l'une l'autre, il n'y a pas de liens explicites entre les règles. Cependant l'ajout, la modification et la suppression d'une règle peuvent avoir une influence importante sur la chaîne de raisonnement. L'indépendance des règles rend cet effet difficile à prévoir. Une autre conséquence du manque de visibilité est que la structure des règles est plate et n'est pas adaptée pour représenter des relations globales du système [Aikins 83]. Pour les systèmes dont l'objectif est d'interpréter des objets en utilisant des connaissances de niveaux d'abstraction faibles, ce point n'a pas d'influence importante.

La solution au problème de structuration des règles est de structurer les règles suivant leur fonction, leur niveau d'abstraction et leur nature. Mais la structuration diminue les avantages.

3.4 Conclusion

Dans notre problème de l'interprétation des formes contextuellement déformées, les connaissances de niveaux d'abstraction élevés fournissent peu de contraintes et ne sont pas utilisées. Les connaissances qui sont bien structurées et qui nécessitent l'héritage de propriétés sont rarement exploitées. Nous n'avons donc pas adopté les réseaux sémantiques, ni les prototypes ou les représentations orientées objets dont la manipulation de la base de connaissances nécessite une implantation préalable complexe et dont le commencement de la construction demande l'existence de l'ensemble complet des connaissances.

Dans la plupart des domaines de l'intelligence artificielle, la représentation de connaissances par des règles de production offre un outil convivial de codification des connaissances qui ne sont pas encore formulées systématiquement. Nous utilisons des règles de production pour représenter notre connaissance sur la déformation contextuelle du signal dans notre système.

4 Traitement de l'imprécision et de l'incertitude

4.1 Introduction

La solution des problèmes liés aux objets réels, tels que l'interprétation de signaux incertains, par l'approche intelligence artificielle nécessite la représentation de deux types de connaissances relatives à la *définition* et à *l'observation* des symboles.

- Les connaissances du premier type décrivent des objets abstraits et des relations entre ces objets. Ces connaissances sont par nature complètement certaines et exactes. Les règles de la grammaire d'un langage en sont un exemple.

- Le second type de connaissances décrit des objets du monde réel et des relations observées entre eux. La correction de ces connaissances est possible mais n'est pas toujours vérifiée car
  - quelque soit l'effort, on utilise un vocabulaire limité pour décrire des phénomènes à variation continue – l'effet de quantification – et les énoncés dans les descriptions peuvent être vagues;
  - les observations sont toujours empiriques et limitées par expérience, on ne voit jamais le monde en entier;
  - l'application de ces connaissances oblige l'identification préalable des objets décrits et donc l'introduction d'un niveau de décision supplémentaire qui contient des erreurs intrinsèques;
  - la base de connaissance n'est qu'une description incomplète des phénomènes qu'elle est destinée à modéliser.

Les techniques de représentation et d'utilisation des connaissances du premier type est actuellement relativement mûre. La logique classique par exemple est adaptée. Par contre, pour le deuxième type de connaissances on vient de réaliser qu'elles peuvent influencer profondément, sinon changer complètement, les opérations définies sur la base de connaissance et par conséquent sur le comportement du système [Mylopoulos 83].

L'imprécision et l'incertitude sont deux aspects intrinsèques liés aux connaissances du type d'observation. Il existe un compromis naturel entre imprécision et incertitude: une connaissance peut être incertaine parce que trop précise, ou cenrtaine parce que suffisamment imprécise [Prade 87]. En effet, on rencontre le phénomène similaire à celui dans la physique quantique: il est impossible de mesurer avec présicion à la fois la position et la vitesse d'un objet.

4.2 Imprécision et prédicats flous

Le monde réel est de complexité illimitée. Les objets naturels (en opposition avec les objets artificiellement fabriqués à partir de modèles), possèdent des propriétés indénombrables. La description d'un objet réel, quelque soit l'effort, ne peut qu'être partielle, incomplète et floue. L'être humain, pour pouvoir raisonner et communiquer avec d'autres individus, a développé des termes, des méthodes, des heuristiques et des habitudes pour traiter ces descriptions.

Certaines connaissances du type d'observation que nous avons acquises sur la déformation contextuelle sont des descriptions linguistiques mais pas numériques. Des variables linguistiques sont souvent utilisées. Par exemple, la fréquence est *élevée*, la position est *très haute*, l'angle est *petit*, etc. Pour exploiter ces connaissances une conversion de la sémantique de ces termes en langage naturel vers une représentation quantitative est nécessaire. Ces variables linguistiques ont des caractéristiques suivantes:

- La limitatation pour que la description soit correcte n'est pas précisée;

- L'intervalle de cette limitation floue varie selon des objets décrits et le contexte de description. Par exemple, avec la même description "grand", les valeurs de *l'intervalle* autorisées dans la description *l'intervalle entre les deux segments est grand* peuvent être très rarement les mêmes que celles autorisées pour la *pente* dans la description *la pente de la fréquence fondamentale du segment*;
- Le nombre de catégories de ces variables linguistiques est petit;
- Ces descriptions représentent une possibilité et non une probabilité. Elles sont imprécises et non aléatoires.

Dans cette application la logique classique présente les problèmes suivants:

- Le résultat d'une inférence a toujours deux valeurs, vrai ou faux. Elle n'a pas de notion de qualité et donc, en présence d'incertitude, elle ne permet pas de comparer qualitativement plusieurs solutions et donner la meilleure.
- Il est impossible de propager l'incertitude dans la chaîne d'inférences.
- Les descriptions des faits imprécis ne peuvent pas être traitées.

Des théories sur les ensembles flous ont été développées pour modéliser et manipuler des objets réels [Zadeh 65]. Ces théories, parmi d'autres, ont permis de généraliser la logique classique à deux valeurs.

Nous modélisons ces descriptions par des prédicats flous dans notre système. Un prédicat flou donne la vérité en valeur réelle continue dans un intervalle. Un prédicat flou est une fonction de l'ensemble d'objets U sur l'intervalle $[0,1]$, avec une restriction R qui donne le degré d'appartenance d'un élément $e \in R$ dans l'ensemble flou F:

$$P_F : U \times R \longmapsto [0,1].$$

Une valeur de 1 d'un prédicat flou signifie une appartenance entière et indique que le prédicat est complètement vrai. Une valeur de 0 d'un prédicat signifie une exclusion entière et indique que le prédicat est complètement faux. Une valeur entre 0 et 1 donne le dégré de certitude du jugement défini par le prédicat. L'étendue du degré d'appartenance spécifiée par R est paramètrable pour représenter les différents types de flou.

La plupart des connaissances du type d'observation défini dans 4.1 peuvent être formulées sous la forme suivante:

Si X_1 est A_1 et X_2 est A_2 et X_3 est A_3 et ...
Alors Y est B
Avec $CF = \alpha$
Action f_1, f_2, ...

où X_i est un attribut d'un objet, A_i est un sous-ensemble flou d'un intervalle réel, CF est le coefficient de certitude de la règle et "Action" est une liste de fonctions à exécuter lorsque la règle est déclenchée. Le prédicat flou est approprié pour représenter ces connaissances.

Représentation des descriptions floues

La transition entre 0 et 1 de la fonction d'appartenance est établie de façon subjective afin de contenir la plupart des descriptions rencontrées par le système. Nous utilisons une fonction numérique qui interpole de façon monotone entre 0 et 1. Puisque les descriptions sont elles-même déjà vagues et imprécises. la forme de cette transition n'a pas d'influence critique sur la performance du système et il n'est pas nécessaire de la spécifier pour chaque ensemble. On peut définir ainsi le prédicat *EstGrand* par exemple comme

$$EstGrand(u,a,c) \equiv \begin{cases} 0 & u \leq a \\ 2\frac{(u-a)^2}{(c-a)^2} & a < u \leq b \\ 1 - 2\frac{(u-c)^2}{(c-a)^2} & b < u \leq c \\ 1 & u > c \end{cases} \tag{5.1}$$

où

$$b = \frac{a+c}{2} \text{ et } a < c.$$

Il est à noter que $EstGrand(a,a,c) = 0$ et $EstGrand(c,a,c) = 1$.

Comme exemple, nous considérons le prédicat flou "EstGrand" utilisé pour décrire un intervalle:

$$EstGrand(Intervalle, Inf, Sup) \rightarrow [0,1]$$

dans lequel

- *Intervalle* est un attribut paramétrique de l'objet dont on parle son intervalle,
- *Inf* est la limite inférieure en desous de laquelle la valeur de EstGrand est strictement égale à zéro signifiant que *l'intervalle n'est pas grand du tout*,
- *Sup* est la limite supérieure au dessus de laquelle la valeur de EstGrand est strictement égale à un signifiant que *l'intervalle est entièrement grand*,

(Inf, Sup) spécifient le sens de EstGrand. Une valeur de EstGrand entre 0 et 1 donne la certitude du prédicat EstGrand.

On peut construire EstPetit et EstEnviron en termes de EstGrand:

$$EstPetit(u,a,c) \equiv 1 - EstGrand(u,a,c) \tag{5.2}$$

et

$$EstEnviron(u, centre, \delta) \equiv \begin{cases} EstGrand(u, centre - \delta, centre) & u < centre \\ EstPetit(u, centre, centre + \delta) & u \geq centre \end{cases} \tag{5.3}$$

Dans Eq-5.3 *centre* est une valeur référentielle par rapport à laquelle on parle de "environ" et δ définit les deux extrémités à partir desquelles u n'est plus du tout considéré comme être à l'approximité du *centre*.

Nous rencontrons essentiellement deux catégories de prédicats flous:

- La première sont des prédicats qualitatifs, tels que EstGrand, EstPetit, EstEnviron, etc.

- La deuxième sont des prédicats comparatifs pour lesquels on donne une référence de comparaison et on spécifie le degré de différence avec la référence. Par exemple, *TrèsSupérieurA*, *TrèsInférieurA*, *JusteSupérieurA*, *etc.* En fait, cette catégorie de prédicats peut être construite à partir des prédicats de la première catégorie. Voici un exemple:

$$TrèsSupérieurA(u,r,a,c) = EstGrand(u\text{-}r,a,c).$$

où r est la référence de comparaison.

Les prédicats flous clarifient l'imprécision et contiennent plus d'information descriptive sur l'observation que des prédicats précis. Ils aident l'utilisateur à spécifier la partie imprécise d'une description et fournissent un outil de conversion des variables linguistiques imprécises vers une représentation de la machine. Les modèles flous sont largement utilisés dans la reconnaissance d'objets réels telle qu'en reconnaissance des phonèmes [DeMori 80,Gubrynowicz 82,Gubrynowicz 86], un problème typique de la reconnaissance de formes déformées par le contexte.

4.3 Incertitude et inférence inexacte

Nous appellons le processus inférentiel dont le résultat n'est ni totalement *vrai* ni totalement *faux* l'inférence inexacte et le mécanisme d'inférence associé le mécanisme d'inférence inexacte .

La qualité d'inférence peut être influencée par deux facteurs liés, dues à l'incertitude:

- Le raisonnement est basé sur des propriétés du signal dont la mesure est parfois biaisée, donc les symboles que l'on utilise pour décrire le signal dans le raisonnement sont incertains. Par conséquent les conclusions obtenues sont jamais absolument fiables.

- Les règles contiennent des observations incertaines et incomplètes et donc ne sont pas certaines. On n'est pas sûr de chaque paire de $< condition, conclusion >$. Donc les conclusions, même si déduites à partir des faits parfaitement certains, sont incertaines.

Il est donc nécessaire de modéliser ces deux sources d'incertitude dans le processus d'inférence.

La théorème de Bayes a été utilisé dans le raisonnement [Duda 76]. Cependant, dans le contexte réel d'inférence,

- il est difficile de vérifier et de maintenir la supposition que l'espace d'hypothèses soit exhaustif et les hypothèses soient exclusives multuellement [Ramsey 86] et

- il n'y a souvent pas assez d'évidence a priori, pour l'estimation des probabilités.

L'estimation des probabilités a priori relatives aux événements décrits par la base de connaissances est problématique.

Les coefficients de certitude sont utilisés avec succès en représentation et en utilisation des connaissances incertaines [Liu 85,Shortliffe 76]. La représentation et la propagation de l'incertitude sont basées sur deux types de coefficients:

- pour modéliser le fait que l'on n'est pas sûr de la cohérence entre prémisse et conclusion, on associe à chaque règle un coefficient de certitude,

- à chaque fait est associé un coefficient de certitude calculé en fonction des faits qui le conditionnent et des coefficients de certitude des règles conduisant au fait,

Nous voyons comment le coefficient d'un fait est calculé:

La résolution d'un problème par réduction du problème consiste à décomposer le problème en sous-problèmes jusqu'à ce que, finalement, le problème original soit réduit en un ensemble de problèmes primitifs solubles immédiatement. La solution d'un problème peut se représenter sous forme d'un graphe ET-OU. Une solution du problème original est en fait un arbre instancié dont toutes les feuilles sont des problèmes résolus.

Comme les règles ne sont pas certaines, les conclusions obtenues sont naturellement incertaines. Le mécanisme d'inférence doit laisser propager correctement l'incertitude et ne doit pas forcer une conclusion d'une manière binaire. Le critère de la propagation est que la conclusion finale soit proche de l'intuition humaine. Dans les deux paragraphes suivants, nous discutons le calcul de coefficient de certitude pour l'arbre ET et l'arbre OU.

arbre ET

Un arbre-ET représente un ensemble de problèmes dont la solution de chacun est nécessaire pour la solution du problème original. Les faits dans la prémisse des règles constituent un arbre-ET. Pour la conclusion d'une règle, ces faits contribuent comme si chacun était une boucle formant une chaîne, - la chaîne sera cassée si une seule boucle est cassée. L'incertitude des faits dans la prémisse d'une règle est propagée vers la conclusion par la formule suivante:

$$CF = C_r \times \min_{\forall p_i \in P} \{p_i\}$$

où P est l'ensemble de certitudes des faits dans la prémisse et C_r est la certitude de la règle.

arbre OU

Un arbre-OU représente un ensemble de problèmes dont la solution de un quelconque peut contribuer à la solution du problème original. En logique classique, il suffit d'aboutir une branche dans un arbre-OU pour que le problème soit résolu. Or, en inférence inexacte, le sens d'un arbre-OU est que plusieurs règles conduisent au même fait. La certitude du fait est donc d'autant plus grande que le nombre des règles validées parmi elles est important. Autrement dit si plusieurs règles impliquent le même fait, la certitude du fait est renforcée et le fait est plus solide. Il est évident que l'ordre de l'application de ces règles n'a pas

d'importance sur la certitude du fait. Nous avons utilisé la formule suivante pour calculer le coefficient de certitude du fait:

$$CF = 1 - \prod_{\forall c_i \in C_f} (1 - c_i)$$

où C_f est l'ensemble de coefficients de certitude obtenus par application séparée des règles conduisant au fait.

4.4 Traitement de la base de règles incomplète

Lorsque nous travaillons sur des objets réels, nous ne pouvons pas connaître tout sur l'univers du problème. Une modélisation complète est ainsi impossible. La base de règles est incomplète dans le sens que tous les faits ou conditions conduisant à une conclusion n'y sont pas mentionnés. Par conséquent il est généralement incorrect de conclure qu'un fait est faux s'il est impossible de prouver qu'il est vrai [Prade 87]. Une décision est particulièrement difficile à prendre lorsque la certitude est proche ni de 1 ni de 0. L'expert humain essaierait de prouver le fait contraire (ou d'éliminer le fait) pour se sortir de cette situation. Au lieu de prouver le fait F il pourrait essayer de prouver non(F). Cette stratégie calcule plus d'informations sur F et donc peut aider à la décision. Dans notre système, nous utilisons deux bases de faits, la BFPA (base de faits prouvés et acceptés) et La BFPR (base de faits prouvés et refusés).

5 Moteur d'inférence

5.1 Représentation des règles

La syntaxe des règles acceptés par le système est donnée dans la figure 5.1.

REGLE ::= si (CONDI) alors (CONSEQUENCE) cv (COEFF)
CONDI ::= [(FAIT) | ((PREDICAT))] CONDI |Λ
CONSEQUENCE ::= concl (CONCLUSION) act (ACTION)
CONCLUSION ::= (FAIT) CONCLUSION |Λ
ACTION ::= S-expression en LISP ACTION |Λ
FAIT ::= chaîne de caractères
PREDICAT ::= S-expression en LISP
COEFF ::= nombre réel compris entre (0,1]

Figure 5.1: la syntaxe des règles

A titre d'exemple, nous donnons une règle utilisée dans le système de reconnaissance des tons du chinois que nous détaillerons dans la section 3 du chapitre 9.

Il s'agit d'interpréter une suite de segments du signal préclassés en termes de 5 formes tonales élémentaires, T_i, $i = 0, .., 5$. Pour cela on utilise des connaissances a priori sur l'effet de la déformation et les informations contextuelles concernant:

- le résultat de préclassification du segment de gauche et du segment de droite,
- les mesures caractéristiques sur le segment précédent, le segment courant et le segment suivant.

Dans cette règle, le segment courant C et le segment précédent P sont préclassés (respectivement comme T_2 et T_3). Le but de la règle est d'éliminer un type de sur-segmentation où le signal du symbole T_3 a été segmenté et classé comme T_3 suivi de T_2. La règle exprime que

- si le segment courant est classé comme T_2
- et le segment précédent est classé comme T_3
- et la durée du segment courant est petite (10,20)
- et l'intervalle entre le segment précédent et le segment courant est petit (8,15)
- et la somme des durées des deux segments est petite (50,60),

alors, avec une certitude de 0.9, le symbole du segment courant est T_3. Une action est ensuite déclenchée pour fusionner les deux segments et pour décrementer le compteur du segment courant. Les nombres dans le prédicat flou "EstPetit" spécifient les limites de la fonction d'appartenance.

```
RULE-011:
(IF    ((in-class C "T2")
        (in-class P "T3")
        (is-small (duration C) 10 20)
        (is-small (interval P C) 8 15)
        (is-small (+ (duration C)(duration P)) 50 60))
THEN   ((is-a C "T3"))
ACTION ((merge P C)
        (decrease POINTER))
RULE-CERT 0.9 )
```

Figure 5.2: exemple de règles

Pour un fait particulier, il est possible que plusieurs règles soient applicables. Cette augmentation de redondance permet de conforter des situations incertaines.

5.2 Algorithmes

On peut aborder le problème d'interprétation des formes déformées par d'abord émettre des hypothèses sur l'existence de toutes les formes à reconaître, et ensuite verifier ces hypothèses en exploitant des règles et des observations sur le signal. Dans un système plus complet, des connaissances issues des niveaux supérieurs peuvent réduire le nombre des hypothèses. Le mécanisme d'inférence que nous avons construit est donc un moteur fonctionnant en chaînage arrière [Farreny 85] qui prouve tous les faits possibles initialement hypothésés. Il exploite des branches dans le graphe d'inférence en profondeur d'abord. Le résultat est la liste d'hypothèses soutenues par les faits mesurables.

```
possibles = liste d'hypothèses (faits) à prouver: liste de faits
ProuverTout(possibles) → liste de (fait,certitude)
    Si Vide(possibles) Alors VIDE
    Sinon
        UnFait = ProuverUnFait(Premier(possibles));
        Si UnFait
            Alors AjouteTête(UnFait,ProuverTout(Reste(possibles)))
            Sinon ProuverTout(Reste(possibles));
        FinSi;
    FinSi;
```

Nous distinguons trois cas lors de la preuve d'un fait. Un fait à prouver peut être un fait déjà prouvé et établi, et donc présent dans la base de faits établis. Il peut être déjà prouvé mais rejeté et donc stocké dans la base de faits rejetés. Ou alors le fait n'a jamais été rencontré depuis l'initialisation des deux bases de faits, dans ce cas il sera prouvé ici. Prouver un fait signifie vérifier d'abord les règles qui le soutiennent, ensuite celles qui soutiennent son contraire puis déterminer si on l'établit ou si on le rejette, en fonction de son coefficient de certitude. Si aucune règle n'est applicable pour le fait, ce fait est un fait terminal mesurable et on déclenche des fonctions de calcul par l'appel de la fonction Mesurer pour vérifier son existence dans le signal. La conclusion du raisonnement est mémorisée dans la base de faits établis ou dans la base de faits rejetés selon le résultat de l'inférence. Le résultat est le couple $< fait, certitude >$ si prouvé sinon VIDE. La fonction Recherche(f,bf) retourne le couple $< f, certitude >$ dans bf. La fonction DansConc(f) retourne toutes les règles dont la partie conclusion contient le fait f.


```
lefait = le fait à prouver: fait
BaseFaitsE = base de faits établis
BaseFaitsR = base de faits rejetés
ProuverUnFait(lefait) → (fait,certitude)
    Si Dans(lefait,BaseFaitR) Alors VIDE
    Sinon Si Dans(lefait,BaseFaitE)
        Alors Recherche(lefait,BaseFaitE)
        Sinon                RègleP = DansConc(lefait);
            RègleN = DansConc(Contraire(lefait));
            CertCoeff =
                Si Null(RègleP) et Null(RègleN)
                Alors Certitude(Mesurer)
                Sinon
                    C_p = ArbreOu(RègleP,VIDE);
                    C_n = ArbreOu(RègleN,VIDE);
                    CertFinal(C_p, C_n);
                FinSi;
            Mémoriser(lefait,CertCoeff);
        FinSi;
    FinSi;
```

La fonction Mémoriser décide si un fait est établi ou rejeté selon le coefficient de certitude. Elle retourne le couple $< fait, certitude >$ si le fait est établi ou VIDE s'il est rejeté.

```
SeuilRejet = 0.3: Constant
Mémoriser(lefait,CertCoeff) → (fait,certitude)
    Si CertCoeff < SeuilRej
    Alors
        BaseFaitR = AjouteTête((lefait,CertCoeff),BaseFaitR);
        VIDE;
    Sinon
        BaseFaitE = AjouteTête(<lefait,CertCoeff>,BaseFaitE);
        (lefait,CertCoeff);
    FinSi;
```

Le coefficient de certitude d'un fait est limité à l'intervalle [0,1]. La formule suivante est utilisée pour combiner le coefficient de certitude du fait C_p et celui du contraire du fait C_n. Dans cette formule R est une valeur entre [0,1] prédéfinie.

$$CertFinal(C_p, C_n) = R + (C_p - C_n) \times (1 - R)$$

Pour prouver un fait en présence d'incertitude, il est absolument nécessaire de parcourir exhaustivement toutes les branches constituant un "OU" dans le graphe d'inférence et ensuite de synthétiser la certitude obtenue du fait en question. S'il existe plusieurs branches donnant des évidences sur le fait, la certitude sera renforcée. Cette tâche est réalisée par la fonction

ArbreOu qui retourne la certitude finale. Ceci est la différence essentielle entre un mécanisme d'inférence capable de traiter l'incertitude et la logique à deux valeurs.

```
Règles = liste de règles concluant un fait: liste de règles
ListeCert = liste de certitudes données par des règles déjà utilisées
ArbreOu(Règles,ListeCert)
    Si Null(Règles)
    Alors CertOu(ListeCert)
    Sinon
        UnCert = UneRègle(Premier(Règles));
        ArbreOu(Reste(Règles),AjouteTête(UnCert,ListeCert));
    FinSi;
```

La validation de chaque règle se fait par la vérification de tous les faits dans la partie prémisse de la règle et la multiplication du coefficient de certitude obtenu par le coefficient d'atténuation de la règle. Si tous les faits sont vérifiés, les actions associées à la règle sont exécutées.

```
LaRègle = la règle à valider: règle
CertSeuil = seuil pour accepter la validation d'une règle: constante
UneRègle(LaRègle) → certitude
    Faits = Prémisse(LaRègle);
    Cert = CertEt(ArbreEt(Faits,VIDE)) × Atténuation(LaRègle);
    Si Cert > CertSeuil
    Alors Exécuter(Action(LaRègle));
    FinSi;
    Cert;
```

La fonction ArbreEt vérifie successivement tous les faits passés en premier argument. Une récursivité croisée est utilisée entre cette fonction et la fonction ProuverUnFait. Les branches de l'arbre "ET" sont examinées jusqu'à la première impasse, – soit un fait ne peut pas être établi soit la certitude du fait est trop faible. Le résultat est une liste de certitudes correspondant aux faits à l'entrée.

```
Faits = liste de faits à vérifier: liste de faits
Certs = liste de coefficients de certitude des faits déjà examinés
ArbreEt(Faits,Certs) → liste de CoeffCert
    Si Null(Faits)
    Alors Certs
    Sinon
        FaitCert = ProuverUnFait(Premier(Faits));
        Si Null(FaitCert)
            Alors VIDE
            Sinon ArbreEt(Reste(Faits),AjouteTête(Certitude(FaitCert),Certs));
        FinSi;
    FinSi;
```


Les fonctions CertOu et CertEt sont des fonctions qui combinent des coefficients de certitude venant de plusieurs branches d'un nœud "OU" (CertOu) ou d'un nœud "ET" (CertEt) du graphe d'inférence dont les formules ont été décrites antérieurement dans 4.3.

L'entrée du système est une liste de segments. En général, la prémisse d'une règle décrit le segment courant à interpréter et le contexte du segment: Le segment gauche, ou précédant, et le segment droite, ou suivant. Comme l'influence contextuelle décroit généralement rapidement lorsqu'on s'éloigne du segment courant, il est rare qu'on se serve d'autres segments pour décrire le segment courant. L'interprétation se fait de la gauche vers la droite du signal mais elle est capable d'empiler les sous problèmes créés pour l'interprétation du segment courant afin d'interpréter le segment gauche ou droite.

5.3 Interface au signal

Dans 2.3 nous avons expliqué que la partie de raisonnement nécessite deux types de données, les symboles préclassés et les propriétés primitives mesurées sur le signal. A travers l'interface, ces données sont fournies à l'interprète.

Les données du premier type sont obtenues par une classification basée sur la comparaison avec des formes de référence. Pour garder l'information sur la sûreté de la préclassification, on associe a chaque symbole un coefficient indiquant avec quel degré de vraisemblance le signal est préclassé comme tel symbole.

L'interprétation des formes utilise des connaissances du type d'observation. Ce sont des descriptions sur la morphologie du signal dans un certain espace de représentation. Les descriptions sont basées sur un certain nombre de propriétés, ou de paramètres de base du signal. Nous avons construit un ensemble de fonctions qui extraient ces propriétés que l'homme utilise dans sa perception du signal.

La liaison entre la partie traitement du signal et la partie traitement symbolique est interactive permettant

- l'utilisation des connaissances du niveau inférieur dans les références de nature statistique pour réduire efficacement la quantité de données, et
- l'application du traitement symbolique seulement sur les segments où une interprétation du niveau supérieur est nécessaire.

6 Conclusion

Le problème de l'interprétation de formes déformées par le contexte ne peut pas être résolu par l'utilisation unique de techniques de classification où seul l'information locale est exploitée. Un système basé sur des connaissances spécifiques sur le phénomène de la déformation est nécessaire. Ce système permet d'inclure des connaissances du type d'observation et des connaissances liées avec le mécanisme interne de la production du phénomène de la déformation contextuelle. Il est indispensable d'étudier la représentation et l'utilisation des connaissances du domaine, le traitement du signal, la modélisation des descriptions floues, et l'inférence approximative.

Notre travail présenté dans ce chapitre insiste sur les points suivants:

- La représentation et l'exploitation de la connaissance sur la déformation contextuelle par des règles de production.
- L'application d'un mécanisme d'inférence inexact capable de traiter l'incertitude des données pour éviter que la décision soit trop "binaire".
- La modélisation de l'imprécision linguistique dans la description des phénomènes par les prédicats flous.
- L'utilisation des techniques d'I.A. dans l'interaction entre le traitement symbolique et le traitement du signal de façon à ce que l'extraction des primitives soit guidée par des buts de raisonnement, permettant ainsi de réduire les calculs.

L'application à la reconnaissance des tons du chinois montre que notre système peut servir comme un outil de raisonnement incertain. Nous constatons

- que les prédicats flous sont un moyen souple et efficace pour représenter des connaissances imprécises,
- que l'examen des symboles préclassés et des formes dans leur contexte a amélioré la qualité d'interprétation, et
- que le système est un outil utile d'acquisition des connaissance du type d'observation.

Le résultat expérimental du système sera présenté dans le chapitre 9 Cependant, l'acquisition de la connaissance complète sur le problème sous forme de règles de production demande un travail considérable. Il serait souhaitable que cet apprentissage du système soit en partie automatisé.

Partie III

ANALYSE DE STRUCTURE DU SIGNAL

Chapitre 6

Analyse de Structure sous Incertitude

L'incertitude du signal rend impossible la détermination fiable de deux informations élementaires dans l'analyse de la structure du signal: les terminaux et les séparateurs de terminaux. Ceci demande un nouveau mécanisme d'inférence. Nous présentons un moteur d'inférence qui est capable de mener le raisonnement en mode mixte ascendant et descendant, de mesurer et de propager l'incertitude du signal jusqu'au résultat final, de privilégier l'analyse à partir d'îlots de confiance et de fusionner les interprétations partielles, enfin de développer en parallèle plusieurs interprétations plausibles et d'effectuer une stratégie de recherche en faisceau.

1 Introduction

La structure d'un signal se décompose récursivement en sous-structures et en symboles primitifs et peut être engendrée par une grammaire, donnée sous forme de règles qui précisent la manière d'écrire ou de disposer l'information transportée par le signal. L'interprétation du signal est ainsi guidée par la structure syntaxique du message.

La difficulté essentielle de l'analyse de la structure du signal est la conversion incertaine du signal en symboles élémentaires, ce qui conduit a une reconnaissance peu fiable des terminaux et des séparateurs. L'incertitude est provoquée par le bruit de transmission, l'insuffisance du modèle de signal ou l'influence contextuelle. L'indéterminisme de la conversion induit soit un refus des phrases légèrement perturbées, soit une explosion combinatoire des solutions admissibles, si un analyseur syntaxique classique, tel que ce qu'on a développé pour les langages de programmation, est utilisé.

Il est donc nécessaire d'utiliser des mécanismes adaptés d'inférence. Les différentes solutions peuvent se regrouper en deux catégories:

- Une première voie consiste à apporter des améliorations sur la qualité de l'étape de l'identification de symboles terminaux, l'analyseur restant le même. Cette solution a le défaut de forcer le reconnaisseur de symbole à prendre des décisions très tôt malgré

l'ambiguïté en n'utilisant que des connaissances locale de bas niveau. Or avons montré dans le chapitre 5 que cette ambiguïté ne peut pas être enlevée localement.

- Dans la deuxième voie l'incertitude est traitée par l'autorisation explicite des erreurs due à la reconnaissance de symboles et par la modélisation de ces erreurs par une grammaire. L'exemple typique est l'analyse grammaticale avec correction des erreurs [Fu 77]. L'idée de base de cette méthode est d'augmenter la grammaire des structures à analyser de façon que toutes les dérivations possibles dues aux erreurs sur les symboles primitifs soitent inclues dans les phrases générées. Ces erreurs peuvent être des erreurs de substitution, d'insersion ou d'omision. L'analyseur lui même reste en principe un analyseur ordinaire. En effet, la méthode consiste à inclure dans la grammaire la propriété de transformation du convertisseur signal-symbole. Lorsque l'incertitude du signal est grande la croissance de la complexité de l'analyse devient importante. D'autre part, du fait que les erreurs produites dans l'étape conversion signal-symbole sont instables par rapport à la variation du signal, la modélisation correcte et complète des erreurs est un travail considérable.

Notre philosophie sur l'analyse de structure du signal sous incertitude est différente de celle traditionnelle. Nous insistons sur les deux problèmes suivants:

- l'utilisation efficace de l'information obtenue pendant l'analyse afin de guider l'interprétation;
- la suppression progressive de l'incertitude du signal sur la position et la nature des terminaux.

Nous pensons que la décision doit se situer au niveau de l'interprétation finale et non au niveau de la conversion du signal en symboles primitifs, car l'unité de symboles primitifs est trop petite pour être fiable. Les décisions prises a ce niveau sont trop fragiles. Au lieu de dire qu'avec tel ensemble de symboles on peut construire telle structure, nous essayons de dire que le signal semble avoir telle ou telle structure. Dans cette conception, la phase de conversion signal-symbole n'est qu'un codage d'information pour réduire la quantité de données et pour effectuer la préparation des raisonnements. Ainsi, il n'y a pas d'erreurs à corriger, telles que dans l'approche d'analyse syntaxique à correction d'erreurs, il n'y a que des qualités différentes entre différentes interprétations possibles.

Nous partons du principe que l'incertitude existe dans la conversion signal-symbole et doit exister à travers tout le processus d'interprétation jusqu'aux solutions finales. En conséquence, nous introduisons la notion de mesure de qualité d'interprétation et nous évitons des décisions locales prématurées.

Nous considérons que dans l'analyse, lorsque une décision ne peut pas être prise avec certitude, toutes les branches issues de cette décision doivent être développées. Ces branches sont exploitées jusqu'à ce que l'évidence soit suffisante pour prendre une décision fiable.

Notre analyseur adapté au signal incertain a les caractéristiques suivantes:

- utilisation des connaissances syntaxiques et sémantiques sous forme d'une grammaire sémantique,

- capacité à utiliser des règles recursives, à gauche ou à droite,
- les symboles terminaux sont reconnus de façon à associer au resultat un coefficient de vraisemblance,
- l'analyse peut démarrer à plusieurs endroits temporels du signal, choisis de façon que leur taux de vraisemblance associés aux terminaux soient les meilleurs; les arbres partiels sont fusionnés au cours de l'analyse,
- le raisonnement est mixte, en chaînage avant et arrière,
- devant chaque choix possible (branches du type "OU"), le raisonnement se poursuit en parallèle en cas d'incertitude sur les terminaux,
- une stratégie de recherche en faisceau conserve les meilleures interprétations.

2 Mesure de l'incertitude du signal

En présence de l'incertitude, l'identification des symboles ou la conversion signal-symbole ne peut pas être réalisée avec sûreté. L'incertitude du signal temporel à une dimension peut être mesurée quantitativement par *l'incertigramme* défini de la façon suivante.

A chaque instant t du signal s, on effectue une conversion signal-symbole

$$C : S \longmapsto V$$

du domaine du signal S vers le domaine des symboles à vocabulaire V. A cause de l'incertitude, l'application C a comme résultat des valeurs multiples

$$C[S(t)] \rightarrow \hat{v}_i \in V, \ \ i = 1, 2, ...n$$

à chaque valeur est associée une fonction d'appartenance

$$0 \leq \mu(t, \hat{v}_i) \leq 1$$

qui mesure le degré d'appartenance du signal à la classe de symbole $\hat{v}_i$. Nous construisons la série

$$I_g(t, i) = \mu(t, v_1), \mu(t, v_2), ...\mu(t, v_n) = \mu(t, v_i)$$

de telle manière que

$$I_g(t, i + 1) \leq I_g(t, i) \ \ \forall i$$

où

$$\{v_i\}_{i=1,2,..n} = \sigma\{\hat{v}_i\}_{i=1,2,..n}$$

est une permutation de $\{\hat{v}_i\}_{i=1,2,...n}$. $I_g(t, i)$ est l'incertigramme du signal $s(t)$.

On définit ensuite l'incertitude de l'ordre m du signal à l'instant t comme la moyenne d'ordre m de la première dérivée de la série $I_g(t, i)$ par rapport à i:

$$I_m(t) = 1 - \frac{1}{m} \sum_{i=1}^{m} (\mu(t, v_{i+1}) - \mu(t, v_i)) \quad m < n.$$

$I_m(t)$ est d'autant plus grand que le signal est incertain; en particulier:

- $I_m(t) = 1$ implique que $\mu(t, v_i) = \mu(t, v_j) \forall i, j < m$ et indique que le signal est complètement incertain car il y a pas de différence entre les valeurs des fonctions d'appartenance;
- $I_m(t) = 0$ implique que $\mu(t, v_1) = 1$ et $\mu(t, v_i) = 0 \ \ i > 1$ qui signifie que le signal est complètement certain car il y a pas d'ambigüté pour associer le signal à l'instant t $s(t)$ à un symbole unique.

3 Analyseur de structure syntaxique

L'objectif de l'analyseur syntaxique en interprétation de signal est de construire un (ou plusieurs) arbre(s) représentant la structure syntaxique du signal en cours d'analyse. Le processus de construction est aussi appelé interprétation. Un constructeur de la structure syntaxique peut être vu comme l'ensemble des composantes actives qui fonctionnent sur un modèle commun de description de problème et qui sont soumises à un contrôle commun. Les données sont de deux types: celles qui donnent les contraintes sur le modèle et celles qui donnent les contraintes sur les primitives, ou les symboles terminaux, fournies par le niveau d'abstraction inférieur à partir d'une signature partiellement reconnue et identifiée du signal. L'incertitude du signal oblige le constructeur de structure du signal à tenir compte des aspects suivants:

- l'introduction d'heuristiques de recherche dans la construction. La construction devient plus efficace mais il ne peut pas garantir la meilleure solution;
- l'augmentation de la dimension de l'espace de recherche. En effet, l'incertitude du signal exige le développement de plusieurs lignes de raisonnement et l'utilisation de techniques de recherche en faisceau car il est nécessaire de comparer différentes solutions avant de choisir les meilleures interprétations retenues;
- la stratégie de raisonnement en chaînage mixte; le moteur doit être capable de mener l'interprétation en modes chaînage avant et arrière afin d'utiliser du mieux l'information structurelle du signal obtenue au cours de l'analyse;
- l'exploitation du concept "îlot de confiance"; L'interprétation commence par quelques primitives dont la qualité d'identification est relativement élevée. Ceci permet d'augmenter le pourcentage du nombre d'arbres corrects sur le nombre total d'arbres générés et développés;
- la conception du processus d'interprétation permettant de procéder à la construction dans le cas où quelques primitives sont très mal reconnues.

Les analyseurs syntaxiques traditionnels sont incapables d'analyser les structures contenant des terminaux erronés, comme c'est le cas en reconnaissance et interprétation de parole et d'images [Reddy 74,Fu 77]. Plusieurs auteurs ont consacré leurs efforts à l'analyse syntaxique capable de commencer son analyse à un terminal quelconque dans la grammaire. Ce type d'analyse améliore la performance de l'analyseur quand la detection de terminaux n'est pas fiable [Miller 74,Woods 79] (le système de BBN HWIM), [Lesser 75] (le système

HEARSAY II), [Haton 76,Mari 79] (le système du CRIN MYRTILLE II), [Masini 85] (le système de vision TRIDENT)]. En particulier, fondé sur un modèle decrit par des règles définissant les relations et des contraintes entre des objets, le formalisme de Gérald Masini [Masini 85] combine l'analyse ascendante et l'analyse descendante. A un moment donné, l'arborescence représentant un objet partiellement reconnu peut être étendue en greffant une feuille d'un autre arbre sur la racine ou une racine d'un autre arbre sur une de ces feuilles. L'information obtenue au cours de l'analyse est utilisée pour guider l'interprétation et pour lever les ambiguïtés. L'indéterminisme est résolu par retour-arrière, permettant d'introduire des stratégies de construction et d'ajouter des heuristiques et des contraintes pour réduire la complexité. Cette methode est puissante et souple.

L'idée de départ de notre travail est inspirée de ce travail. Nous insistons cependant l'aspect incertain du signal à interpréter. D'où proviennent deux différences au niveau de la stratégie:

- La première est sur la *stratégie de construction*. Nous visons à maintenir simultanément plusieurs interprétations plausibles et non une seule,
- La seconde est sur la *stratégie de recherche*. Une analyse en faisceau fondée sur la qualité d'interprétation partielle est utilisée au lieu du mécanisme de retour-arrière.

Nous allons maintenant justifier notre choix et présentons notre approche.

4 Stratégie d'interprétation

4.1 Définition

L'interprétation du signal consiste à spécifier progressivement le signal par des strutures connues. L'incertitude du signal et la complexité du problème nécessitent l'utilisation de l'information dépendant du signal afin d'aider à la réduction de l'espace de solutions recherché [Kanal 86]. A chaque étape de l'interprétation, la décision sur l'étape suivante

- n'est pas explicitement définie à cause des choix multiples,
- ne peut pas être définie avant que certaines informations sur le signal ne soient acquises.

Une stratégie est un ensemble de principes qui détermine, en fonction de l'état de recherche et de l'information dynamique, l'étape suivante pour conduire l'interprétation vers les solutions finales.

4.2 Chaînage avant, arrière et mixte

Chaînage avant

Partant des faits mesurables et des faits établis, l'inférence en chaînage avant consiste à appliquer des règles de façon exhaustive et récursive jusqu'à ce que l'ensemble des faits déduits soit stable. Ce type d'interprétation tente de synthétiser les évidences existantes et

d'en déduire d'autre par raisonnement. Sans objectif, le but du raisonnement est aléatoire. Selon l'espace de recherche défini par l'ensemble de règles, ce type d'interprétation est restreint par sa localité de vue qui entraîne des risques de perdre beaucoup de temps pour construire des structures partielles qui seront finalement abondonnées à cause des évidences venant d'une autre région. Le chaînage avant est inefficace lorsque le nombre de règles qui mentionnent les mêmes faits terminaux ou non, en parties droites des règles est important.

Chaînage arrière

L'inférence en chaînage arrière tente d'arriver à un but en divisant le but en un ensemble de sous-buts plus simples et en réduisant récursivement ces sous-buts jusqu'à des tâches élémentaires. Ce processus récursif peut avoir une profondeur importante. Avant d'arriver aux faits mesurables, il est impossible de vérifier des hypothèses et donc impossible de supprimer celles qui sont invalides. Un nombre prohibitif d'hypothèses en attente risque donc d'être généré. Le raisonnement peut construire beaucoup de structures intermédiaires avant de constater que la plupart d'entre elles ne se justifient pas par l'observation du signal. Ignorant complètement ce qui existe au plus bas niveau, et par conséquent risquant de développer des chemins de raisonnement partiels abérants par rapport au faits primitifs [Naylor 85], le chaînage arrière est inefficace lorsque le nombre des règles dont la validation nécessite une chaîne d'inférence profonde est important.

Chaînage mixte

Le raisonnement de l'être humain ne fonctionne jamais sous forme purement avant ou arrière. Il y a toujours une interaction entre l'intuition et l'examen de la réalité.

Les mécanismes d'inférence uniquement dotés d'un mode de raisonnement en chaînage avant ou en chaînage arrière ne sont pas efficaces, comme on vient de le voir pour des problèmes d'interprétation de données complexes. En effet, dans ce cas, une interaction entre le modèle défini par les règles et les données doit être réalisée pour que le raisonnement soit à la fois guidé par le modèle et contrôlé par les données.

Le chaînage mixte consiste à mélanger les deux types de raisonnement purs pour construire des solutions de manière efficace. Le processus commence à partir de quelques faits relativement sûrs et utilise la notion de sous-but pour diriger l'interprétation. Un mécanisme d'inférence capable de mêler les deux types de raisonnement est très souhaitable en interprétation du signal.

4.3 Interprétation en parallèle

Introduction

Le processus d'interprétation doit rechercher la structure syntaxique la plus proche de celle du signal observé. Après avoir décomposé la structure vers les niveaux inférieurs en primitives mesurables, la cohérence est évaluée par examen d'une suite d'états. La recherche d'un chemin optimal dans la suite d'états garantissant la meilleure solution demande en général des opérations de complexité expotentielle.

Notre idée de base sur la réduction de calcul est d'utiliser au mieux l'information portée par le signal en cours d'analyse pour guider l'interprétation. Nous effectuons notamment

- le développement en parallèle de plusieurs structures plausibles selon le signal observé; l'espace de recherche n'étant examiné qu'en partie,
- la recherche de solution en faisceau ce qui permet de ne conserver que les interprétations de meilleure qualité.

Dans les paragraphes qui suivent, nous justifions la nécessité de l'interprétation en parallèle en montrant la différence entre **l'indéterminisme** et **l'incertitude**. Nous indiquerons qu'en présence d'incertitude, le retour-arrière est insuffisant pour donner la meilleure solution et l'interprétation en parallèle est indispensable.

Correction locale et globale

Le processus d'interprétation consiste à mettre en correspondance une instance générée par un modèle multi-niveaux et le signal observé. La mise en correspondance vérifie l'existence des propriétés supposées par le modèle dans le signal. Dans le signal physique, les propriétés ne sont pas énumérables. Un système d'interprétation se contente de n'en verifier qu'un sous-ensemble qui sont considérées comme caractérisantes et pertinentes. La vérification se fait dans un système complet en plusieurs phases, en plusieurs niveaux de concepts. Une interprétation est dite *correcte* si et seulement si toutes les propriétés prédites par le modèle sont cohérentes avec l'observation pendant la vérification. Une interprétation pour laquelle toutes les propriétés sont vérifiées jusqu'à un moment donné au cours de la vérification est dite *localement correcte*.

L'interprétation de signal est un processus nécessitant une coopération de sources multiples de connaissances [Haton 85]. Ces sources de connaissances sont par nature réparties en niveaux différents d'abstraction. La compétence de chaque source de connaissance est limitée dans un domaine de spécialité particulier. Au cours de l'interprétation, certaines ambiguïtés ne peuvent pas être résolues par l'application des contraintes locales. Une décision prise par une source de connaissance n'est qu'un point de vue, fonction de sa propre connaissance. Cette décision n'est donc pas forcément jugée correcte par d'autres sources de connaissances, en particulier par celles appartenant à des niveaux supérieurs. Les décisions dans le système peuvent être correctes localement mais pas nécessairement globalement. Si on force le système à donner une solution unique en présence d'incertitude on risque alors de perdre les solutions correctes et d'introduire des erreurs d'interprétaion irrécupérables par les niveaux supérieurs, même si cette décision est localement optimale. Par exemple, toutes les séquences des mots syntaxiquement correctes ne sont pas correctes sémantiquement. Par conséquent, pour ne pas perdre l'interprétation correcte, il faut conserver les solutions concurrentes à même niveau donné, le choix étant fait ultérieurement par d'autres sources de connaissances.

Indéterminisme et incertitude

Il peut apparaitre des situations d'ambiguïté dans lesquelles plusieurs solutions sont équiprobables. L'ambuiguïté est provoquée par deux raisons – l'indéterminisme ou l'incertitude:

- **L'indéterminisme** du modèle. A une étape du processus d'interprétation le modèle indique plusieurs alternatives devant une situation. Par exemple dans un système à règles de production, plusieurs règles sont applicables et il est localement impossible de savoir quelles sont les règles qui donnerons l'interprétation correcte globalement.
- **L'incertitude** de la reconnaissance des primitives due à l'incertitude du signal. Les primitives sont reconnues seulement en fonction de connaissances locales et il se peut que globalement ces primitives soient en conflits avec des évidences venant des autres regions. Dans ce cas, une interprétation en parallèle est obligatoire. La décision sera prise lorsque l'évidence devient suffisante.

Nous avons vu que l'ambiguïté provient de deux sources: l'indéterminisme des modèles (notamment des règles de production) et l'incertitude du signal. Dans le cas où le signal est certain (l'existence d'une primitive peut être affirmée ou infirmée), la notion de qualité est évidemment inutile. Pour pallier l'indéterminisme des règles, le mécanisme de retour-arrière en cas d'impasse est suffisant. Il suffit de trouver la première instance de modèle qui est cohérente avec le signal.

Or, en présence d'incertitude, il faut en plus donner la meilleure ou les meilleures instances. Un simple retour-arrière donnant une solution unique est insuffisant. Cette insuffisance est non seulement due au fait que l'on doit fournir plusieurs interprétations pour des décisions ultérieures mais aussi à la capacité de décision locale. En particulier, il est possible qu'un tel mécanisme trouve une interprétation dont le score dépasse la limite inférieure et l'annonce comme résultat alors qu'il existe une ou plusieurs interprétations qui seraient, si la recherche avait été poursuivie, de meilleur score. Il résulte de ce qui précède que, avec le mécanisme de retour-arrière, la meilleure interprétation n'est pas garantie. Ceci est du au fait que le choix dans retour-arrière est purement local.

En conséquence, lorsque l'on n'a pas plus de raisons de garder une interprétation particulière qu'une autre, la réaction correcte consiste à reporter la décision jusqùà ce que suffisamment d'évidence soit acquise. Avant cette décision, toutes les alternatives sont exploitées parallèlement.

Plusieurs auteurs ont constaté que la recherche en faisceau s'avère en général plus efficace que la recherche avec retour-arrière [Quinton 82,Lowerre 76,Charpillet 85,Haton 85].

Hypothèse fondamentale

On peut se poser une question: "Le développement parallèle de plusieurs interprétations provoque-t-il une explosion combinatoire?" Notre réponse est fondée sur l'hypothèse suivante:

> *Si un chemin dans le graphe de recherche ne correspond à aucune interprétation correcte, la qualité de la branche ainsi construite diminue rapidement.*

La qualité d'une branche est une mesure de la cohérence entre entre le modèle instancié représenté par la branche et le signal observé.

Sous cette hypothèse, si une branche poursuivie n'est pas une interprétation correcte alors au fur et à mesure de l'interprétation, la qualité se détériore de plus en plus. Il est donc possible de supprimer les instances incohérentes et d'éviter l'explosion combinatoire. La recherche en faisceau, qui élimine les branches de faible qualité et augmente la possibilité que l'interprétation correcte finalement soit conservée, est utilisée pour contrôler le processus d'interprétation.

4.4 Ilots de confiance

Un îlot de confiance est un arbre d'interprétation partielle contenant des symboles terminaux et non-terminaux instanciés sur une zone de signal dont les symboles primitifs sont reconnus avec une plausibilité relativement elevée.

En interprétation de signal incertain, on peut démarrer le processus en s'appuyant sur plusieurs îlots de confiance qui s'agrandissent en cours d'interprétation par extension des arbres en chaînage avant et arrière. On cherche alors à fusionner des arbres partiels afin de former des arbres plus complets. L'interprétation se termine lorsque tous les arbres partiels sont unifiés et qu'on atteint l'axiome des règles.

La notion d'îlot de confiance est raisonnable au sens

- que l'on attaque le problème d'interprétation d'abord par les zones dont l'interprétation est la moins difficile et
- que l'on peut rapidement avoir une vue morcelée mais globale sur la structure du signal pour guider le processus d'interprétation. Il y a donc une meilleure utilisation de l'information obtenue pendant l'interprétation.

Un îlot a les propriétés suivantes:

- Toute les contraintes locales (niveau d'abstraction inférieur du niveau de l'îlot) sont vérifiées.
- Au moins un fait primitif est identifié et qualifié. Ces faits constituent une zone de signal reconnue.
- Il est garanti qu'un îlot est la meilleure construction structurelle parmi toute les possibilités depuis le commencement de l'interprétation, au sens d'un critère spécifique.

La notion de l'îlot de confiance est attrayante mais la réalisation est difficile. Les difficultés principales concernent:

- le choix des îlots de départ. Par exemple, comment estimer la qualité de la reconnaissance des symboles primitives? quels sont les zones où l'on peut faire croitre les arbres rapidement et avec de meilleurs taux de réusite?

- la fusion des arbres partiels. On risque de développer inconsciemment plusieurs arbres identiques construits sur deux îlots de confiance de départ différents.
- le contrôle du processus d'interprétation. Par exemple, le nombre convenable des îlots de départ pose souvent des problèmes. Si ce nombre est trop grand, le comportement de l'analyseur s'approche du chaînage avant. L'aspect prédictif du modèle ne peut pas être exploité normalement car au départ les îlots sont indépendants. On risque de construire beaucoup d'interprétations partielles qui sont incorrectes globalement. En revanche, si ce nombre est trop petit, pendant le processus d'interprétation, l'analyseur manque d'une vue g lobale et donc ne peut pas utiliser immédiatement l'information sur la structure du signal pour guider l'interprétation.

4.5 Propagation de l'incertitude

L'ensemble des interprétations peut être décrit dans un espace de représentation. Le problème de l'interprétation consiste à indiquer, étant donné un signal observé, une ou plusieurs interprétations possibles. Cette indication est equivalente à la définition d'un sous-espace de représentation. Nous appellons les actions dans la determination d'un tel sous-espace des ***décisions***. Une décision est prise sur deux types de connaissances:

- les contraintes sur l'ensemble de choix possibles qui sont les connaissances a priori sur le problème et
- la mesure sur le signal qui donne l'information sur une instance particulière de l'interprétation.

En général, ces deux types de connaissances ne sont pas toujours certaines. L'incertitude du premier type de connaissances est principalement due à l'observation des phénomènes à modéliser et à la formulation imcomplète des contraintes alors que l'incertitude du signal, notamment les phénomènes de superposition et de dispersion, rend l'interprétation des connaissances du deuxième type non-fiable. Puisque les élements de base d'une décision sont incertains, nous concluons qu'une décision ne peut pas être obtenue de façon absolument fiable.

Au moment de prendre une décision, si l'évidence est insuffisante un système peut se comporter de deux manières possibles:

- soit forcer le choix une seule solution, localement la meilleure. Le système risque de prendre une mauvaise décision qui n'est pas globalement cohérente avec l'interprétation correcte. Cette situation peut avoir des conséquences graves et irrécupérables ultérieurement.
- soit ne pas prendre la décision et attendre que l'évidence s'augmente. Il est certain que dans ce cas l'unité de décision ne contribue pas au processus d'interprétation, mais ceci est imposé par l'incertitude.

La technique consistant à reporter les décisions en présence d'incertitude revient à suspendre les décisions dans les zones où il y a plusieurs choix possibles et où la discrimination

est dans l'immédiat impossible ou peu sûre. On laisse l'incertitude du signal se propager dans les interprétations partielles et on l'enlève graduellement et non pas brutalement. Cette technique évite les décisions prématurées ou arbitraires aux bas niveaux de l'interprétation. Elle a été implantée par plusieurs auteurs [Scagliola 85,Masini 85].

4.6 Décisions non-exclusives

Nous avons indiqué dans le chapitre 4 que en présence d'incertitude, les décisions prises ne doivent pas être exclusives mutuellement. Nous donnons ici un exemple pour illustrer comment cette stratégie est utilisée dans notre analyseur syntaxique du signal. Il s'agit du traitement du résultat de la localisation d'un symbole terminal dans le signal, étant donnée la description des propriétés du symbole et la position approximative prédite par l'information obtenue depuis l'analyse du signal. Bien souvent, le niveau inférieur chargé de la localisation retourne la liste suivante de couples $C =< t, p >$ comme résultat:

$$(C_1, C_2) = (< t_1, p_1 >< t_2, p_2 >).$$

où t est la position du symbole et p le rapport de vraisemblance. Supposons que $p_1 > p_2$. Si l'on accepte que le symbole se trouve à t_1 parce que $p_1 = \max(p_1, p_2)$, et si on abondonne l'autre proposition C_2, dans le cas où l'interprétation construite sur C_2 est l'interprétation correcte globalement, l'erreur due au mauvais choix est irrécupérable. Notre méthode consiste à prendre des décisions qui ne s'excluent pas mutuellement et si nécessaire à développer simultanément les deux branches. Le couple qui ne correspond pas à l'interprétation correcte sera enlevé plus tard lorsqu'il y aura suffisamment d'évidence. L'algorithme suivant schématise cette idée:

```
Décision-deux-couple(C1, C2) → listedecouple < t, p >
    Si Très(EstGrand(p(C1)/p(C2)))
        Alors (C1)
        Sinon
            Si Très(EstGrand(p(C2)/p(C1)))
                Alors (C2)
                Sinon (C1, C2)
            Finsi
    Finsi
```

4.7 Qualité d'interprétation

Pour contrôler correctement et efficacement le processus d'interprétation, il est naturel et indispensable de mesurer qualitativement les interprétations partielles, en fonction de l'incertitude [Erman 80,Woods 82]. Nous avons parlé d'"évidence", "certitude", "sûreté" et "confiance" pour traduire le fait que le signal et les primitives à interpréter ne sont pas identifiés de façon unique. Souvent une même unité primitive est déclarée appartenir à plusieurs classes, avec des valeurs de fonction d'appartenance associées. Le résultat d'interprétation possède en conséquence aussi la propriété d'être incertain. Dans l'analyseur syntaxique,

cette incertitude est représentée par la qualité de chaque îlot de confiance qui constitue l'interprétation.

Comme l'incertitude est due à la conversion signal-symbole, nous mesurons la qualité d'une interprétation partielle directement en fonction des valeurs des fonctions qui déterminent le dégré d'appartenance du signal aux symboles primitifs. Une définition concrète de la qualité est donnée au paragraphe 5.8.

Il est très coûteux en temps de calcul de garantir la meilleure solution pour l'interprétation, car cela revient à parcourir exhaustivement l'espace de solutions. Comme mentionné précédemment, notre constructeur de solution cherche à garantir les meilleurs îlots, et donc si le rayon de recherche est suffisamment grand les meilleures interprétations, qui sont des combinaisons de ces îlots, sont aussi garanties.

5 Algorithmes de construction de structure

5.1 Présentation générale

L'ensemble des connaissances sur la structure à interpréter est donné sous forme d'une grammaire de type hors contexte décrit par des règles. L'indéterminisme est entièrement autorisé. Il est bien entendu qu'une analyse déterministe serait plus efficace mais l'écriture des règles pourrait devenir un problème délicat et la maintenance du système serait plus difficile [Francopoulo 86,Richie 83].

Un îlot est une interprétation partielle; il représente à la fois une partie identifiée du signal et la structure syntaxique construite sur celle-ci. L'analyse peut débuter sur plusieurs parties du signal et chaque partie peut avoir des interprétations concurrentes qui sont toutes correctes localement.

Le problème d'appariement des formes, consistant à déterminer la correlation entre une hypothèse générée et le signal, sera présenté en partie V, dans le cadre de l'interprétation du signal de la parole.

5.2 Représentation des objets

Le constructeur accepte des règles exprimées sous forme de grammaire BNF, étendue par des possibilités d'adjonction à chaque règle d'actions pouvant être exécutées lors le déclenchement de la règle. Un exemple de ces actions est la vérification de cohérence sémantique entre le sujet groupe nominal et l'objet groupe nominal. La récursivité est autorisée, ce qui est utile en parole où des phrases, des modificateurs, etc, ont souvent des structure récursives. Des heuristiques sont utilisées pour limiter la profondeur de recherche.

Dans les algorithmes qui suivent la fonction d'accès à la propriété P d'un objet X s'écrit simplement comme $P(X)$. Le constructeur de structures manipule deux types d'objets composés: les nœuds et les îlots.

Un objet du type nœud comprend les attributs suivants:

- *Nom:* le symbole terminal ou nonterminal des règles associé au nœud.

- *Fils:* la liste des fils qui sont eux mêmes des nœuds.
- *Père:* le père du nœud. Nous avons utilisé des pointeurs dans les deux sens, ascendant et descendant, afin de faciliter le parcours des arbres.
- *RègleAvant:* le numéro de la règle qui a conduit à l'instanciation du nœud. Cette propriété est utilisée pour la fusion des arbres et permet de suivre le raisonnement.
- *Lexc:* le symbole du type lexique liant un nœud terminal au niveau inférieur.

Un objet du type îlot est composé de propriétés suivantes:

- *Racine:* la racine de l'arbre d'interprétation de l'îlot.
- *TfGauche, TfDroite:* les deux nœuds terminaux identifiés aux extrêmités gauche et droite, fournissant l'information sur l'emplacement de l'îlot.
- *LexcListe:* la liste des symboles terminaux identifiés de l'îlot. Cette liste est utilisée pour évaluer la qualité de l'interprétation de l'îlot.
- *ListeSuivante:* la liste d'îlots qui peuvent éventuellement être fusionnés avec l'îlot.
- *Qualité:* la qualité d'interprétation de l'îlot. Cette qualité est utilisée dans le prise de décision au cours de la recherche de solutions en faisceau.

5.3 Inférence en chaînage avant

L'inférence en chaînage avant consiste à effectuer une extension des nœuds à partir de la racine d'un îlot. Ce traitement se fait pour chaque îlot dans chaque liste d'interprétations partielles et concurrentes. Pour un îlot, une règle applicable en chaînage avant peut donner naissance à un nouvel îlot. S'il y a plusieurs règles de tel type alors il y aura plusieurs interprétations possibles, qui sont toutes localement correctes. A la fin du parcours de l'ensemble d'îlots ListeIlots, tous les îlots pouvant être développés sont dans l'ensemble de nouvels îlots NouvListeIlots. La différence de ces deux ensembles est supprimée afin de récupérer l'espace de mémoire.

```
ChaînageAvant
    PourChaque ListeIlots Dans Conférence Faire
        NouvListeIlot = VIDE;
        PourChaque îlot Dans ListeIlots Faire
            ActRègles = AccèsRègle(NomRacine(îlot), DROITE);
            Si ActRègles
            Alors
                PourChaque r Dans ActRègles Faire
                    NouvIlot = ChaîneAvantExtension(îlot,r);
                    Si NouvIlot
                    Alors
                        NouvListeIlots = AjouteTête(NouvListeIlots,NouvIlot);
                        NoterCompleteIdent(Racine(NouvIlot));
                    FinSi;
                FinPour;
            Sinon NouvListeIlots = AjouteTête(NouvListeIlots,îlot);
            FinSi;
        FinPour;
        SupprmerIlots(DiffEns(ListeIlots,NouvListeIlots));
        ListeIlots = NouvListeIlots;
    FinPour;
```

L'extension des nœuds donnés par la règle "règle" sur l'îlot "îlot" est réalisée par la fonction ChaîneAvantExtension(îlot,règle). Cette fonction copie d'abord l'îlot puis instancie la règle en créant tous les nœuds correspondant aux symboles dans la règle. Le résultat de l'extension est le nouvel îlot.

```
ChaîneAvantExtension(îlot,r) → îlot
    NouvIlot = CopieUnIlot(îlot);
    NouvRac = Générer(NŒUD, (nom=Gauche(r),RègleAvant=Numéro(r)));
    AncienRac = Racine(îlot);
    Fils(NouvRac) =
        PourChaque Symbole Dans Droite(r) Faire
            Si NomRacine(îlot) ≠ symbole
            Alors Générer(NŒUD,(nom=Symbole,Père=NouvRac))
            Sinon Père(AncienRac) = NouvRac;
            FinSi;
        FinPour;
    Racine(îlot) = NouvRac;
    NouvIlot;
```

5.4 Inférence en chaînage arrière

L'inférence en chaînage arrière assure l'installation d'un arbre sur une feuille non-terminale d'un îlot. Par rapport à la partie déjà identifiée de l'îlot, la feuille peut être soit le premier

nœud à gauche soit le premier nœud à droite. Comme dans l'inférence en chaînage avant, ce traitement est appliqué à tous les îlots dans toutes les listes d'interprétation concurrentes. Si plusieurs règles sont applicables pour un nœud, tous les arbres correspondants aux règles sont engendrés.

```
Direction = la direction ([GAUCHE,DROITE]) de l'extension
ChaînageArrière(Dirrection)
    PourChaque ListeIlots Dans Conférence Faire
        NouvListeIlot = VIDE;
        PourChaque îlot Dans ListeIlots Faire
            ListeIlotsEtendus = ChaîneArrièreExtension(Direction,îlot);
            NouvListeIlots =
                Si ListeIlotsEtendus
                Alors Concate(NouvListeIlots,NouvIlot)
                Sinon AjouteTête(îlot,NouvListeIlots);
                FinSi;
        FinPour;
        SupprmerIlots(DiffEns(ListeIlots,NouvListeIlots));
        ListeIlots = NouvListeIlots;
    FinPour;
```

La fonction ChaîneArrièreExtension se réalise en deux phases:

- La première consiste, étant donné la racine de l'îlot et la direction, à trouver le nœud non-terminal et nonidentifié dont le voisin (gauche ou droite, selon direction) est identifié.
- La seconde phase consiste, à partir du nœud localisé dans la première phase, à trouver récursivement toutes les règles applicables jusqu'à ce qu'un nœud terminal soit rencontré. Le frère est donné par la fonction PremierNonIdent.

Pendant la deuxième phase, chaque règle dans la chaque liste de règles trouvées est installée sur le nœud de départ. Cette installation est réalisée par la fonction ChaînageArrExtension qui copie d'abord l'îlot puis, pour chaque règle dans la liste de règles conduisant du nœud non-identifié du départ jusqu'au nœud terminal, construit l'arbre d'instanciation.

```
dir = la direction ([GAUCHE,DROITE]) de l'extension
îlot = l'îlot d'interprétation d'origine à étendre
ChaîneArrièreExtension(dir,îlot) → liste d'îlots
    Nœud = NœudExter(Racine(îlot),dir);
    Si Nœud
        Alors DescendAuTerminal(dir,Nœud,Nom(nœud),îlot,VIDE,VIDE)
        Sinon VIDE;
    FinSi;
```

```
NœudExter(Nœud,dir) → nœud
    Si Identifié(Nœud) Alors VIDE
    Sinon
        Si Null(Fils(Nœud)) Alors Nœud
        Sinon
        FilsOrdonnés =
            Si dir=GAUCHE Alors Fils(Nœud)
            Sinon Renverse(Fils(Nœud));
            FinSi;
        NœudNonIdent = PremierNonIdent(FilsOrdonnés);
        Si NœudNonIdent
        Alors NœudExter(NœudNonIdent,dir)
        Sinon
            Si NœudIdentifié(Premier(FilsOrdonnés))
            Alors VIDE
            Sinon NœudExter(Dernier(FilsOrdonnés,dir);
            FinSi;
        FinSi;
    FinSi;
```

```
dir = la direction ([GAUCHE,DROITE]) de l'extension
nœud = le nœud départ de la phase descendnte
nom = le nom du nœud
îlot = l'îlot d'interprétation d'origine à étendre
Nîlots = la liste de nouveaux îlots générés
SéqRègles = la séquence de règles appliquées depuis le premier appel
PROFONDEUR = la limite du nombre de règles
```

```
DescendAuTerminal(dir,nœud,nom,îlot,Nîlots,SéqRègle) → liste d'îlots
    Si Terminal(nom)
    Alors AjouteTête
        (ChaînageArrExtension(îlot,nœud,dir,Renverse(SéqRègles)),Nîlots)
    Sinon       Si Longueur(Nîlots) < PROFONDEUR
        Alors           PourChaque règle Dans AccèsRègle(nom,GAUCHE) Faire
                Nîlots = DescendAuTerminal
                    (dir,nœud,ContreDirElt(dir,droite(règle)),
                    îlot,Nîlots,AjouteTête(règle),SéqRègle);
            FinPour;
        FinSi;
        Nîlots;
    FinSi;
```

```
ChaînageArrExtension(îlot,nœud,dir,SéqRègles) → liste d'îlots
    NouvIlot = CopieUnIlot(ilot);
    InstallerArr(NouvIlot,NouvNœud(nœud),SéqRègles,dir);
    NouvIlot;
```


La fonction InstallerArr génère l'arbre correspondant à la liste de règles donnée en argument. Au dernier nœud, forcément un nœud terminal, une fonction DéplacerMarques est appelée pour déplacer la marque de zone à identifier vers une nouvelle position.

```
InstallerArr(îlot,nœud,ListeRègles,dir) → VIDE
    Si Null(ListeRègle)
    Alors DéplacerMarques(dir,nœud,îlot)
    Sinon UneRègle = Premier(ListeRègle);
        Fils(nœud)=             PourChaque sym Dans Droite(UneRègle) Faire
                Générer(NŒUD,(nom=sym,Père=nœud));
            FinPour;
        RègleAvant((nœud) = Numéro(UneRègle);
        InstallerArr
            (îlot,ContreDirElt(dir,Fils(nœud)),Reste(ListeRègles),dir);
    FinSi;
```

```
dir = la direction ([GAUCHE,DROITE]) de l'extension
liste = une liste d'éléments
```

```
ContreDirElt(dir,liste) → élément
    Si dir = GAUCHE
        Alors Dernier(liste)
        Sinon Premier(liste);
    FinSi;
```

La recherche de solution ayant lieu en parallèle, lors de l'interprétation le mécanisme d'inférence demande fréquemment le copie d'un îlot. Afin d'éviter de recopier inutilement la partie identifiée d'un îlot, les nœuds identifiés de chaque arbre d'interprétation sont partagés au maximum. Cette technique économise non seulement de la place mémoire mais réduit aussi considérablement le temps d'exécution. Nous décrivons l'algorithme de copie qui ne copie que des nœuds nonidentifiés tout en conservant la relation père et fils de chaque nœud.

```
Nœud = un nœud de l'arbre
NouvPère = le nouveau père du nœud
```

```
CopieNonIdentNœud(Nœud,NouvPer'e) → nœud
    Si Identifié(Nœud) Alors Nœud
    Sinon NouvRac = CopieRéel(Nœud);
        NouvNœud(Nœud) = NouvRac;
        Père(NouvRac) = NouvPère;
        Fils(NouvRac) =
            PourChaque fils Dans Fils(Nœud) Faire
                CopieNonIdentNœud(fils,NouvRac);
            FinPour;
        NouvNœud;
    FinSi;
```

5.5 Fusion des arbres d'interprétation partielle

Comme indiqué précédemment, pour garantir la meilleure interprétation en présence d'incertitude, les arbres d'interprétation possibles sont développés de façon parallèle à partir de plusieurs îlots dans le signal. Pour ne pas conserver la même structure dans deux ou plusieurs arbres différents, il faut fusionner des arbres qui présentent une structure commune et instanciée sur la même zone du signal. La difficulté de la fusion se manifeste dans la détection, parmi des nombreuses possibilités de combinaisons, des arbres à fusionner. Par ailleurs, la présence de la relation éventuelle d'inclusion ou d'égalité entre deux arbres n'est pas suffisante pour fusionner les deux arbres, car il est nécessaire que la situation spatiale ou temporelle indique qu'ils peuvent être fusionnés. Cette difficulté est augmentée par le fait que, pour prendre une décision de fusion, il est indispensable de déterminer correctement si les zones identifiées sous deux arbres d'interprétation sont superposées. A titre d'exemple, deux configurations d'îlots où des fusions sont nécessaires sont présentés dans 6.1. Nous présentons dans ce paragraphe l'ensemble des algorithmes de fusion.

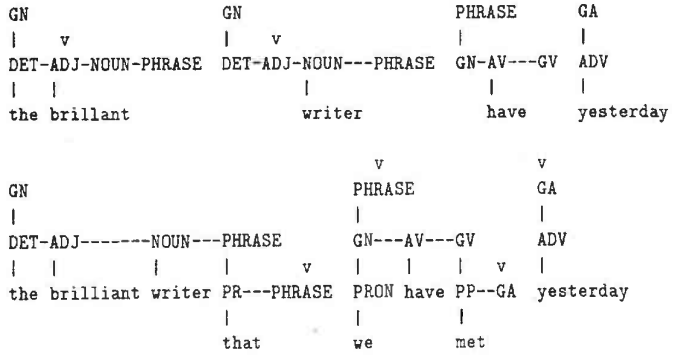


Figure 6.1: Quelques configurations des îlots nécessitant des fusions (indiquées par "v")

Les arbres d'interprétation sont regroupés en listes d'îlots, chacune contenant les îlots construits à partir de la même zone initiale. Par conséquent, il est impossible de fusionner des îlots à l'intérieur d'une telle liste. En revanche, il est possible qu'un îlot puisse fusionner avec les îlots dans une liste correspondant à une zone de signal voisine. La fusion doit donc se faire entre chaque paire d'îlots pris dans deux listes voisines. La fonction FusionToutListeIlots ci-dessous prend l'ensemble des listes d'îlots et retourne l'ensemble des listes d'îlots fusionnés.


```
EnsListeIlots = l'ensemble de listes d'îlots à fusionner
FusionToutListeIlots(EnsListeIlots) → ensemble de listes d'îlots
    Si Null(EnsListeIlots)
    Alors VIDE
    Sinon
        ListeIlotsSuivant = FusionToutListeIlots(Reste(EnsListeIlots));
        NouvListe = FusionListeIlots(Premier(EnsListeIlots));
        AjouteTête(NouvListe,ListeIlotsSuivant);
    FinSi;
```

On essaie de fusionner chaque îlot dans ListeIlots avec les îlots de la liste suivante (obtenue par la fonction ListeSuivante). Si la liste suivante est vide, l'îlot atteint déjà l'extrêmité du signal et il est tout simplement conservé dans la nouvelle liste d'îlots.

```
FusionListeIlots(ListeIlots) → liste d'îlots
    NouvListe = VIDE;
    PourChaque îlot Dans ListeIlots Faire
        NouvListe =
            Si Null(ListeSuivante(îlot))
            Alors AjouteTête(îlot,NouvListe)
            Sinon TesterListeSuivante(îlot,ListeSuivante(îlot),NouvListe);
            FinSi;
    FinPour;
    NouvListe;
```

Deux rôles sont assurés par la fonction TesterListeSuivante.

- Le premier est de fusionner les éléments dans la liste passée en argument avec l'îlot passé en argument.

- Le second est de tester si l'îlot n'est plus fusionnable. Dans le cas de confirmation, l'îlot est définitivement supprimé. Ce test est réalisé par la fonction TousSuperposéVoisins.

```
TesterListeSuivante(îlot1,liste,IlotsTraités) → liste d'îlots
    NouvListeSuivante = VIDE;
    Concate(
        PourChaque îlot2 Dans Liste Faire
            IlotFusionné = Fusion2Ilots(îlot1,îlot2);
            Si IlotFusionné
            Alors
                AjouteTête(îlot2,NouvListeSuivante);
                IlotFusionné;
            FinSi;
        FinPour;,
        Si Non(TousSuperposéVoisins(îlot,liste)) et NouvListeSuivante ≠ liste
        Alors AjouteTête(îlot1,IlotsTraités)
        Sinon          Supprimer(îlot1);
            IlotsTraités;
        FinSi;);
```

La fonction Fusion2Ilots essaie de fusionner les deux îlots passés en argument. Elle retourne le nouvel îlot dans lequel les deux îlots donnés sont unifiés ou VIDE si la fusion n'est pas possible. Soient AD et AG respectivement l'arbre droit et l'arbre gauche que l'on teste. Dans cette fonction, nous distinguons trois cas entre les arbres AG et AD:

- l'arbre gauche AG peut constituer un sous-arbre de l'arbre droit AD,
- l'arbre droite AD peut constituer un sous-arbre de l'arbre gauche AG ou
- il peut être impossible de fusionner les deux.

Deux conditions sont nécessaires pour que deux îlots soient fusionnables:

- les zones identifiées sous les deux îlots ne doivent pas être superposées et
- il ne doit pas exister de zone nonidentifiée entre les deux zones.

Ce test est réalisé par la fonction Voisin. Si un test de sous-arbre est réussi, la fonction Unifflots est appelée pour effectuer l'opération d'installation du sous-arbre dans l'arbre maître.


```
AG = l'arbre à gauche
AD = l'arbre à droite
```

```
Fusion2Ilots(AG,AD) → îlot
    Si Voisin(AG,AG)
    Alors
        NdMaître = SousArbre(Racine(AG),NœudExter(Racine(AD),GAUCHE));
        Si NdMaître
        Alors Unifflots(AG,AD,NdMaître,DROITE)
        Sinon
            NdMaître = SousArbre(Racine(AD),NœudExter(Racine(AG),DROITE));
            Si NdMaître
            Alors Unifflots(AD,AG,NdMaître,GAUCHE)
            Sinon VIDE;
            FinSi;
        FinSi;
    FinSi;
```

La fonction SousArbre(A1,A2) teste si A1 peut être un sous arbre de A2. Le résultat est soit le nœud qui peut être le père de A1 soit VIDE au cas où un tel nœud n'existe pas.

```
SousArbre(A1,A2) → nœud
    Si A1 et A2
    Alors
        Si Nom(A1)=Nom(A2) et
            (Non(Fils(A1)etFils(A2)) ou RègleAvant(A1)=RègleAvant(A2))
        Alors A2
        Sinon SousArbre(A1,Père(A2));
        FinSi;
    Sinon VIDE;
    FinSi;
```

5.6 Contraintes positionnelles

L'objectif de l'introduction des contraintes positionnelles est d'éviter à l'analyseur de construire des structures partielles qui seront finalement abandonnées à cause de l'incohérence de leur position temporelle dans le signal réel. Ces contraintes sont appliquées sur le nombre approximatif de terminaux qui doivent exister à gauche et à droite d'un symbole terminal spécifique. Ce nombre est obtenu lors d'une phase de précompilation des règles et est stocké comme une des propriétés de chaque règle. Expérimentalement, sans aucune dégradation de qualité de l'interprétation, le gain de temps de calcul est de l'ordre de 50%. Nous présentons la fonction de vérification de ces contraintes "PositionContrainte".

```
dir = la direction dans la quelle on vérifie la contrainte
PositionContrainte(dir)
    PourChaque ListeIlots Dans Conférence Faire
        NouvListeIlot = VIDE;
        PourChaque ilot Dans ListeIlots Faire
            Règle = AvRègle(RègleAvant(Racine(ilot)));
            Si
                Si dir = GAUCHE
                Alors TestPos(dir, NumFait(TfGauche(ilot)), MinMaxTf(Règle,GAUCHE));
                Sinon TestPos(dir, NumFait(TfDroite(ilot)), MinMaxTf(Règle,DROITE));
                Finsi
            Alors NouvListeIlots = AjouteTête(NouvListeIlots,ilot);
            Finsi;
        FinPour;
        SupprimerIlots(DiffEns(ListeIlots,NouvListeIlots));
        ListeIlots = NouvListeIlots;
    FinPour;
```

La fonction NumFait(fait) retourne le numéro du fait dans le signal et la fonction MinMaxTf(règle, direction) donne le couple (minimum,maximum) attaché au nombre de faits voisins autorisés sous le symbole de la partie gauche de la règle, dans la direction indiquée. Nous signalons que l'erreur de comptage des nombres des faits, inévitable en présence d'incertitude, ne peut pas perturber le fonctionnement de la fonction grâce à TOLNBERR. La fonction TestPos est définie de la façon suivante.

```
TOLNBERR = le nombre d'erreurs maximum au cours du comptage des faits terminaux
MAX-NB-TF = le nombre de faits terminaux dans le signal
dir = la direction de test
numfait = le numéro du faits dans le signal
MinMaxTf = couple des nombres min et max de fait terminaux sous une règle
TestPos(dir, numfait, MinMaxTf)
    N =
        Si dir = DROITE
        Alors MAX-NB-TF - numfait + 1
        Sinon numfait;
        Finsi;
    (N ≤ max(MinMaxTf) + TOLNBERR) et (N ≥ min(MinMaxTf) - TOLNBERR);
```

5.7 Emission des hypothèses

Cette fonction collecte toutes les instances des symboles terminaux qui sont générées comme hypothèses et qui n'ont pas été vérifiées.


```
HypothèsePrepa → liste de nœuds terminaux
    TFait = VIDE;
    PourChaque ListeIlots Dans Conférence Faire
        PourChaque îlot Dans ListeIlots Faire
            Si NœudIdentifié(TfGauche(îlot))
            Alors TFait = AjouteEnTete(TfGauche(îlot), TFait);
            Finsi;
            Si NœudIdentifié(TfDroite(îlot))
            Alors TFait = AjouteEnTete(TfDroite(îlot), TFait);
            Finsi;
        FinPour;
    FinPour;
    TFait;
```

5.8 Evaluation de qualité

La qualité d'une interprétation mesure le degré de cohérence entre celle-ci et le signal observé. Du fait de l'incertitude, il est indispensable de mesurer la qualité des interprétations partielles pour contrôler le processus d'interprétation et effectuer des stratégies de recherche.

Dans un but de fiabilité, la mesure de qualité est calculée directement à partir des sources d'incertitude: les coefficients de certitude des terminaux. La qualité d'un îlot est calculée en fonction de la reconnaissance de ces symboles terminaux mesurables:

$$Q(i) = \frac{1}{t_b(i) - t_a(i)} \sum_{\forall m \in W_T(i)} \int_{t_1(m)}^{t_2(m)} \mu_m(t) dt$$

où $t_a(i)$ et $t_b(i)$ sont respectivement le début et la fin de la zone du signal sur lequel est construit l'îlot i, $t_1(m)$ et $t_2(m)$ sont respectivement le début et la fin du mot m, et $\mu_m(t)$ est la fonction d'appartenance du signal de la parole à l'instant t à la classe m, et $W_T(i)$ est l'ensemble des terminaux de l'îlot i. On approche $\mu_m(t)$ par la qualité du mot m $Q_{mot}(m)$ définie en 7.5.

5.9 Détermination des faisceaux

Le nombre maximum d'interprétations partielles en parallèle conservées après une élimination des îlots à faible qualité constitue le *diamètre* de la recherche en faisceau. En présence d'incertitude, lorsque le nombre de symboles terminaux d'un îlot est petit, il n'y a que peu d'information sur des zones voisines. On n'est à ce moment que peu sûr que l'îlot en question puisse devenir une interprétation correcte. Donc la fiabilité de l'îlot est faible. Autrement dit la cohérence entre la qualité de l'îlot et la possibilité qu'il soit correct globalement est petite. On dit que l'îlot est *fragile*. En revanche, lorsqu'un îlot contient un nombre important de symboles terminaux, il est relativement *rigide* et il est probable qu'une de ses extensions devienne une interpretation correcte.

Par conséquent, à chaque étape d'analyse, le nombre d'îlots retenus doit être défini en fonction du nombre de symboles terminaux (N.S.T.) dans l'arbre syntaxique. Nous utilisons

notamment deux fonctions liées à ce nombre:

La première fonction donne le diamètre. Le diamètre est autant plus grand que les îlots sont fragiles. La figure 6.2 montre un exemple, où la relation est obtenue expérimentalement.

diamètre	0	50	30	15	12	10	8	8	7	7	7	6	6	6
N.S.T.	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Figure 6.2: Diamètre de recherche en fonction du nombre de symboles terminaux (obtenu dans d'une expérience d'interprétation de parole)

La seconde donne la qualité minimale qu'un îlot en validation doit avoir. La figure 6.3 montre un exemple tiré d'une application d'interprétation de parole.

N.S.T.	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Qualité	.0	.29	.22	.23	.23	.23	.235	.235	.24	.24	.25	.26	.26	.26

Figure 6.3: Qualité minimale qu'un îlot doit avoir pour être admis dans le faisceau (obtenue dans d'une expérience d'interprétation de parole)

6 Contrôle du processus d'analyse

6.1 Introduction

Nous avons présenté les différentes composantes de l'analyseur. Nous allons expliquer comment ces composantes sont activées de façon coopérative dans le processus d'interprétation. Le schéma d'inférence mixte, (ou analyse descendante et ascendante) est simple mais le contrôle d'un tel processus est compliqué. Le concept est clair mais la construction d'un tel système demande un effort important [Nagao 84].

L'objectif du contrôle est de faire coopérer les composantes afin de parvenir à la structure correcte de façon efficace. L'idée essentielle de ce contrôle est d'utiliser explicitement et au mieux la dépendance connue entre les composantes sur les relations de production et de consommation d'information. Les contraintes de causalité sur la circulation de l'information déterminent en grande partie l'ordre d'exécution. S'il y a encore plusieurs composantes exécutables à un instant donné, une priorité simple est utilisée. Le contrôle ainsi réalisé est rigide, lisible, facile à construire et moins empirique, par rapport au contrôle basé uniquement sur la priorité des composantes tel que celui utilisé dans Hearsay-II [Erman 80].

6.2 Règles de dépendance

La dépendance des composantes du constructeur est représentée sous forme de règles, sous forme de liste de:

< (nom de la composante), (condition d'activation)>

qui signifie qu'un composante est exécutable si la condition associée est vérifiée. La partie condition est une expression logique. Un interprète teste successivement les conditions d'activation de ces composantes et exécute la composante dont la condition est vérifiée. La condition d'activation représente la situation actuelle de l'interprétation. Comme pour tous les systèmes à base de règles, l'avantage est que pour assurer le fonctionnement il suffit de vérifier la correction locale de chaque dépendance. En modifiant les règles de contrôle, il est facile de changer et de tester une stratégie de contrôle. Nous donnons un ensemble des règles dans la figure 6.4. A titre d'exemple, la deuxième règle définit la condition pour déclencher la composante *backward-chain-left*:

backward-chain-left est exécutable
SI la composante précédente est *forward-chain-main*
OU ALORS la composante précédente est *unify*
ET le sens du dernier chaînage-arrière est gauche
OU ALORS tous les îlots sont déjà fusionnés
ET la composante précédente est *predict-ph*
ET le sens du dernier chaînage-arrière est gauche

7 Expériences de fonctionnement

7.1 Application des règles

Dans le tableau 6.1 nous donnons un exemple de la répartition de l'exécution des actions de base de notre constructeur de structure syntaxique, pris dans un processus réel d'interprétation de la parole continue.

7.2 Processus d'interprétation

Pour présenter le déroulement de la construction de la structure syntaxique, nous utilisons l'algorithme "ImpriArbre" qui imprime un arbre sous forme de texte. Cet algorithme parcourt l'arbre en profondeur d'abord et affiche les symboles des nœuds au cours du parcours.

```
(setq CTRL-RULES '(
      (forward-chain-main;
                (or (and (precessor 'position-constrain)
                         (last-backward 'R))
                    (precessor ()))))

      (backward-chain-left;
          (or (precessor 'forward-chain-main)
              (and (precessor 'unify)
                   (last-backward 'L))
              (and (all-unified)
                   (precessor 'predict-ph)
                   (last-backward 'L))))

      (backward-chain-right;
          (or (and (precessor 'position-constrain)
                   (last-backward 'L))
              (and (precessor 'unify)
                   (last-backward 'R))
              (and (all-unified)
                   (precessor 'predict-ph)
                   (last-backward 'R))))

      (position-constrain;
          (and (or (precessor 'backward-chain-left)
                   (precessor 'backward-chain-right))
               (not (last-success))))

      (hypoth-prepare;
          (and (or (precessor 'backward-chain-left)
                   (precessor 'backward-chain-right))
               (last-success)))

      (qualify;
          (precessor 'hypoth-prepare))

      (beam-search;
          (precessor 'qualify))

      (predict-ph;
          (precessor 'beam-search))

      (unify; (or (precessor 'forward-chain-main)
          (and (precessor 'predict-ph)
               (not (all-unified))))
))
```

Figure 6.4: Règles de contrôle

Nom de fonction	Nombre d'activations
forward-chain-main	6
position-constrain	10
backward-chain-right	24
predict-ph	23
beam-search	23
qualify	23
hypoth-prepare	23
backward-chain-left	9

Table 6.1: les fonctions de construction de structure et leur nombre d'activation pendant une interprétation complète

nœud = le nœud de la racine de l'arbre
x, y = la position où la racine doit être imprimée
last = le fait que le nœud est le dernier fils de son père
TousLesFils = la liste des fils du nœud
PosSuivante = la position suivante s'il n'y pas de fils
Sx = la position suivante pour imprimer un fils
DéplEnX = l'espace en x entre deux noms des nœuds, constante
DéplEnY = l'espace en x entre deux noms des nœuds, constante

```
ImpriArbre(nœud, x, y, last) → déplacement maximum en x de l'arbre
    Si nœud
    Alors
        TousLesFils = Fils(nœud);
        PosSuivante = x + DéplEnX + Longueur(Nom(nœud));
        Sx = Si TousLesFils Alors x Sinon PosSuivante;
        PourChaque fils dans TousLesFils Faire
            Sx = ImpriArbre(fils, Sx, y + DélpEnY, Dernier(fils,TousLesFils));
        FinPour;
        Afficher(nœud, x, y, Max(Sx, PosSuivante), last);
        Max(Sx, PosSuivante);
    Finsi
```

La fonction "Dernier(nœud, listeNœuds)" teste si "nœud" est le dernier élément dans "listNœuds" alors que la fonction "Afficher" imprime un nœud sur le support d'impression.

L'exemple suivant illustre le fonctionnement de l'analyseur syntaxique. L'ensemble des règles prises en exemple représente la grammaire d'un sous-ensemble de langage naturel est est représenté en *LISP*. N est le numéro de la règle. L et R sont respectivement la partie gauche et droite de la règle. Pour simplifier, nous n'avons pas listé les actions associées à chaque règle.

```
(setq #:GRAMMAR:winograd '(
 ((N . R1)  (L . SENTENCE) (R On PHRASE Off))
 ((N . R2)  (L . PHRASE)   (R GN GV))
 ((N . R3)  (L . PHRASE)   (R PR PHRASE))
 ((N . R4)  (L . PHRASE)   (R GN AV GV))
 ((N . R5)  (L . GN)       (R PRON))
 ((N . R6)  (L . GN)       (R DET ADJ NOUN PHRASE))
 ((N . R7)  (L . PRON)     (R he))
 ((N . R8)  (L . PRON)     (R she))
 ((N . R9)  (L . PRON)     (R we))
 ((N . R10) (L . GV)       (R VB GA GN))
 ((N . R11) (L . VB)       (R appreciate))
 ((N . R12) (L . GA)       (R ADV))
 ((N . R13) (L . GV)       (R PP GA))
 ((N . R14) (L . ADV)      (R yesterday))
 ((N . R15) (L . ADV)      (R much))
 ((N . R16) (L . DET)      (R the))
 ((N . R17) (L . ADJ)      (R brilliant))
 ((N . R18) (L . NOUN)     (R writer))
 ((N . R19) (L . PR)       (R that))
 ((N . R20) (L . AV)       (R have))
 ((N . R21) (L . PP)       (R met))
))
```

```
Ilot No-100 Ilot No-101 Ilot No-102 Ilot No-103

appreciate  writer      have        yesterday
```

Figure 6.5: Les propositions initiales émises par le niveau inférieur, le niveau lexical, sont constitués de quatre mots, chaque mot forme respectivement un îlot de confiance dans l'ensemble des interprétations partielles initiales du niveau syntaxique.

```
Ilot No-104 Ilot No-105 Ilot No-106 Ilot No-107

VB          NOUN        AV          ADV
|           |           |           |
appreciate  writer      have        yesterday
```

Figure 6.6: Une inférence chaînage en avant produit les nouveaux îlots. Aucun nœud n'ayant de frères, il est impossible d'appliquer l'inférence en chaînage arrière.

La phrase de test est *We appreciate much the brilliant writer that we have met yesterday.* Nous illustrons le processus de construction par une suite de figures commentées.


```
Ilot No-108       Ilot No-109            Ilot No-110 Ilot No-111

GV                GN                     PHRASE      GA
|                 |                      |           |
VB---------GA-GN  DET-ADJ-NOUN---PHRASE  GN-AV---GV  ADV
|                       |                  |         |
appreciate              writer             have      yesterday
                                                      .
```

Figure 6.7: Après une autre phase de chaînage avant, les îlots 108, 109 et 110 acquièrent des frères. Il est possible d'hypothétiser les symboles terminaux de ces îlots.

```
Ilot No-108       Ilot No-112                Ilot No-115    Ilot No-111

GV                GN                         PHRASE         GA
|                 |                          |              |
VB---------GA-GN  DET-ADJ-------NOUN---PHRASE  GN---AV---GV  ADV
|                     |         |              |    |        |
appreciate            brilliant writer         PRON have      yesterday
                                               |
                                               we
```

Figure 6.8: L'inférence en chaînage arrière à gauche crée pour les îlots 108, 109 et 110 respectivement 1, 1 et 3 interprétations partielles. Ces ambiguïtés sont dues à l'indéterminisme des règles et doivent être levées en utilisant des évidences de bas niveau. L'analyseur syntaxique transmet ensuite les terminaux nonidentifiés de ces interprétations comme hypothèses au niveau inférieur afin de supprimer les interprétations dont la qualité est trop faible. L'émission et la qualification sont réalisées par deux fonctions du niveau. Il n'y a que l'interprétation ci-dessus qui a été retenue.

```
Ilot No-117         Ilot No-122                    Ilot No-128       Ilot No-111

GV                  GN                             PHRASE            GA
|                   |                              |                 |
VB---------GA---GN  DET-ADJ-------NOUN---PHRASE     GN---AV---GV      ADV
|          |            |         |      |          |    |    |       |
appreciate ADV          brilliant writer PR---PHRASE PRON have PP--GA  yesterday
           |                             |          |         |
           much                          that       we        met
```

Figure 6.9: L'analyseur active ensuite une inférence en chaînage arrière à droite. Pour les îlots 108, 112 et 115, respectivement 2, 9 et 2 interprétations partielles sont générées. Après le calcul de la qualité, une seule configuration (ci-dessus) est conservée. Il existe maintenant des îlots unifiables, ce sont d'une part 122 et 128 et d'autre part 128 et 111.

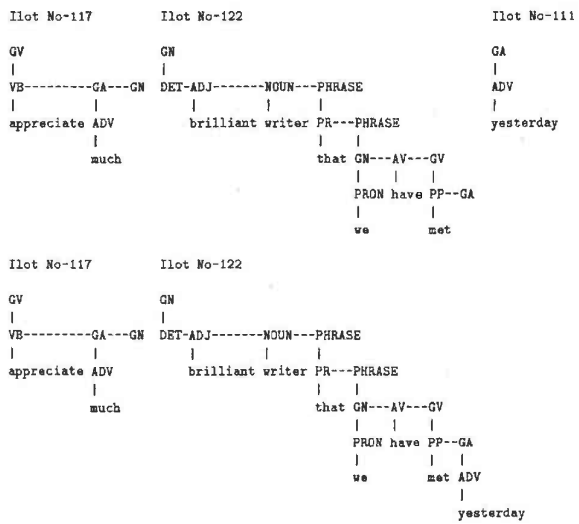


Figure 6.10: La fonction "unifieur" a détecté les îlots unifiables et puis les unifie, comme indiquent les deux arbres. A cause des tailles des arbres partiels, dans les discussions suivantes seuls les îlots qualifiés seront présentés.

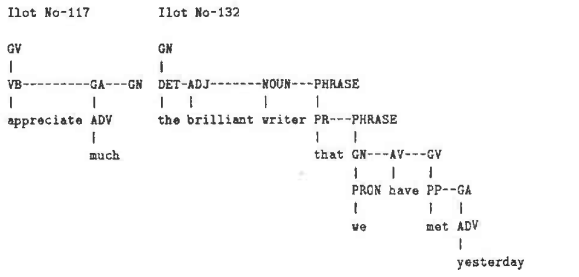


Figure 6.11: L'inférence en chaînage arrière à gauche a installé le terminal "the" sur l'îlot 122, donnant l'îlot 132.

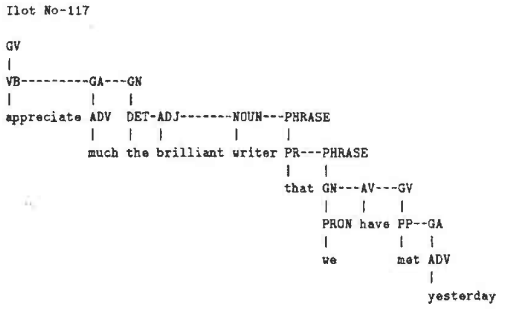


Figure 6.12: Les deux îlots dans la figure précédente sont unifiés.

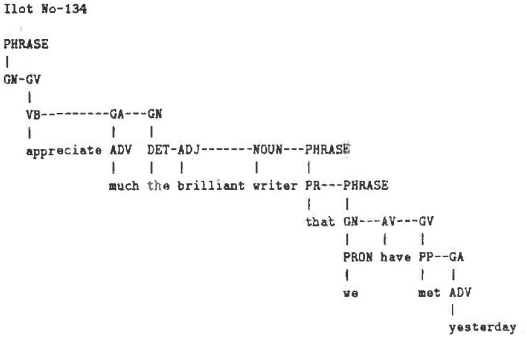


Figure 6.13: Seule une inférence en chaînage avant est applicable dans cette situation. Cette inférence crée deux interprétations en parallèle, à cause du fait qu'il y a plusieurs règles applicables. L'autre arbre est présenté dans la figure 6.14.

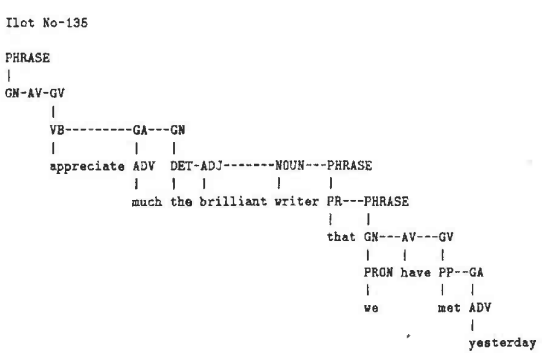


Figure 6.14: c.f.: la figure 6.13

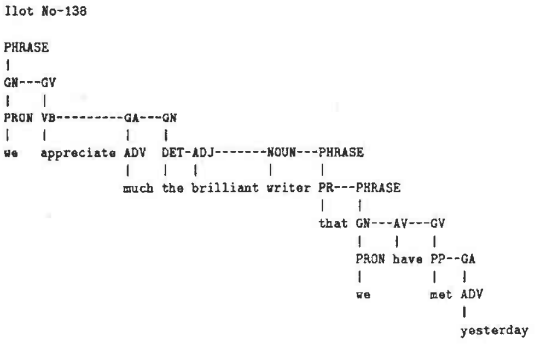


Figure 6.15: En utilisant l'évidence fournie par le niveau inférieur, l'îlot 134 est qualifié.

A cette étape, il n'y a plus qu'un îlot dans la conférence et la racine est le fait "axiome". L'interprétation de l'association est donc terminée. L'association propose donc l'îlot 140 comme l'interprétation au niveau supérieur.

7.3 Séquence de création des nœuds

Pour illustrer davantage le fonctionnement de notre constructeur syntaxique, nous montrons ici l'ordre de création des nœuds d'une interprétation complète. L'exemple représente une phrase chinoise dont l'arbre syntaxique est donné dans la figure 6.19. La traduction

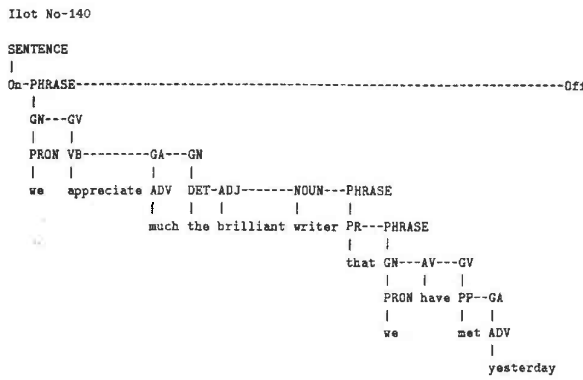


Figure 6.16: Une inférence en chaînage avant donne trois possibilités. Les deux autres sont présentées dans les figures 6.18 et 6.17. Toutes les regions du signal de la parole sont maintenant examinées. Parmi les trois possibilités, seule la première (140) est cohérente avec le signal et donc retenue. Les deux autres (141, 142) sont éliminées car leur validation demande encore des regions nonéxaminées.

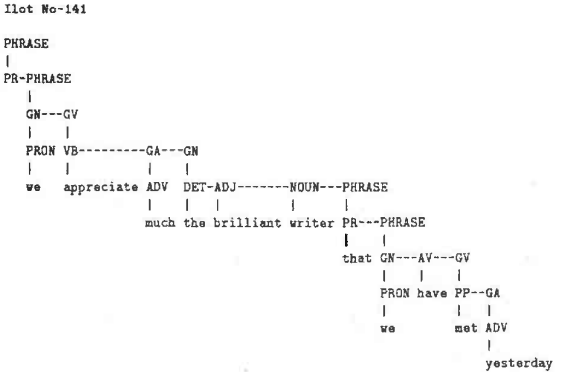


Figure 6.17: c.f.: la figure 6.16

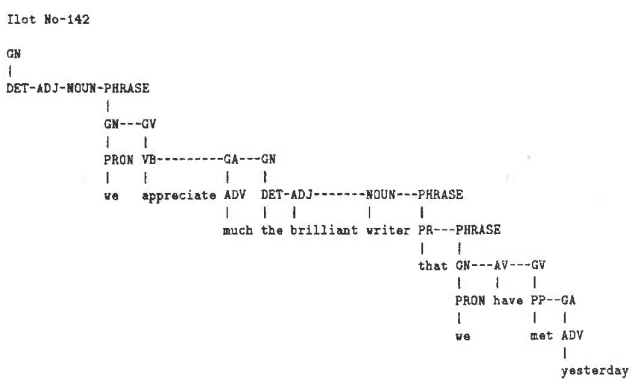


Figure 6.18: c.f.: la figure 6.16

de la phrase est *"déplacer le long cube à gauche du crayon bleu de l'arrière de la grande boule vers 21.83"*. Nous avons numéroté les nœuds successivement crées pour construire les interprétations et dans la figure 6.20 sont présentés les nœuds de l'interprétation finale.

On peut constater que la construction a commencé par l'instanciation du mot *bian.1* et la création du nœud *g105*, qui se trouve approximativement au milieu de la phrase. Puis plusieurs activations des fonctions de base, le chaînage avant, le chaînage arrière, la fusion, etc, sont déclenchées et l'arbre devient de plus en plus complet vers les extremités gauche et droite.

7.4 Complexité

Nous avons effectué un test pour mesurer de façon grossière la relation entre d'une part le temps d'execution et d'autre part le nombre des regles et la taille du vocabulaire. Nous avons ajouté notamment des règles et, pour l'interprétation une même phrase, nous obtenons le tableau 6.21.

8 Conclusion

L'incertitude de la conversion signal-symbole ne peut pas être résolue entièrement au niveau de la conversion. Il est indispensable d'utiliser d'autres contraintes a priori sur la structure du signal pour réduire l'incertitude, la syntaxe représantant une de ces contraintes.

Notre travail a porté sur l'interprétation des structures syntaxiques du signal en insistant sur le traitement de l'incertitude:

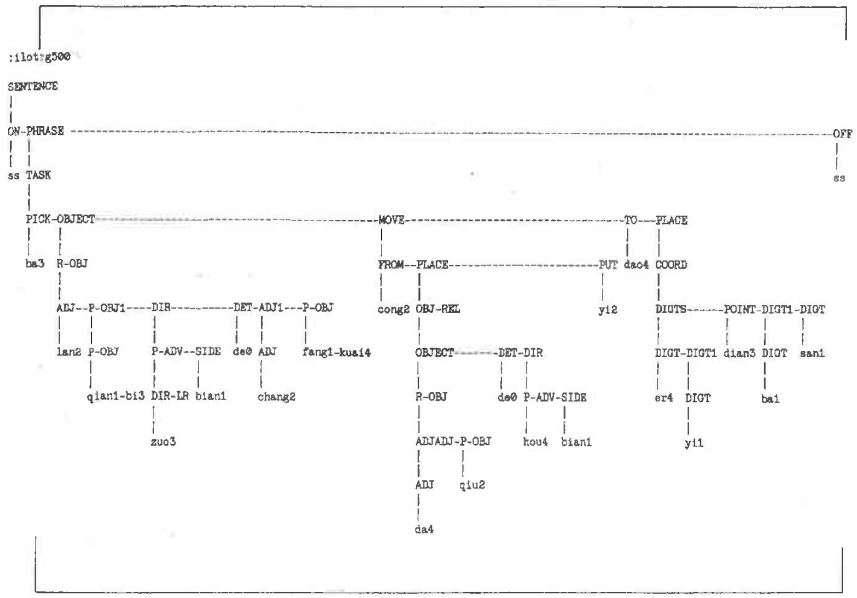


Figure 6.19: arbre syntaxique d'une phrase

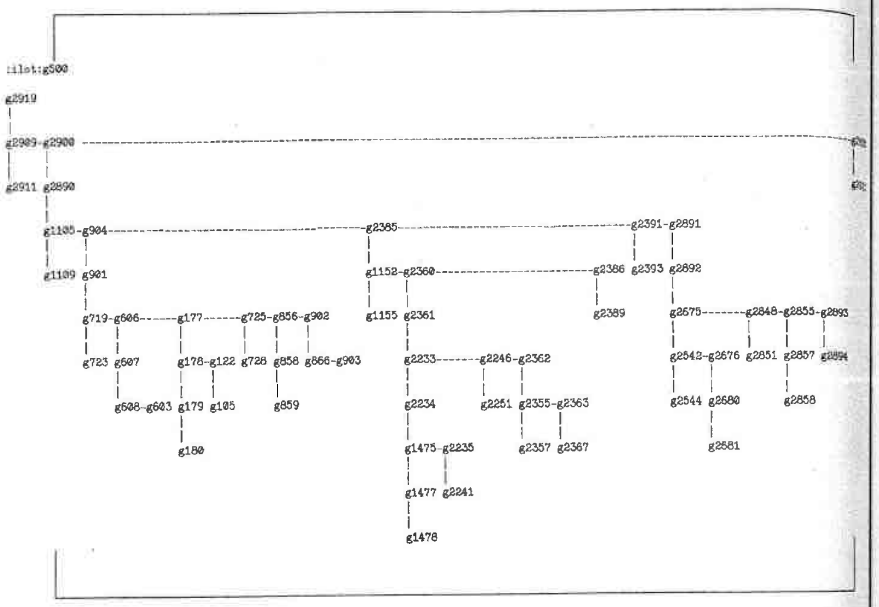


Figure 6.20: arbre syntaxique construit en partant du milieu approximatif de la phrase

Terme	Avant Modification	Après Modification	Augmentation(%)
Nombre de règles	160	217	35
Nombre de terminaux	130	163	25
Temps d'interprétation(s)	80	84	5

Figure 6.21: Résultat de test sur l'augmentation du temps de calcul en fonction du nombre de règles et du nombre de terminaux, dans l'interprétation d'une même phrase

- Nous avons montré que l'indéterminisme et l'incertitude sont deux problèmes de nature différente et doivent être traités par des méthodes différentes,
- Nous avons indiqué qu'en présence d'incertitude, l'identification des symboles terminaux doit se faire dans le processus d'analyse de structure abstraite et l'incertitude doit être supprimée de façon progressive et non brutale,
- Nous avons proposé un ensemble d'algorithmes d'analyse des structures sous incertitude. Ces algorithmes permettent d'effectuer l'analyse en chaînage mixte et d'utiliser au mieux l'information sur la structure du signal obtenue au cours d'analyse.

Le constructeur de structure syntaxique du signal que nous avons réalisé est capable de traiter les ambigutés due à la connaissance (*l'indeterminisme* de règles, la récursivité en particulier) et aux données (*l'incertitude* du signal à interpréter). Le processus d'interprétation est hautement guidé par l'information acquise pendant la construction. La structure du constructeur permet facilement d'ajouter des composantes, représentant des nouvelles sources de connaissances, afin d'améliorer la fiabilité et l'efficacité de l'interprétation.

Nous avons mesuré la complexité de notre constructeur et nous estimons qu'elle est en général inférieur à l'ordre $O(n^3)$. Ceci est normal car les grammaires ambigues sont acceptées et pour analyser une telle grammaire, des algorithmes d'analyse syntaxique équivalents tel que celui d'Earley [Aho 72] demandent la même complexité.

Partie IV

ARCHITECTURE D'UN SYSTEME D'INTERPRETATION

Chapitre 7

Société de Specialistes en Interprétation

La détermination de la validité d'une interprétation finale demande la vérification de la cohérence entre l'ensemble des connaissances a priori et le signal observé. La représentation à niveaux multiples, l'organisation et la coopération de sources de connaissances sont donc fondamentales dans des systèmes pouvant reconnaître et interpréter des structures complexes pour des signaux incertains. Nous proposons une architecture homogène pour organiser et contrôler un ensemble de sources de connaissances actives. Elle est fondée sur la décomposition du problème d'interprétation en sous-problèmes à des niveaux conceptuels successifs et sur la définition du contrôle explicitement lié avec ces niveaux. L'ensemble des connaissances est partitionné en groupes de sources de connaissances selon les niveaux d'abstraction. Chaque groupe a un contrôle local. L'échange d'information est assuré par une structure de données commune entre les sources de connaissances d'un même niveau et par un mécanisme de courrier entre deux niveaux différents. Le processus d'interprétation consiste en l'exécution de groupes de sources de connaissances guidée par des données et contrainte par des modèles des domaines, permettant de construire la solution de façon incrémentale. L'incertitude du signal est propagée à travers les différents niveaux. Cette architecture permet la coopération de connaissances pluridisciplinaires.

1 Introduction

1.1 Système de décision

La suppression de l'incertitude dans un processus d'interprétation du signal n'est possible que si certaines contraintes dans l'ensemble des connaissances a priori sur le signal à interpréter sont utilisées. En général, ces connaissances ont des propriétés importantes:

- Elles sont fragmentales, chacune définissant en partie un aspect du signal;
- Elles sont regroupables en niveaux conceptuels différents selon leur domaine d'intervention, bien que ce regroupement soit parfois flou;

- Elles contiennent de la redondance puisque certaines d'entre elles sont corrélées.

Ces connaissances fournissent un ensemble des contraintes sur les variations du signal et déterminent partiellement un modèle du signal.

L'interprétation du signal est un processus qui permet de donner une instance du modèle, parmi l'ensemble A indénombrable des possibilités. L'instanciation consiste à prendre une suite de décisions, en fonction de

- l'observation du signal ou par inférence des propriétés P du signal et
- les connaissances a priori.

L'application d'une séquence de décisions permet ainsi de réduire les alternatives et de donner un sous-ensemble d'instances du signal.

Une décision est un choix d'un sous-ensemble d'alternatives A' tel que:

$$D : P \times A \rightarrow A' \subseteq A$$

Une source de connaissances (knowledge source KS) est une entité de calcul dans un système de décision qui, munie d'une certaine connaissance a priori de nature homogène, est chargée de prendre une certaine catégorie de décisions.

Le modèle réel d'un signal incertain, – toutes les contraintes déterminant le comportement du signal –, contient tellement d'information que l'on ne peut jamais le déterminer entièrement. Si on imagine le modèle comme un réseau complexe, l'image de nos connaissances sur le signal dans le réseau peut être représenté par des squelettes du réseau, – un sous ensemble des liens. A l'aide de ces connaissances, en cas d'incertitude d'une décision, une source de connaissances peut émettre des hypothèses selon ces squelettes, pour combler l'information manquante. Ces hypothèses seront vérifiées (acceptées ou refusées) par d'autres sources de connaissances. Autrement dit, en utilisant les connaissances a priori, la collaboration de différentes sources de connaissances permet d'instancier le modèle du signal en présence d'incertitude.

1.2 Nécessité d'une architecture

Les sources de connaissances se caractérisent par:

- la relation qui les lie: il existe des niveaux conceptuels différents. Cela résulte du fait que les décisions sont hiérarchisables. A chaque niveau d'abstraction, pour décrire le signal, différents vocabulaires, termes et syntaxes sont utilisés. Nous utilisons le terme *domaine* pour désigner un niveau d'abstraction. L'architecture doit permettre
  - le groupement des sources de connaissances du même domaine et
  - la communication entre ces groupes de sources de connaissances.
- Il est nécessaire d'avoir un contrôle de ces sources de connaissances capable de s'adapter à la nature de chaque domaine car différents domaines peuvent avoir différentes façons d'utiliser des sources de connaissances. C'est à dire qu'il faut ordonner correctement la prise de décisions. Une décentralisation du contrôle est donc souhaitable.

- Les décisions ne sont pas toutes indépendantes. Une décision peut nécessiter ou en influencer d'autres. L'incertitude du signal exige un schéma d'utilisation de sources de connaissances particulière, sinon contradictoire:
  - le raisonnement pendant l'interprétation nécessite une identification des symboles et
  - l'identification des symboles demande un raisonnement.

  Par conséquent la coopération des sources de connaissances en interprétation, notamment l'utilisation d'une stratégie d'interprétation contrôlée par un modèle et guidée par des données est indispensable.
- En présence d'incertitude, l'évidence locale pour prendre une décision peut être insuffisante. L'incertitude du signal doit être modélisée explicitement pendant toute l'interprétation. Il faut laisser propager l'incertitude tant qu'on n'a pas suffisamment d'evidence pour la reduire. L'incertitude doit être diminuée progressivement dans le processus d'application des source de connaissances.
- Pour la construction de solutions, il n'y existe pas un chemin de décision systématique, prédéterminé et réaliste. En revanche, il est indispensable de construire des solutions par morceaux, de façon incrementale et indéterministe.

L'organisation des sources de connaissances est donc un problème fondamantal avec des retombées pour l'ensemble de l'I.A.

1.3 Architecture de Hearsay-II

La représentation et l'utilisation de connaissances pour un système d'interprétation est la première étape vers la solution. La recherche d'un modèle adéquat est donc de toute première importance. Un modèle d'organisation de sources de connaissances est une spécification des relations entre des sources de connaissances plus des règles d'activation de celles-ci, dans l'objectif de la construction d'une solution du problème. Le modèle doit permettre une coopération des connaissances distribuées dans les domaines du problème [Erman 80,DeMori 85b, Haton 85].

La structure du *blackboard* (tableau noir) [Lesser 77,Erman 80] fournit un modèle d'organisation de connaissances multiples. Un blackboard est une structure de données globale du système, à plusieurs niveaux, dans lequel différentes sources de connaissances indépendantes saisissent et enregistrent leurs données et leurs résultats d'interprétation. Dans une telle structure, chaque source de connaissances peut accéder aux résultats des autres et les différentes source de connaissances sont indépendantes au sens que

- elles ne s'appellent pas l'une l'autre directement,
- chacune ignore la présence des autres et
- l'intervention d'une source de connaissances est seulement provoquée par une modification dans le blackboard.

Cette structure est assez bien adaptée aux problèmes d'organisation d'un système où l'utilisation et la coopération de plusieurs sources de connaissances est indispensable. Elle a été reprise dans plusieurs travaux d'I.A. en plus de son utilisation d'origine – l'interprétation du signal de parole [Erman 80]:

- interprétation des scène [Hanson 78],
- analyse des photographies complexes [Nagao 79],
- interprétation de signaux sonar [Nii 82],
- interprétation de données des capteurs [Maser 86],
- estimation de la position d'un robot [Ong 86],
- outil de construction des systèmes experts à bases de connaissances multiples [Balzer 80, Aiello 81,Erman 81,HayesRoth 84,Laasri 87]
- compréhension du dialogue homme-machine [Mann 79],
- modélisation du processus de planification [HayesRoth 79],
- structuration du modèle cognitif [McClelland 81,Rumelhart 82],
- contrôle de systèmes à bases de connaissances [HayesRoth 85].

Ce modèle est devenu maintenant un modèle général pour la solution de problèmes [Nii 86b, HayesRoth 83]. Cependant, le blackboard est une entité conceptuelle et non une spécification de calcul [Nii 86b] et nous allons commenter cette architecture sur les plans de l'organisation des niveaux d'abstraction, l'indépendance entre les sources de connaissances et le mécanisme de contrôle.

Niveaux d'Abstraction

Dans un système à sources de connaissances multi-niveaux, les informations (données et résultats partielles) circulant à un niveau particulier ont des propriétés particulières et représentent un ensemble de concepts spécifiques du même niveau d'abstraction. Ces informations doivent être représentées sous une forme convenant la structure de données, les relations et les proriétés à exprimer explicitement à ce niveau. Un nveau différent entraîne une signification et une représentation différentes. Cette obervation a été signalée dans [Saitta 83]. Les KSs à chaque niveau disposent d'un langage (une representation des objets) différent des ceux des autres, il est indispensable de traduire les hypothèses entre des niveaux [Reddy 74]. L'information spécialisée (connaissances du domaine par exemple) doit être manipulée par un mécanisme uniforme. L'accès non restreint d'un niveau aux autres nécessite un grand nombre de transformations de représentation puisque généralement les symboles, le sens associé et les concepts ne sont pas les mêmes.

D'ailleurs, les accès libres entre chaque niveaux ne sont pas nécessaires, car en conséquence de ces niveaux d'abstraction, la communication entre les KSs est restrainte [Charniak 85]. Une KS génère inférence en général entre deux niveaux de concept consécutifs [Nii 82].

Dans le système Hearsay-II La plus part d'échange d'information sont faits entre des KSs de même niveau. Il n'y a que 3 accès qui ne sont pas concernées aux niveaux non consécutifs (ségment → mot, mot → phrase et phrase → mot) dont l'existence est discutable.

La contrainte de limitation de l'accès libre entre les niveaux non consécutifs ne diminue pas l'avantage de la structure du blackboard. Le regroupement de KSs de même niveau de concept rend le contrôle et l'accès de l'information simple et efficece.

Indépendance Entre KSs

L'influence entre KSs dans Hearsay-II est indirecte et implicite. Il faut clarifier cependant le sens de l'indépendance d'une KS vis-à-vis des autres. Ceci n'est vrai que si les données d'entrée d'une KS sont convenablement fournies par d'autres KSs. Autrement dit que le fonctionnement du blackboard est basé sur l'hypothèse que l'information nécessaire pour résoudre un problème par une KS est fournie systématiquement par d'autres KSs. Nous avons indiqué dans 1.2 que les décisions sont liées par données et par résultats. En conséquence, l'ordre d'intervention de KS sur le blackboard est très important, sinon imposé, entre certaines KSs. L'indépendance et l'ignorance concernent plutôt la réalisation et la modification internes (expertice, performance, etc) des KSs.

Les sources de connaissances ne peuvent pas être totalement indépendantes; il existe des relations du type production-consommation et d'ordre d'intervention entre les KSs.

Mécanisme de Contrôle

Le contrôle du blackboard d'Hearsay-II est implicite, guidé par l'interprétation partielle. Le mécanisme de contrôle du blackboard (scheduler) de Hearsay-II dont la stratégie est d'exécuter la KS qui a la plus haute priorité, décrite en termes de "effect potentiel", "signification globale" et "désirabilite de l'action de KS" [Erman 80] paraît trop empirique [Woods 79], pas suffisamment contraignante et difficilement modifiable pour une nouvelle application. Le comportement du système est largement dépendant de l'association de priorité. Ce mécanisme de contrôle est consédéré complexe et sophistiqué en général [HayesRoth 85]. En effect, les priorités associées aux KSs doivent obligatoirement prendre en compte l'ordre d'intervention des KSs à l'interprétation. Cet ordre est en grande partie imposé par la causalité de la production de l'information. Par exemple, en compréhension de la parole, chaque niveau d'abstraction interagit seulement avec les niveaux qui lui sont adjacents [Kay 85]. Par conséquent, l'ordre d'activation peut être défini en basant sur la dépendance de KSs.

Dans un système d'interprétation, chaque niveau d'abstraction traite un sous-problème de nature très différente et nécessite sa propre méthologie d'aborder des problèmes et sa propre stratégie de contrôle. Or, un contrôle global fondé sur priorité n'autorise pas explicitement cette possibilité. Il serait inadapté et inefficace de programmer en termes de priorité globale de KSs pour que, à chaque niveau, le contrôle ait ses particularités.

Un contrôle monotone est insuffisant pour modéliser l'interprétation. Le contrôle pour un système à connaissances de multi-niveaux est plus raisonnable et plus simple s'il est décentralisé dans plusieurs unités indépendantes et autonomes.

Conclusion

Le modèle d'Hearsay-II est insatisfait dans les aspects suivants, à cause de son mécanisme de contrôle global, indépendant des niveaux d'abstraction et à un seul niveau:

- l'implémentation explicite d'une stratégie de recherche donnée;
- l'étude du fonctionnement indépendante d'un groupe de sources de connaissances du même niveau;
- l'inclusion d'un contrôle local pour chaque niveau d'abstraction.

Nous essayons de résoudre ces problèmes. Notre travail consiste à définir une architecture à sources de connaissances multiples permettant une évolution d'une telle structure sur quelques points importants détaillés dans la section 2.2. Nous comparerons notre architecture avec d'autres modèles issus du blackboard d'Hearsay-II dans la section 6.2.

2 Société de spécialistes en interprétation

2.1 Présentation générale

Nous proposons un modèle pour l'organisation et l'utilisation de sources de connaissances multiples en interprétation du signal, la *société de spécialistes* avec niveaux conceptuels multiples.

Une société de spécialistes est constituée de plusieurs associations séparées, chacune modélisant un niveau d'abstraction du problème relatif à un domaine spécifique. Les connaissances sur le problème sont partitionnées en associations selon leur niveau d'abstraction et, dans chaque association, implantées par un groupe de spécialistes indépendants.

Un spécialiste est une entité de calcul spécialisée dans un aspect de la construction de la solution finale a un niveau donné. Les spécialistes appartenant à une même association partagent une conférence où ils peuvent lire et écrire des solutions partielles et l'état de solution du problème. La communication entre les associations est réalisée par un mécanisme de courrier à travers lequel les sous-problèmes et les solutions partielles sont transmis.

Le contrôle d'une société de spécialistes est en deux étapes:

1. le contrôle global qui réalise le passage de messages et actionne les associations;
2. le contrôle local dans une association qui dynamiquement détermine l'ordre d'intervention des spécialistes d'un niveau, permettant la réalisation d'une stratégie particulière.

Pendant une session d'activation, une association reçoit d'abord des solutions partielles d'une autre association d'un niveau d'abstraction supérieur ou inférieur. Les spécialistes discutent ensuite dans leur conférence, effectuent la création, la modification ou la suppression des solutions partielles. Les solutions, correctes seulement localement dans l'association, sont envoyées par un message vers une autre association compétente. En actionnant les associations aux différents niveaux, la solution finale est construite de façon incrémentale et l'incertitude de la parole est progressivement diminuée.

2.2 Nouveautés

La société de spécialistes est caractérisée par deux niveaux d'organisation hiérarchique (à distinguer avec les niveaux d'abstraction qui sont définis selon les concepts traités d'un problème):

- **Niveau société:** Dans une société de spécialistes l'ensemble des connaissances est partitionné en domaines (niveaux conceptuels). Chaque groupe de sources de connaissances assure la transformation de concepts d'un domaine à un autre. Tous les niveaux de transformation sont gérés par un mécanisme de contrôle global qui permet de réaliser des stratégies d'interprétation guidées à la fois par les données et par des modèles. L'échange d'informations entre niveaux est assuré par un mécanisme de transmission de messages.
- **Niveau association:** Une association correspond à un domaine spécifique, la transformation est faite par plusieurs spécialistes. Chacun d'entre eux est spécialisée dans un type de problèmes. Les spécialistes d'un domaine disposent d'une structure de données accessible à tous, appelée conférence, pour stocker données et résultats. Pour mesurer la qualité des solutions, chaque domaine est équipé en plus d'un mécanisme d'évaluation de l'interprétation partielle. Le contrôle de l'interprétation d'un niveau est confié à une source de connaissances particulière qui n'intervient pas directement dans la génération de solutions.

Cette organisation présente des caractéristiques nouvelles:

- Des sources de connaissances des niveaux conceptuels différents sont distinguées et regroupées explicitement. Chaque niveau dispose d'une structure de données commune pour l'échange de l'information. Cette distinction modélise le fait qu'une différence de niveau entraîne des différences dans la signification et la représentation des objets [Saitta 83] et évite certaines traductions des vocabulaires entre deux niveaux [Reddy 74]. Elle permet d'introduire la notion de niveau explicitement dans le contrôle de l'interprétation.
- La dépendance entre sources de connaissances est prise en compte dans le contrôle. En effet, l'ordre d'intervention d'une source de connaissances est en grande partie déterminé par la causalité de la production de l'information dans un problème réel.
- Au lieu de n'utiliser qu'un contrôle global, nous utilisons un contrôle à chaque niveau d'abstraction. Ceci autorise des stratégies différentes dans des domaines où les problèmes traités et les méthodes utilisées sont de natures différentes.

3 Définition du vocabulaire

La société de spécialistes est schématisée dans les figures 7.1 et 7.2.

Les termes introduits dans le texte ayant un sens particulier seront écrits en gras lors de leur première apparition et seront définis dans ce paragraphe.

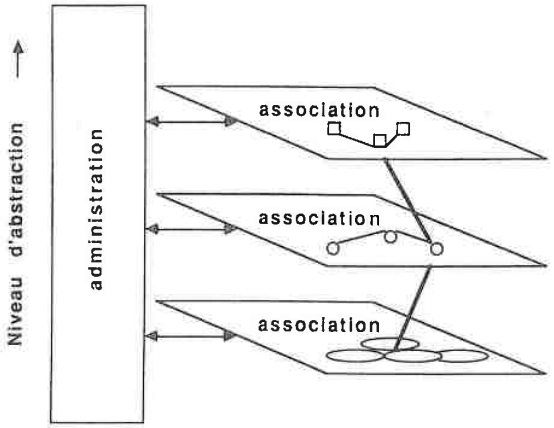


Figure 7.1: Société de Spécialistes — Architecture Générale d'une Société de spécialistes

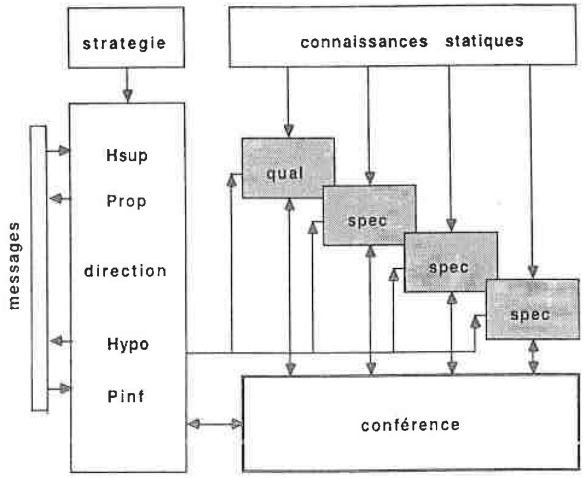


Figure 7.2: Société de Spécialistes — Structure d'une Association

3.1 Société de spécialistes

Notre modèle de société consiste en un ensemble d'**associations** de spécialistes, interactives et soumises à un contrôle commun. **administration**, qui organise le processus de transformation (interprétation). Chaque association réalise un niveau de transformation conceptuelle dans un domaine conceptuel particulier. Une société de spécialistes transforme un signal en concepts du domaine depuis le niveau d'abstraction le plus bas jusqu'au niveau le plus haut.

3.2 Association

C'est un groupe de sources de connaissances (**KS**s) manipulant des concepts de données et de résultats d'interprétation au même niveau et disposant d'une **conférence**. Les KSs dans une association emploient le même vocabulaire pour décrire l'information manipulée.

Une association assure l'interaction entre différentes KSs d'un même niveau conceptuel. Une association est capable de transformer des concepts de deux façons: du niveau inférieur vers le niveau courant (interprétation guidée par les données) et du niveau courant vers le niveau inférieur (interprétation contrôlée par le modèle). Les sources de connaissances d'une association donnée travaillent au même niveau d'interprétation partielle. Les tâches assurées par une association sont essentiellement les transformations de concepts, notamment:

- la génération des **hypothèse**s vers le domaine inférieur (où les concepts manipulés sont plus concrets),
- la vérification des hypothèses dévéloppées par le domaine supérieur (concepts plus abstraits),
- la génération des **proposition**s vers le domaine supérieur et
- la vérification des propositions fournies par le domaine inférieur.

Les différentes KSs que l'on trouve dans une association sont: un ou plusieurs **spécialiste**s associés, un **qualifieur** et une **direction**, mécanisme de contrôle de l'association. La communication entre les différentes associations se fait par **messages**. Les associations de différents niveaux sont physiquement indépendantes et conceptuellement superposées.

3.3 Direction

La direction est une KS définissant la stratégie (règles donnant l'ordre et manière dont une KS intervient à un point d'interprétation) d'une association. A chaque étape de l'interprétation, la direction doit décider, en fonction de l'interprétation partielle, le spécialiste à activer, en se fondant sur la notion de causalité de production-consommation des informations manipulées par les KSs, ainsi que sur des critères de priorité entre spécialistes.

3.4 Spécialiste

C'est une KS munie d'une compétence dans un aspect d'un domaine. Un spécialiste est une entité de calcul dont la tâche est :

- de créer (instanciation des modèles morceaulés),
- de supprimer (application des contraintes fragmentales du domaine),
- de modifier (reconstruction des liens entre les objets),
- d'intégrer (unification des solutions partielles),
- d'évaluer (calcul de la qualité)

une interprétation partielle. La priorité d'un spécialiste en cas de conflit est l'ordre d'intervention dans la conférence. Celui qui a effectué la dernière modification est le plus prioritaire.

3.5 KS

Il s'agit d'une source de connaissances. Une KS est une entité munie de connaissances spécialisées et capable de résoudre une partie de problème dans un domaine particulier. Les spécialistes, les qualifieurs et les directions sont tous des KSs. Une KS peut être un programme classique (structures de données et algorithmes), un système à base de connaissances (système expert ou même une autre société de spécialistes), ou d'autres agents de calcul. Certaines KSs fonctionnent en mode contrôlé par les données, certaines par les modèles et d'autres ont la possibilité de fonctionner dans les deux sens.

3.6 Qualifieur

Le qualifieur assure une évaluation de la qualité de l'interprétation après une phase d'activation des KSs. Dans un système d'interprétation, toutes les données, et par conséquent tous les résultats, sont incertains, ce qui nécessite une mesure de qualité de l'interprétation. Une interprétation correcte doit donner une cohérence entre le signal observé et le concept final extrait. Dans chaque domaine, l'interprétation partielle doit être consistante avec le modèle du domaine et l'évidence acquise à partir du signal et fournie par le niveau inférieur. Le qualifieur donne une mesure de cette consistance. Il doit décider si une interprétation est acceptée ou refusée. La méthode de calcul de qualité employée par un qualifieur dépend bien entendu du domaine.

3.7 Connaissances statiques

Ces connaissances constituent l'ensemble des contraintes a priori sur le problème du niveau constituant le modèle de l'interprétation qui ne varie pas au cours de l'interprétation. Elles limitent à chaque instant de l'interprétation l'ensemble des choix possibles.

3.8 Conférence

C'est une structure de données où les spécialistes d'un même domaine saisissent les données et enregistrent les résultats de l'interprétation. L'état de la recherche, par exemple l'ensemble des solutions déjà parcourues pendant les **sessions** précédentes, est aussi stocké dans la conférence. La conférence contient des instances du modèle du domaine qui sont instanciées où partiellement instanciées. Autrement dit les conférences contiennent ce qui est actuellement supposé avoir eu lieu, tenu compte des évidence extraites du signal. La conférence est en quelque sorte un blackboard pour les spécialistes d'une association.

Dans la conférence est aussi stockée une liste de problèmes déjà résolus, avec leurs solutions. Si le même problème est reposé, l'association peut renvoyer directement la solution sans faire du calcul répétitif. Ceci n'est possible que grâce à la décomposition du problème en niveaux multiples.

3.9 Administration

C'est un mécanisme de contrôle des différentes associations. Il assure la réalisation d'une stratégie de contrôle, l'organisation des sessions d'interprétation pour chaque association et la transmission des informations, données et résultats, d'une association. L'administration transmet des messages entre les associations et donc assure la communication. Le destinataire d'un message est déterminé par deux niveaux de décisions:

- Il peut être spécifié par l'association qui envoie le message. Ceci permet à l'association de donner sa préférence mais nécessite une connaissance des différentes associations.
- Si le destinataire n'est pas spécifié, l'administration analyse l'objet du message (cf. 3.12) et l'envoie à une association compétente.

L'administration peut contrôler la circulation des messages et réaliser une stratégie. Un autre rôle est aussi de fournir un moyen de consultation des interprétations partielles dans des domaines différents.

3.10 Hypothèse

L'hypothèse représente une interprétation potentielle de niveau conceptuel du domaine D_{i-1} émise par l'association A_i en fonction de son modèle, des propositions de $A_{i-1}, A_{i-2}, ...$ et des hypothèses de $A_{i+1}, A_{i+2}, ...$ Les hypothèses générées par A_i sont envoyées sous forme d'un message vers l'une des associations de niveau inférieur (A_{i-1}, A_{i-2}...). Une hypothèse peut devenir une proposition si elle est vérifiée. L'hypothèse modélise la notion d'analyse descendante (la notion d'analyse ascendante est modélisée par la proposition expliquée au paragraphe suivant).

3.11 Proposition

C'est une interprétation potentielle émise par A_i en fonction de son modèle, des propositions de $A_{i-1}, A_{i-2}, ...$ et des hypothèses de $A_{i+1}, A_{i+2}, ...$ Une proposition générée par A_i est

envoyée vers l'une des associations supérieures pour être vérifiée par A_j avec $j = i+1, i+2$, etc ... La proposition modèlise la notion d'analyse ascendante.

Les propositions ont la propriété de vérifier toutes les contraintes locales imposées par des connaissances du niveau d'abstraction et toutes les contraintes des niveaux inférieurs (elles auraient été enlevées sinon). Les propositions peuvent être compétitives (plus d'une solution pour un même problème) et peuvent être fausses du point de vue des niveaux supérieurs. Le nombre de propositions dépend de la qualité de l'interprétation partielle. Notamment, plus les mesures de qualité des unités du niveau inférieur sont proches, plus ce nombre est grand.

3.12 Message

Le message permet la communication entre deux associations arbitraires à travers l'administration. Un message contient les attributs suivants:

- *identifieur:* numéro d'identification du message,
- *expéditeur:* association qui envoie le message,
- *destinataire:* association qui devrait recevoir le message, optionnel,
- *réponse-à:* un identifieur de message,
- *objectif:* la tâche demandée,
- *données:* une liste de problèmes à résoudre ou résolus et
- *instruction:* information du contrôle de l'interprétation.

Par l'envoi d'un message, une association transmet une liste de problèmes du domaine résolus localement (propositions) vers les niveaux supérieurs et des problèmes non résolus du fait de la limitation des connaissances du domaine (liste de problèmes à résoudre) vers les niveaux inférieurs.

3.13 Session

Une session est une phase complète d'interprétation d'une association. La constitution d'un ensemble de propositions d'une association nécessite en général plusieurs phases d'activation. Entre-temps les associations voisines sont activées pour confirmer les hypothèses et les propositions émises. Lors d'une session, une association a donc besoin de mémoriser l'état de recherche de solutions dans sa conférence pour pouvoir débuter la session suivante.

4 Fonctionnement et contrôle

Dans cette section, nous discutons l'échange d'information, le fonctionnement et le contrôle dans une société de spécialistes, au niveau de l'association et au niveau de la société.

4.1 Echange d'information

L'échange d'information est indispensable pendant l'interprétation pour que les sources de connaissances construisent des solutions de façon incrémentale.

Dans une société de spécialistes, l'échange d'information s'effectue à deux niveaux hiérarchiques:

- Le premier est au niveau des spécialistes, assuré par la conférence de chaque association. Comme la conférence est inaccessible par les spécialistes en déhors de l'association, une association est fermée et donc est indépendante des autres associations.
- Le deuxième est au niveau d'une association, assuré par le mécanisme de courrier. Tous les messages sont passés par l'administration. Si le destinataire est precisé par l'association qui envoie le message, l'administrattion transmet directement le message à l'association destinée. Sinon elle va analyser l'objectif du message et transmettre le message à une association compétente.

Les deux modes de transmission de message sont appellés respectivement *transfert direct* et *transfert filtré.*

4.2 Fonctionnement

Spécialistes

Contrairement à un programme classique où toutes les connaissances disponibles sur le domaine sont intégrées dans un module et où une interprétation ne sera proposée et enregistrée qu'après avoir vérifié toutes les préconditions, dans une association chaque spécialiste *dit ce qu'il pense* dans la conférence et *on discute.* Les interprétations proposées sont ensuite vérifiées, modifiées, complétées ou supprimées par l'intervention des autres KS.

Association

Les opérations fondamentales d'une association se caractérisent par

- plusieurs phases de génération des hypothèses vers une association de niveau inférieur et
- une phase de génération des propositions vers une association de niveau supérieur,

sous la stratégie de contrôle définie par la direction. Le résultat d'une session de l'association A_i est soit d'émettre un ensemble d'hypothèses vers une des associations du niveau inférieur par l'envoie d'un message, soit de donner un ensemble de propositions à une des associations supérieures, toujours par un message. Tant qu'il est possible d'améliorer l'interprétation, A_i ne propose pas d'interprétation partielle. Mais pendant le construction de l'interprétation, l'association peut émettre des hypothèses à vérifier. La figure 7.3 illustre le flux d'informations d'une association.

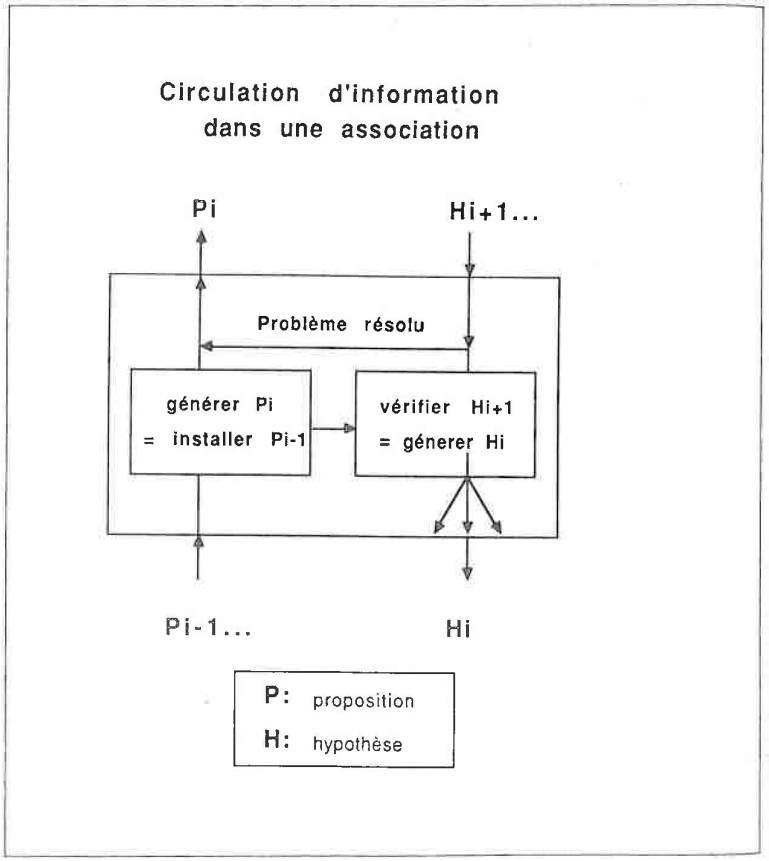


Figure 7.3: Les entrées et les sorties d'une association

Société

Du point de vue de la société, le fonctionnement est caractérisé par l'activation des associations. Cette activation est provoquée par l'échange de messages entre les associations et dépend donc de l'état et du résultat de l'interprétation partielle. Organisé par l'administration, le fonctionnement d'une société est du type *déclenché par le résultat d'interprétation* et de nature opportuniste. L'interprétation en alternance ascendante et descendante est facilement réalisable.

4.3 Contrôle

Le comportement d'une societé de spécialistes, une entité composée de plusieurs composants interactifs, ne dépend pas uniquement de ces composants mais aussi de la façon dont ils sont structurés et activés. La structure de contrôle doit permettre de définir une coopération des associations afin que l'utilisation des KSs et des informations données soient effectives lors de la résolution d'un problème.

Dans le processus d'interprétation du signal, il est généralement impossibe de spécifier a priori la séquence d'exécution des sources de connaissances. Dans la société de spécialistes, le contrôle d'interprétation est spécifié par des règles qui permettent de déterminer dynamiquement l'exécution des membres de la société. L'architecture de la société, le mode de communication entre les membres de la société sont orientées vers la réalisation d'un mécanisme de contrôle distribué à deux niveaux:

- Le niveau inférieur *(niveau association)* est réalisé par le directeur de l'association. Ce contrôle est chargé de choisir un spécialiste à exécuter. Pour cela, le directeur interprète un ensemble de règles à travers lesquelles une stratégie est codée en termes de:
  - la relation causale sur la production et la consommation de l'interprétation partielle ou la dépendance d'information entre les spécialistes,
  - l'état de l'interprétation de la conférence et
  - la connaissance sur les spécialistes dans l'association.
- Le contrôle du niveau supérieur *(niveau société)* est réalisé par l'administration. L'objectif est de déterminer une association entreprenant la session suivante. Ce contrôle est caractérisé par les quatre types suivants:
  - *demande-réponse:* issue du mode *transfert direct* qui active l'association demandée par le message. Ce type permet aux associations d'avoir la possibilité de contacter une association préférée mais nécessite la connaissance sur l'autre association.
  - *select:* issue du mode *transfert filtré* du mécanisme de transfert de message. L'administration examine le contenue du message puis décide, en fonction de sa connaissance sur les associations et d'une stratégie, l'exécution d'une association compétente.
  - *diffusion:* Ce type de contrôle active toutes les associations pour qu'un certain problème soit éventuellement résolu par une association de compétence inconnue ou pour que le processus d'interprétation soit synchronisé.

– *global:* Dans ce type de contrôle, l'administration n'organise pas immédiatement l'interprétation pour répondre la tâche demandée par une association dans le message courant. Par contre, pour certaines raisons globales, elle active une autre association. Ce contrôle autorise la planification du processus d'interprétation.

Le contrôle doit permettre de mêler convenablement les deux types de stratégie de solution de problèmes, i.e.: la stratégie descendante et la stratégie ascendante. En termes de raisonnement, Le première est une stratégie de chaînage arrière et la seconde est une stratégie de chaînage avant. Nous avons montré dans le chapitre 6 que pour les problèmes d'interprétation complexes, une approche purement descendante ou purement ascendante provoque inévitablement une explosion combinatoire de solutions partielles dûe à l'incertitude intrinsèque du le signal à interpréter et à la faible restriction applicable aux interprétations partielles concurrentes. Une stratégie mixte permettant d'utiliser l'information sur la structure du signal obtenue pendant l'interprétation est souhaitable. Le contrôle de la société de spécialistes est adapté de façon naturelle à ce type de stratégie.

L'architecture de la société de spécialistes décentralise le contrôle de l'interprétation dans les associations, permettant un contrôle local variable adapté au domaine et un contrôle global simple et efficace. L'exécution d'une session peut avoir deux résultats possibles: soit il y a des hypothèses partielles générées ou soit il y a des propositions partielles générées. Dans le premier cas, le contrôle doit se transférer vers le domaine du niveau inférieur et dans le deuxième cas vers le niveau supérieur. Cette description est recursive. Par conséquent, il y a un certain nombre d'interactions entre différentes associations avant d'arriver à donner l'interprétation finale. Dans la figure 7.4 est déssinée l'évolution de l'association en activité en fonction du nombre de sessions.

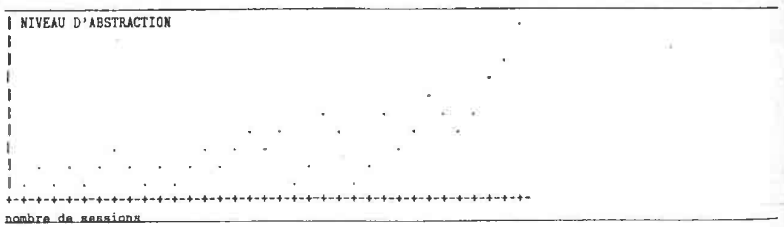


Figure 7.4: Le niveau d'abstraction de l'association active en fonction du nombre de sessions d'interpretation

Les interprétations dans chaque conférence deviennent anssi itérativement de plus en plus sûres et complétes, et l'incertitude diminue. La structure de contrôle au niveau de l'administration est donc simple et indépendante des domaines.

Le processus d'interprétation se termine d'autant plus rapidement que l'incertitude du signal d'entrée est faible. Lorsque les qualités des interprétations concurrentes sont approximativement égales, c'est à dire que l'ambiguïté dûe à l'incertitude du signal est grande,

l'espace de recherche est automatiquement élargi afin de reporter les choix jusqu'à ce que plus d'informations soitent disponibles.

5 Discussions

Le problème de l'interprétation est décomposé de façon naturelle en sous-problèmes dans des domaines de niveau d'abstraction différents. L'espace des solutions et l'ensemble des connaissances sont partitionés dans les domaines:

- L'espace des solutions est divisé en niveaux d'analyse contenant des solutions partielles et
- l'ensemble des connaissances est divisé en sources de connaissances spécialisées dans un aspect d'un domaine particulier.

Cela facilite l'introduction de connaissances humaines dans des diciplines différentes.

L'architecture présente une bonne modularité. Les associations sont indépendantes et la composition, le fonctionnement extérieur et la communication avec l'extérieur de chacune est homogène. Dans la société de spécialistes il y a à la fois une séparation nette et une coopération suffisante entre les KSs. L'ajout et la modification d'un domaine ou d'un spécialiste est relativement facile dans une société de spécialistes, permettant d'inclure de nouveaux résultats de recherches dans des domaines concernés. En particulier, puisque les domaines sont indépendants, cette architecture favorise la réalisation des grands systèmes implantés par plusieurs auteurs travaillant indépendemment. Ces auteurs ont la liberté de concervoir leur propre ensemble de KSs et d'associations. L'architecture fournit un support d'expérimentation de l'interprétation.

De différentes stratégies de contrôle d'interprétation ou de recherche de solutions peuvent exister et peuvent être modifiées facilement dans des domaines différents. Cela fournit la possibilité d'introduire des stratégies adaptées à la nature du problème à chaque niveau d'abstraction.

La vérification des hypothèses émises par un domaine supérieur peut se faire dans tous les domaines inférieurs. Autrement dit, non seulement il existe une coopération explicite entre deux domaines de niveau consécutif mais aussi l'aspect descendant de l'analyse se réalise à travers plusieurs niveaux, ce qui complète l'analyse ascen-descendante traditionnelle fonctionnant entre deux niveaux consécutifs et donne une stratégie d'interprétation plus générale et plus proche de l'intelligence humaine.

Une fois que l'interprétation du domaine le plus supérieur est terminée, les interprétations des tous les domaines inférieurs sont aussi validées. Autrement dit, les unités conceptuelles de tous les niveaux sont reconues.

6 Comparaisons avec d'autres architectures

6.1 Système expert

Nous utilisons le terme *système expert* pour désigner les logiciels utilisant une base de règles, un mécanisme d'inférence avec un seul type de raisonnement. La base de règle est relative à un domaine spécifique, pour offrir un comportement comparable à celui d'un expert humain face à un problème bien restreint.

Le modèle du problème dans un système experts reste trop simplifié et incomplet. La simulation de la partie importante de l'intelligence humaine provenant de son pouvoir d'adaptation et de sa faculté d'utiliser différentes stratégies en fonction de situation rencontrée reste difficile.

Les systèmes experts ont été utilisés pour l'interprétation de signaux [Nazif 84,Carbonell 86]. Un système expert n'est pas adapté à la représentation des connaissances multiples car l'architecture ne permet pas de définir des niveaux différents et d'actionner convenablement des sources de connaissances.

6.2 Architecture blackboard

Depuis Hearsay-II, un certain nombre de systèmes que l'on peut qualifier de *descendant du blackboard* ont été proposés. Ces systèmes se différencient essentiellement par leur mécanisme de contrôle dont nous allons commenter quelques uns.

CRYSALIS

Le contrôle de CRYSALIS se fait en deux niveaux [Nii 86b]:

- En fonction d'un résumé de la solution courante dans le blackboard et des règles données par l'utilisateur, un superviseur appelle séquentiellement une liste de tâches;
- Une tâche évalue des conditions portées sur des changements du blackboard et des variables, et exécute une liste de fonctions qui correspondent aux sources de connaissances d'application et modifient le blackboard.

D'autres systèmes utilisant un mécanisme de contrôle similaire incluent HASP [Nii 86a], ATOME [Laasri 87]. Il n'y pas de scheduler dans cette famille de contrôles, adaptés aux problèmes où les tâches sont parfaitement prédéfinies.

BB1

Le système BB1 [HayesRoth 86] a une architecture blackboard dans son contrôle. Les source de connaissances se divisent donc en deux parties: celles qui construisent les solutions sur le blackboard de domaine et celles qui construisent les plans d'action sur le blackboard de contrôle. Le contrôle se caractérise par les trois étapes suivantes:

- Un *interprète* exécute l'action d'une source de connaissances. Le contenu d'un blackboard est modifié.
- Un *agenda-manager* ajoute les sources de connaissances dans un agenda. La partie condition de ces sources de connaissances est satisfaite par le changement du blackboard.
- Un *scheduler* donne la priorité à chaque source de connaissances dans l'agenda, selon le plan courant du blackboard de contrôle.

L'introduction du blackboard de contrôle permet de réaliser la planification hiérarchique au cours de la construction de solution. Un autre point intéressant est que les traitements des sources de connaissances de domaine et de contrôle sont unifiés dans ce mécanisme de contrôle.

Le nœud du DVMT

Un nœud dans DVMT [Durfee 87] contient deux blackboards, le blackboard des buts et le blackboard des évenements (des données). Le contrôle est défini par trois tables indépendantes

- la table *évennement* → *but*,
- la table *but* → *sous-but*,
- la table *but* → *source de connaissances*,

et un scheduler. Les tables permettent de calculer les sources de connaissances exécutables à partir des évennements. Le scheduler choisit et exécute la source de connaissances selon un critère basé sur la confiance des hypothèses qui pourraient être produites et sur le score du but.

Ce type de contrôle est conceptuellement claire et rigide.

Conclusion

Parmi les structures de contrôle existentes, nous distinguons deux types de contrôles:

- les contrôles basés sur la décomposition de tâches: HASP, CRYSTALIS.
- les contrôles basés sur un scheduler: BB1, un nœud d'un DVMT.

Dans tous systèmes, il peut avoir un contrôle hiérarchique, mais le sens du *niveau* dans le contrôle n'est pas le même que le sens du *niveau* du blackboard. Dans l'interprétation du signal, le contrôle dépend non seulement des changements sur le blackboard, mais aussi de la nature de ces changements (niveau conceptuel). Le contrôle d'une société de spécialistes est basé sur *demande-réponse* entre les associations de différents niveaux. Ce type de contrôle est adapté aux problèmes où on a besoin de distinguer les deux types d'intéraction: entre

les KSs du même niveau et celles entre les niveaux différents. Le blackboard est décomposé physiquement en parties et a évolué vers un ensemble de conférences relatives à des domaines différents. Les groupes de KSs fonctionnant à l'intérieur d' un même niveau de concept sont physiquement regroupées. Cette distinction implique que non seulement les KSs sont indépendantes, comme dans un blackboard, mais aussi qu'un niveau d'abstraction peut travailler indépendamment des autres.

L'utilisation de la notion de priorité implique qu'il y a des connaissances pour associer une priorité à une action. Dans une societé de spécialistes, au lieu de coder ces connaissances sous forme numérique, nous introduisons des règles de dépendances qui expriment comment ces priorités sont obtenues. Grâce à l'association du contrôle aux niveaux différents La stratégie de contrôle globale se simplifie considérablement en considérant la contrainte naturelle de dépendances conceptuelles. Puisque les domaines sont indépendants, le contrôle local dans un domaine est aussi plus facile à définir et à réaliser.

Puisqu'il n'y a plus de structure globalement accessible par toutes les sources de connaissances, chaque niveau est relativement fermé, la communication avec d'autres niveaux se faisant par le mécanisme de communication par message.

Enfin, nous signalons qu'aucun contrôle n'est approprié dans toutes les situations. Sans application spécifique, il est difficile d'évaluer une structure de contrôle objectivement.

6.3 Société d'experts

Une société d'experts [DeMori 86] est un ensemble d'experts coopératifs. Un expert est un agent de calcul qui exécute des programmes de raisonnement utilisant des connaissances structurelles et procédurales. La communication entre les experts, essentiellement des demandes et des contrôles, est réalisée par l'échange de messages. En plus de connaissances statiques, appelées *LTM long term memory*, chaque expert dispose d'une mémoire de travail – *STM short term memory* – où sont stockées des données et des hypothèses. Une STM est lisible par tous les experts mais ne peut être modifiée que par son expert.

Les connaissances de chaque expert sont représentées par un ensemble de *plans*. Un plan est une séquence d'*items* donc chacun peut contenir:

- une *précondition* qui doit être vérifiée pour déclencher une règle;
- un *opérateur* qui contient des procédures pour vérifier la relation de la précondition;
- un *algorithme* pour évaluer l'évidence de l'hypothèse générée par la règle.

Si on considère qu'un expert avec ses mémoires associées est equivalent à une association dans notre société de spécialistes, alors les deux architectures sont semblables. En fait, le travail d'un expert dans une société d'experts est accompli par un groupe de spécialistes organisés par un contrôle local dans notre société de spécialistes.

7 Application en interprétation de parole

7.1 Introduction

La compréhension de la parole continue est un problème typique d'interprétation de signal. Le processus d'interprétation identifie et spécifie un ensemble d'objets du monde réel, connus par l'interprète, et établit une relation entre eux, soumise à des contraintes imposées par des modèles de différents niveaux. Dans cette application, le système reçoit en entrée le signal observé et doit fournir une représentation de la structure profonde du signal, réutilisable directement par des agents qui réagissent à la parole d'entrée. Le processus doit

- réduire la quantité de données sans perdre de l'information et
- réaliser des applications successives du domaine perceptif vers des domaines conceptuels.

L'indéterminisme de la parole entraîne un nombre prohibitif de solutions partielles. Il est indispensable d'exploiter les connaissances des différents niveaux d'abstraction dans le processus de communication parlée pour diminuer ce nombre [Pierrel 82,Haton 85]. Une bonne structure d'organisation des différents niveaux est donc nécessaire pour fournir des interprétations les plus cohérentes possibles avec le signal vocal. Notre structure doit permettre de supprimer dès que possible les interprétations partielles jugées incompatibles avec le signal et de retarder des décisions jusqu'à avoir suffisamment d'évidences [Scagliola 85].

Essentiellement trois problèmes sont à résoudre: la diminution de l'ambiguïté et de l'incertitude dans le signal par l'utilisation de sources de connaissances et l'interprétation du signal, le couplage entre différents niveaux de traitement, et l'organisation de stratégies d'analyse.

Nous présentons dans les paragraphes qui suivent une instantiation de la société de spécialistes en interprétation de la parole continue. La présentation est organisée selon les domaines conceptuels des associations.

7.2 Domaine acoustico-phonétique

Ce domaine réalise la transformation d'information d'une représentation numérique et continue (le signal vocal) en une représentation symbolique et fournit des symboles primitives pour les domaines supérieurs. Le résultat est une classification floue de la succession de spectres à court terme. A chaque instant d'échantillonnage, le signal est étiqueté en classes phonétiques associées à une valeur de la fonction d'appartenance.

Les KSs de ce domaine proposent, sur des mesures de paramètres acoustiques, les premières interprétations pseudo-phonémiques. Ces KSs sont essentiellement des algorithmes de classification floue. Pour modéliser le fait qu'il est impossible de segmenter correctement la parole continue à l'aide de connaissances locales, nous adoptons une approche de reconnaissance sans segmentation préalable. Plusieurs expériences ont montré que retarder ou éviter complétement les décisions de segmentation améliorerait la performance d'un système [Klatt 80,Bahl 83].

7.3 Domaine phonologique-lexical

La donnée d'entrée de ce domaine est une liste de symboles phonétiques préclassés. La conférence du domaine consiste en phonèmes et mots identifiés classés selon l'axe du temps et le score de reconnaissance. Des connaissances phonologiques sont utilisées pour traiter les variations contextuelles de prononciation. La connaissance sur la qualité d'interprétation du niveau inférieur est également utilisée. Dans ce domaine les unités conceptuelles *mots* sont identifiées comme propositions à partir des symboles fournis par le domaine acoustico-phonétique. Ce domaine est chargé également de vérifier les hypothèses émises par le domaine syntaxico-sémantique.

7.4 Domaine syntaxico-sémantique

Dans ce domaine, les connaissances syntaxico-sémantiques sont utilisées pour établir la relation entre mots et découvrir la structure logique de la phrase. Le rôle de l'information sémantique devient fondamental si le modèle du langage est assez général et si le vocabulaire est grand car la grammaire peut alors être peu contraignante [Pierrel 81]. La séparation nette entre syntaxe et sémantique n'est qu'une simplification. Notre modèle syntaxico-sémantique est une grammaire sémantique [Burton 76] sous forme de règles, de type context free. A chaque règle de la grammaire est associé un ensemble d'actions qui traduisent la phrase partiellement interprétée en une fonction directement exécutable. L'implémentation des KSs syntaxiques et sémantiques dans une même association rend efficace l'interprétation et élimine eventuellement l'explosion combinatoire des interprétations possibles du niveau.

L'incertitude du signal provient principalement de la variation inter- et intra-locuteur. Un principe en reconnaissance de la parole est de garder en parallèle plusieurs interprétations partielles qualifiées à chaque niveau d'abstraction et de prendre la décision finale lorsque l'évidence est suffisante.

L'interprétation est fondée sur la notion d'îlots de confiance [Haton 76,Erman 80,Wolf 80, Masini 85]. La conférence du domaine syntaxico-sémantique est constituée des îlots d'interprétation partielle, représentés par des arbres d'instances du modèle. Nous utilisons la technique de recherche en faisceau [Lowerre 80], permettant de poursuivre simultanément plusieurs interprétations partielles de qualité suffisamment élevée. Le diamètre du faisceau est contrôlable en fonction des interprétations partielles et des ressources de calcul.

L'association syntaxico-sémantique fonctionne comme un moteur d'inférence capable de mener le raisonnement en chaînage avant et arrière et de propager l'incertitude du signal. Nous avons étudié ce moteur dans le chapitre 6. Nous résumons les caractéristiques du moteur et le rôle de ces composantes, qui sont réalisées par des spécialistes de notre architecture. Des points plus précis concernant lles résultats seront présentés dans le chapitre 9.

L'ensemble des sources de connaissances du domaine permettent

- de développer en parallèle plusieurs interprétations partielles possibles,
- de raisonner sur plusieurs îlots de confiance puis de fusionner ces îlots,
- de mesurer la qualité d'interprétation partielle pour contrôler la recherche en faisceau,

- d'utiliser l'information acquise au cours de l'interprétation (connaissance dynamique).

La direction de l'association est un interprète de règles de dépendance entre les sources de connaissances.

Les spécialistes de l'association assurent les tâches suivantes:

- *chaînage avant:* Effectuer des inférences guidées par la portion de signal examinée, i.e: pour chaque îlot d'interprétation dans la conférence, déclencher toutes les règles dont la partie droite contient la racine de l'îlot.
- *chaînage arrière:* Effectuer des inférences contrôlées par le modèle syntaxico-sémantique, i.e: pour chaque îlot d'interprétation dans la conférence, déclencher toutes les règles dont la partie gauche est une des feuilles de l'îlot. Ce spécialiste génère des hypothèses à envoyer vers le domaine de niveau inférieur. Le chaînage est contrôlé vers la gauche ou la droite d'un îlot qui peut se situer initialement à n'importe quelle place dans une phrase.
- *contrainte positionnelle:* L'apparition d'une sous-structure est légale seulement à certaines positions dans une phrase. Cette connaissance, obtenue pendant la phase de pré-compilation des règles, est utilisée pour détecter une interprétation partielle qui sera finalement impossible et par conséquent pour limiter l'espace de recherche.
- *préparation des hypothèses:* elle consiste à collectionner tous les noeuds terminaux instanciés et non identifiés puis envoyer la liste de ces noeuds vers l'association phonologique-lexicale comme hypothèses.
- *estimation de qualité:* pour mesurer la qualité de chaque îlot dans la conférence et décider ceux qu'il faut conserver. Ce spécialiste contrôle le diamètre de la recherche en faisceau.
- *qualifieur:* calculer la qualité d'une interprétation.
- *unification:* elle consiste à fusionner les îlots de confiance pour construire une interprétation plus complète.
- *prédiction de phonèmes:* Pour augmenter le rapport du nombre d'interprétations conservées sur le nombre d'interprétations générées lors du développement d'un îlot vers ses deux extrémités, seuls les îlots ayant des mots terminaux dont tous les phonèmes qui les constituent ont un score suffisant sont conservés.

Nous avons détaillé le principe et le fonctionnement de ces spécialistes dans le chapitre 6.

7.5 Domaine compréhension

Ce domaine dispose d'un modèle du monde réel comme contrainte exploitee. Le modèle du monde réel décrit les objets physiques manipulés et les relations statiques et dynamiques entre eux dans l'application spécifique. Trois KSs doivent exister: une KS recherchant et transformant en forme explicite le sens profond du discours, une KS exécutant des actions en réaction de la parole comprise, une KS sur la cohérence pragmatique.

7.6 Exemple d'interprétation

Nous donnons dans ce paragraphe des extraits de l'interprétation d'une phrase. La présentation de l'aspect technique de l'interprétation du signal de la parole est dans les chapitre 8 et 9. La phrase chinoise prononcée est *Ba.3 lan.2 qian.1 bi.3 zuo.3 bian.1 de.0 chang.2 fang.1 kuai.4 cong.2 da.4 qiu.2 de.0 hou.4 bian.1 yi.2 dao.4 er.4 yi.1 dian.3 ba.1 san.1* (Place le long bloc à gauche du crayon bleu et derrière la grande boule, à la place deux un point huit trois). Un chiffre de 0 à 4 suivi du point est utilisé pour indiquer la catégorie de la variation tonale. La notation

```
phon=[ian 117 120 124 .8049998 -1]
```

dans la trace d'exécution se lit comme: Le signal de l'indice 117 à l'indice 124 est considéré comme l'image du phonème *ian* avec une vraisemblance moyenne de 0.8049998 (maximum à 120). La figure 7.5 donne un extrait de la liste des symboles à interprèter.

(186	d	67	m	54)				
(187	d	58	h	46)				
(188	d	53	h	43	u	40)		
(189	k	48	d	41	h	39	u	38)
(190	k	69)						
(191	k	65)						
(192	u	46	k	34	h	33)		
(193	u	37	h	32	e	32	ou	31)
(194	e	33	iou	29	ou	28	u	21)
(195	e	28	iou	25	ou	19	ao	19)
(196	er	29	ua	26	e	25	iou	22)
(197	er	35	ua	29	e	24)		
(198	er	39	ao	36	ang	24	ua	24)
(199	er	44	ao	41)				
(200	er	42	ao	42	ao	33)		
(201	er	45	ao	41	ao	33)		
(202	ao	42	er	36	ao	32	e	29)
(203	ao	47	ao	36	e	35)		
(204	ao	38	e	30	ao	29	ang	24)
(205	ss	31	b	28	d	28	m	26)
(206	ss	39	b	35	ss	32	d	31)

Figure 7.5: Extrait de la représentation symbolique du son /kuer/ du mot *kuai(r)*. A chaque instant (186-206) le signal est préclassé en une liste de symboles phonémiques dont chaqu'un est associé d'un coefficient de vraisemblance.

La première étape de l'interprétation de l'association "lexicon" (domaine phonologique-lexical) consiste à localiser des zones où l'incertitude est minimale. Le résultat est la proposition de mots "yan.2", "tian.2", "dian.3" et "bian.1" au milieu du signal de la phrase (les centres de ces mots se trouvent à environ 118):

```
Message from acoustic-phonet to lexicon. Intention: proposition
-> Active association: lexicon
phon=[ian 117 120 124 .8049998 -1]
Verified: lexc=[yan.2 117 120 124 7 .8049998 -1]
phon=[t 112 114 116 .3009999 -1]
Verified: lexc=[tian.2 112 118 124 7 .6111536 -1]
phon=[j 98 99 100 .4448333 -1]
phon=[d 112 114 116 .4299999 -1]
```



```
Verified: lexc=[dian.3 112 118 124 7 .6607691 -1]
phon=[b 112 113 115 .39525 -1]
Verified: lexc=[bian.1 112 118 124 7 .6169999 -1]
```

L'association syntaxico-sémantique formule des îlots de confiance après la réception de la proposition du niveau inférieur. Ensuite les spécialistes du domaine sont activés selon un ordre décrit par des règles de contrôle et déterminé dynamiquement:

```
Message from lexicon to syntactic-semantics. Intention: proposition
-> Active association: syntactic-semantics
ilot:g100 (L) pred=[102 108 115 92 107 113 (h t g b j zh ch c z d s ...]
ilot:g100 (R) pred=[126 unsticked 132 125 140 146 (n g b t d m er ...]
ilot:g101 (L) pred=[97 unsticked 110 92 107 113 (h t g b j zh ch c ...]
Executing the specialist: "forward-chain-main"
New list of ilots = [10]
Executing the specialist: "backward-chain-left"
New list of ilots = [9]
Executing the specialist: "position-constrain"
New list of ilots = [9]
Executing the specialist: "backward-chain-right"
New list of ilots = [5]
Executing the specialist: "hypoth-prepare"
New list of ilots = [5]
```

Il y a émission des hypothèses vers le niveau inférieur. Le contrôle est ensuite transféré à l'association lexicon. Lorsque le contrôle est redonné à l'association syntaxico-sémantique, l'évaluation des interprétations partielles est activée et certaines interprétations sont conservées:

```
Message from syntactic-semantics to lexicon. Intention: hypothese-word
-> Active association: lexicon
phon=[eng 127 128 129 .4269999 -1]
(124,10000,10000,()) node:g124 dengl not found
Message from lexicon to syntactic-semantics. Intention: proposition
-> Active association: syntactic-semantics
Executing the specialist: "qualify"
ilot:g107 (WEEK-DAYS)   0.6112 accepted
ilot:g112 (SOME)   0.6608 accepted
ilot:g113 (POINT)   0.6608 accepted
ilot:g115 (SIDE)   0.6170 accepted
ilot:g116 terminal rejected
New list of ilots = [4]
```

En fonction de connaissances locales, quatre ilots sont conservés. Ils ne sont cependant pas tous corrects globalement. Avant d'effectuer l'extension des arbres d'interprétation, on calcule la liste de phonèmes mesurables à l'extrémité de chaque nouvel ilot afin de limiter le nombre des arbres partiels:

```
Executing the specialist: "predict-ph"
New list of ilots = [4]
Message from syntactic-semantics to lexicon. Intention: hypothese-ph
-> Active association: lexicon
Message from lexicon to syntactic-semantics. Intention: proposition
```

Au cours de l'interprétation, il peut y avoir plusieurs dizaines d'ilots concurrents. Ils sont développés en parallèle jusqu'à ce que l'évidence acquise soit suffisante pour enlever ceux qui sont non cohérents avec le signal:

```
Executing the specialist: "backward-chain-right"
New list of ilots = [35]
Executing the specialist: "hypoth-prepare"
```

```
New list of ilots = [35]
Message from syntactic-semantics to lexicon. Intention: hypothese-word
-> Active association: lexicon
phon=[ao 155 156 157 .4566666 -1]
phon=[x 151 152 153 .4199999 -1]
Verified: lexc=[xiao.3 151 154 157 9 .3757141 -1]
(151,154,157,9) node:g854 xiao3 recognized
(147,154,162,9) node:g767 chang2 recognized
phon=[sh 147 149 151 .5875999 -1]
(142,10000,10000,()) node:g838 shui not found
phon=[uen 165 166 167 .3943332 -1]
phon=[g 152 153 155 .2904999 -1]
(142,10000,10000,()) node:g841 guen4 not found
------
Message from lexicon to syntactic-semantics. Intention: proposition
-> Active association: syntactic-semantics
Executing the specialist: "qualify"
ilot:g160 (PLACE) Multual-excluded : jiu.3
ilot:g166 (PLACE)   0.2042 accepted
------
New list of ilots = [9]
```

Au fur et à mesure de l'interprétation, grâce aux contraintes définies par des modèles des différents niveaux et à l'utilisation obtenue sur la structure de la phrase courante, les ilots actifs correspondant aux interprétations correctes deviennent de plus en plus solides et le nombre d'interprétations partielles est réduit.

```
Executing the specialist: "backward-chain-right"
New list of ilots = [2]
Executing the specialist: "hypoth-prepare"
New list of ilots = [2]
Message from syntactic-semantics to lexicon. Intention: hypothese-word
-> Active association: lexicon
phon=[ss 529 533 538 .8249999 -1]
Verified: lexc=[ss 529 533 538 25 .8249998 -1]
(529,533,538,25) node:g2921 ss recognized
(529,533,538,25) node:g2918 ss recognized
------
Executing the specialist: "position-constrain"
New list of ilots = [2]
Executing the specialist: "forward-chain-main"
New list of ilots = [2]
End of expert society execution.
```

Les interprétations obtenues sont finalement:

1. ilot = ilot:g500 plausibility = 0.32028 ss ba.3 lan.2 qian.1 bi.3 zuo.3 bian.1 de.0 chang.2 fang.1 kuai.4 cong.2 da.4 qiu.2 de.0 hou.4 bian.1 yi.2 dao.4 er.4 yi.1 dian.3 ba.1 san.1 ss

2. ilot = ilot:g499 plausibility = 0.31492 ss ba.3 lan.2 qian.1 bi.3 hou.4 bian.1 de.0 chang.2 fang.1 kuai.4 cong.2 da.4 qiu.2 de.0 hou.4 bian.1 yi.2 dao.4 er.4 yi.1 dian.3 ba.1 san.1 ss

Au cours de l'interprétation de cette phrase, la société a généré 500 ilots avec 2921 nœuds. Il y a eu 119 messages échangés dans des associations. 446 hypothèses lexicales et 209 hypothèses phonémiques ont été formulées.

8 Conclusion

L'approche classique de la solution d'un problème est de donner un algorithme bien défini. Pour cela on n'est obligé d'étudier à fond le problème, les connaissances a priori, et de *compiler* ces connaissances sous un format particulier. Pour les problèmes complexes, la solution risque d'être introuvable à cause de la limitation de capacité de mémoire ou de raisonnement.

De trouver un tel algorithme n'est pas l'objectif de l'I.A. L'approche A.I. cherche à fournir une représentation de ces connaissances de façon *naturelle*, ou *la plus proche de la façon dont les connaissances nous viennent spontanément à l'esprit* [Kayser 84]. Une telle représentation permet non seulement de résoudre le problème en utilisant le plus possible des connaissances existantes, mais aussi l'amélioration de la qualité et l'ajout des nouvelles connaissances.

La structure de société de spécialistes à niveaux conceptuels multiples semble bien appropriée pour réaliser des transformations successives guidées à la fois par des données et par des modèles d'un signal en une description symbolique de haut niveau. Cette stragégie d'interprétation est généralement considérée comme la plus puissante [Nandahakumar 85, Nii 82,Haton 85]. La coopération des sources de connaissances permet de construire des solutions définies par des contraintes de différentes provenances de façon incrementale.

L'architecture de société de spécialistes offre un environnement souple, modifiable et un modèle conceptuellement simple et naturel pour le développement des systèmes d'interprétation. En particulier, du fait que les domaines sont indépendants, cette architecture favorise la réalisation de grands systèmes implantés par plusieurs auteurs travaillant indépendamment. Ces auteurs ont la liberté de concevoir leur propre ensemble de KSs et d'associations. L'architecture fournit un support d'expérimentation de l'interprétation globale.

Nous avons montré un exemple d'application de ce modèle à la compréhension de la parole continue. Grâce à architecture, nous avons pu réaliser facilement

- la stratégie ascendante-descendante d'interprétation entre chaque niveau, et
- un contrôle local pour chaque domaine de traitement.

L'architecture de société de spécialistes est assez générale et peut être utilisée dans les situations où la résolution d'un problème nécessite une coopération entre des sources de connaissances de niveaux conceptuels différents avec un vocabulaire riche à chaque niveau, telles que l'interprétation de signaux, le dialogue homme-machine, la lecture automatique, la gestion et la planification de production [Doumeingts 83].

Partie V

INTERPRETATION DE LA PAROLE CONTINUE

Chapitre 8

Interprétation de la Parole Continue

Le signal de la parole résulte d'un codage séquentiel d'une série de symboles portant des messages à transmettre. La reconnaissance et la compréhension de la parole est un problème typique d'interprétation. Nous étudions le processus d'interprétation de signal de parole continue en exposant notre conception générale. Nous présentons ensuite un système de compréhension de parole continue. Nous proposons un modèle à profils de paramètres pour la reconnaissance des phonèmes, une méthode de recherche lexicale, et une approche de compréhension de la parole continue sans segmentation préalable.

1 Introduction

La parole est une vibration sonore produite par l'appareil phonatoire humain destinée à la communication. Parmi d'autres moyens de communication, la communication orale se caractérise par les points suivants:

- C'est souvent le moyen le plus agréable à utiliser pour communiquer.
- L'homme a parfaitement acquis le moyen de communication. On peut atteindre un degré considérable de profondeur, de précision et de finesse dans la transmission de la pensée.
- Grâce au dialogue, l'échange d'informations est efficace. La communication orale permet rapidement d'enlever l'ambiguïté, de saisir les points non éclaircis et de réorganiser l'information suivant la connaissance des interlocuteurs.

1.1 Production de la parole

Le processus de production de la parole est en fait un codage utilisant un ensemble de connaissances communes aux interlocuteurs. Ce processus est très compliqué et jusqu'à présent aucun modèle justifié et convaincant n'a été établi. On peut schématiser de façon

simpliste ce processus pour expliquer quelques aspects du phénomène de la production de la parole:

$$Concept \overset{codage}{\rightarrow} Symbole \overset{production\ d'excitat.}{\rightarrow} Excitation \overset{mouvement\ muscul.}{\rightarrow} Son \overset{transmi.}{\rightarrow} Signal$$

où *Concept* est le concept de départ, *Symbole* est une suite de symboles respectant une grammaire qui code le concept de façon sérielle afin de le transmettre par la voie orale, *Excitation* est l'excitation musculaire du système phonatoire pour produire le son, *Son* est le signal à la sortie de bouche, et *Signal* est le signal observé et donc la donnée du système d'interprétation.

L'étude des conversions entre les différentes représentations de la parole fournit un ensemble de connaissances a priori, mais cet ensemble est encore incomplet, simpliste et difficile à utiliser en interprétation de la parole.

1.2 Profondeur d'interprétation

Selon la profondeur de l'interprétation du signal, on peut distinguer grossièrement deux types d'interprétations: la reconnaissance et la compréhension.

Reconnaissance

La reconnaissance d'un signal consiste à partitionner et identifier le signal de manière que chaque partition corresponde à un symbole préspécifié dans un vocabulaire connu a priori. La reconnaissance est donc un processus de classification du signal en symboles dans *Symbole* décrit dans 1.1. Un système de reconnaissance possède deux parties essentielles:

- l'extraction des caractéristiques qui consiste à instancier un modèle du signal choisi a priori. Cette partie est entièrement dépendante du signal. Il existe de nombreux modèles du signal de la parole mais, à présent, aucune théorie permet de fournir une solution sur la construction d'un modèle [Chen 87] satisfaisant.
- la classification des instances qui associe le signal à une séquence de symboles.

Compréhension

La compréhension d'un signal fournit en plus une représentation structurelle des objets dans l'ensemble *Concept*, décrit dans 1.1, du signal avec le sens de chaque symbole identifié de façon que le résultat est directement utilisable par un système réagissant sur le monde extérieur.

Dans un système de compréhension, on insiste sur l'utilisation efficace de deux type de connaissances:

- l'ensemble des contraintes qui porte sur le problème et
- l'information dynamique sur la solution obtenue en analysant le signal.

Le degré de profondeur de compréhension peut être très variable selon les connaissances utilisées et selon les mécanismes d'exploitation de ces connaissances.

En raison de la nature de la parole, l'interprétation du signal de la parole demande des études multi-disciplinaires incluant: le traitement du signal uni-dimensionel, la reconnaissance des formes statistique et syntaxique, la statistique, l'intelligence artificielle, l'informatique, la psychologie, la phonétique et la phonologie, la linguistique, les neurosciences, etc.

1.3 Variabilité

Influence de la variabilité

La variabilité de la parole est le phénomène lié au fait que des sons conceptuellement identiques peuvent être physiquement différents. Le locuteur a la liberté de prononcer un symbole de façon différente avec une certaine tolérance. Deux prononciations ne sont, dans aucun cas, identiques. Ceci implique, sauf une mémorisation spéciale, qu'un système d'interprétation de parole ne travaille jamais sur une phrase qui a déjà été reconnue. La perception humaine est capable de tenir compte de la variabilité lors de la saisie du message essentiel, tandis que pour une machine, la suppression de l'influence de la variabilité en reconnaissance de la parole a fait l'objet de travaux depuis plus que vingt ans et rien d'extrodinaire n'est obtenu. A cause de la variabilité, il est difficile de construire des systèmes de reconnaissance de parole indépendants du locuteur — les systèmes capables de reconnaître la parole de n'importe quel locuteur n'ayant pas participé à l'apprentissage.

Variabilité d'origine physiologique

A cause de la différence physique de chaque individu, la parole produite est variable. Par exemple, le conduit vocal du système phonatoire des hommes est en général 15% plus long que celui des femmes.

Variabilité liée à la vitesse d'articulation

Le changement de vitesse d'articulation n'introduit pas seulement le phénomène de dilatation ou compression (linénaire et non linéaire), mais aussi la modification de la répartition spectrale [Miller 81].

Variabilité liée à la coarticulation

La coarticulation est l'influence qu'exerce un son sur un autre son contigu, appelée aussi variation contextuelle car un son placé aux différents endroits peut se prononcer différemment. Cette modification contextuelle résulte sans doute d'un effet d'inertie mécanique mais également d'une réorganisation de l'effet articulatoire [Lonchamp 84]. La coarticulation est liée au mouvement continuel de l'appareil d'articulation et à la tendance à articuler plus facilement, du locuteur. La conséquence de la coarticulation est que:

- les traits acoustico-phonétiques pour un évènement phonétique sont repartis dans les segments adjacents.
- la distance entre les prononciations d'une même unité phonétique dans des contextes différents est augmentée.

Solutions

Pour surmonter les influences de la variabilité, cinq niveaux de traitements peuvent être envisagés:

- Les références multiples de symboles. Les sons ayant fréquemment des variations importantes sont volontairement représentés par plusieurs références. Cette méthode est efficace pour le phénomène la projection dispersée mais elle est incapable de traiter le phénomène de projection superposée.
- L'amélioration de la représentation paramétrique. On essaie de caractériser la parole par un ensemble de paramètres indépendants des locuteurs. On cherche notamment des paramètres ou des traits acoustiques de la parole relativement invariants vis-a-vis des locuteurs [Cole 83].
- La recherche de nouvelles distances. A chaque ensemble de paramètres doit être associé une distance convenable pour mesurer la divergence entre deux vecteurs de paramètres, sinon les paramètres n'ont pas de sens en reconnaissance. Des distances adaptées à la propriété de perception d'humaine [Itakura 75], pondérées par l'inverse de la variance de chaque composante [Tohkura 87] et pondérées dans le domaine fréquenciel [Paliwal 82] ont été proposées.
- La conversion signal-symbole adaptative. On mesure certains paramètres de la parole et on ajuste la fonction de transfert de la conversion selon la valeur de ces paramètres [Lowerre 77,Stern 83,Brown 83,Shikano 86]. La justification de cette approche est que la perception humaine de la parole est essentiellement un processus d'adaptation [Flanagan 72]. En particulier, l'homme est très sensible à la différence de répartitions spectrales présentées successivement [Flanagan 72] et les répartitions absolues ne portent pas beaucoup d'information sur le message transmis.
- L'interprétation symbolique des résultats de préclassification. On effectue une préclassification du signal qui convertit le signal dans un ensemble de symboles intermédiaires on applique et ensuite des règles pour convertir ces symboles intermédiaires en des symboles finaux [Gong 86a].

1.4 Conclusion

Les problèmes fondamentaux à resoudre dans l'interprétation de la parole sont:

- la modélisation du signal de parole;
- l'utilisation efficace des connaissances issues des niveaux différents;

- l'apprentissage automatique du système, c'est à dire l'instanciation automatique des modèles paramétriques dans le système.

A l'heure actuelle, il n'existe pas encore un modèle fonctionnel explicite de la parole capable de décrire la variabilité dûe au locuteur et au contexte et à la vitesse d'élocution.

L'utilisation des références multiples peut résoudre partiellement le problème de variabilité mais ce n'est qu'une technique loin d'être satisfaisante. Lorsque le nombre de locuteurs augmente un tel système doit disposer de plus en plus de références et, par conséquent, il y aura de moins en moins de différences entre les références. La performance est donc limitée.

Le problème de l'influence contextuelle est plus compliqué et plus difficile à résoudre par la technique de références multiples. La déformation, l'insertion et la suppression des formes nécessitent des modèles fonctionnels qui caractérisent explicitement le processus de la production de la parole. Hélas, on n'a pas encore des résultats satisfaisants dans ce domaine.

Cependant, le fait que l'homme arrive véritablement à reconnaître la parole d'une un très grande variété de locuteurs indique que l'information portée par la parole est *suffisante* pour la reconnaissance. Autrement *théoriquement* la reconnaissance de la parole continue multi-locuteurs est possible.

2 Progrès importants

Nous synthétisons dans cette section les progrès les plus remarquables depuis les quinze dernières années dans le domaine de la reconnaissance et compréhension de la parole. Dans [Pierrel 87] on trouve l'historique et les commentaires sur les grands systèmes.

2.1 Programmation dynamique

Vintsyuk [Vintsyuk 68], puis Sakoe et Chiba [Sakoe 71] ont introduit la programmation dynamique pour le récalage temporel non-linéaire de la parole. La programmation dynamique [Bellman 57] permet de comparer de façon optimale deux formes de longueurs différentes avec variations non-linéaires de vitesse de prononciation.

2.2 Architecture I.A.

Le système Hearsay-II de CMU [Lesser 77] propose l'architecture de blackboard avec l'objectif d'inclure des sources de connaissances multiples dans le système de reconnaissance de la parole. Le blackboard permet aux différentes sources de connaissances de communiquer et de construire la solution finale de façon incrémentale. Cette architecture a permis une inclusion facile de différentes entités de calcul non homogènes. Le taux de reconnaissance des mots est de 87% sur un vocabulaire de 1011-mots avec une syntaxe restreinte.

2.3 Réseau uniforme du language

Le système Harpy de CMU [Lowerre 80] adopte un réseau uniforme où le modèle du langage est entièrement intégré, après précompilation. Dans ce système, les phrases grammaticalement correctes sont représentées par une réseau où chaque nœud est un mot terminal. Ces mots sont ensuite remplacés par leurs réseaux de prononciation. Des règles phonologiques sont enfin appliquées dans le réseau résultant afin de produire différentes variations de parole à cause de la coarticulation. La reconnaissance d'une phrase consiste à trouver un chemin dans le réseau qui accepte la phrase, en utilisant la stratégie de recherche en faisceau.

2.4 Modèles Markoviens

L'utilisation des modèles stochastiques, notamment les sources markovennes cachées (HMM) pour la modélisation des évènements acoustiques de la parole [Baker 75b] et la loi d'emission de texte dans la reconnaissance de la parole continue [Jelinek 76] a commencé il y a plus de dix ans. Le système d'IBM est fondé sur la théorie de l'information où le locuteur qui prononce un texte T et le décodeur acoustique (convertisseur signal-symbole) sont combinés conceptuellement, appelé le canal acoustique dont la sortie est une chaîne C. Le problème est, connaissant la probabilité $p(T)$ que le générateur de texte produise le texte T et la probabilité $p(C/T)$ que le décodeur transforme le texte T en symbole C, de trouver un décodeur linguistique qui optimise la vraisemblance $p(T/C)$ en utilisant la formule:

$$p(T/C) = p(T)p(C/T)$$

Le modèle du langage traité et la propriété de transmission du canal acoustique sont supposés être deux sources markoviennes. Fondé sur ces modèles, le système de IBM a obtenu un taux de reconnaissance

- de 98% en 1983 [Bahl 83] sous les mêmes contraintes que celles utilisées pour tester Hearsay-II et
- de 92.2% à 99% en 1985 [Hoskins 85] sur un vocabulaire de 5000 mots en mode monolocuteur et mots isolés.

2.5 Quantification vectorielle

Les laboratoires Bell [Rabiner 77,Buzo 80] ont introduit la quantification vectorielle et les méthodes de clustering pour créer des références statistiquement plus représentatives et en même temps économiques au stockage. L'idée essentielle de cette technique est de profiter du fait que, dans l'espace de représentation de la parole:

- les vecteurs n'occupent que des sous-espaces sous forme de nuages;
- il est possible de représenter des nuages par leurs représentants sans trop de perte d'information;
- un vecteur peut être remplacé par le numéro de son plus proche représentant.

Ceci permet l'inclusion de références multiples pour un son du même locuteur ou de locuteurs différents. Les systèmes basés sur cette technique ont obtenu un taux de reconnaissances jusqu'à 99.8% dans le contexte de multilocuteurs sur un vocabulaire de 10 chiffres pour les mots isolés et connectés.

2.6 Machines connexionnistes

L'application des machines connexionnistes dans l'interprétation de la parole consiste à effectuer la conversion du signal de la parole en des symboles primitifs. Une machine connexionniste est un réseau d'éléments de traitement simple et peut réaliser une fonction de conversion par l'apprentissage. Le réseau peut stocker des connaissances sur la fonction sous forme des coefficients de lien entre les éléments et peut servir en reconnaissance. Les résultats sur les mots isolés, petits vocabulaires et dans un contexte monolocuteur sont impressionnants: un taux de reconnaissance de presque 100% [Rumelhart 86,Bourlard 87, Peeling 86]. Une présentation de ces machines en détail est dans 5.3.

Malgré les progrès effectués, le problème de l'interprétation de la parole n'est pas encore résolu car les systèmes actuels sont tous soumis à une ou plusieurs des contraintes suivantes:

- la dépendance au locuteur;
- la limitation de taille du vocabulaire accepté par le système;
- la prononciation en mots isolés;
- la syntaxe limitée;
- la tâche restreinte.

Lorsque l'on enlève ces contraintes, le taux de reconnaissance baisse tellement qu'un système peut devenir pratiquement inutile.

3 Approches de reconnaissance

3.1 Approche programmation dynamique

L'approche la plus intuitive et directe en reconnaissance automatique de la parole est de comparer globalement le signal de la parole prononcée, ou la forme de test, avec toutes les formes de référence connues à priori, dans un espace de représentation. La phrase reconnue sera la suite de symboles représentée par la forme de référence qui réalise la plus petite distance à l'image du signal inconnu. A cause de la variabilité de la parole, le problème essentiel de cette approche est de pouvoir comparer deux formes de longueurs différentes. La solution de ce problème se fonde sur le principe de programmation dynamique de Bellman [Bellman 57]:

Si la solution d'un problème est décomposable en une suite de solutions intermédiaires (*S.I.*) et les *S.I.* vérifient certaines conditions, alors l'optimisation de la solution finale implique que toutes les S.I. sont elles aussi optimales.

En application de la reconnaissance de parole [Vintsyuk 68], la programmation dynamique autorise la distorsion linéaire et non-linéaire dans l'axe du temps dans le calcul de la similarité acoustique entre une forme inconnue et une forme de référence, permettant ainsi de traiter convenablement les variations qui provoqueraient la substitution, l'insertion et la suppression. La programmation dynamique est une technique efficace car l'alignement optimal et la distance minimale sont calculés en même temps. Selon l'espace de représentation choisi, l'image des formes de test et de référence peut être:

- une suite de vecteurs paramétriques [Sakoe 71],
- des traits acoustiques [DeMori 86] ou
- des chaînes de phonèmes préclassés [Charpillet 85].

L'application adaptée de la programmation dynamique est la reconnaissance des lettres de l'alphabet, chiffres ou noms de villes.

Le modèle mathématiquement rigide, la solution optimale (au sens d'un critère de distance spécifique) garantie, le temps de calcul relativement constant et les algorithmes concis sont généralement considérés comme des propriétés intéressantes de cette approche.

En général, la programmation dynamique demande une complexité algorithmique de $n \times m$. n et m étant respectivement la longueur temporelle de la forme de test et la longueur de la forme de reférence. Des compromis peuvent être faits pour réduire le calcul jusqu'au $O(n)$ [Nakagawa 83].

La technique de programmation dynamique est développée d'abord pour des mots isolés avec petit vocabulaire [Sakoe 71]. Ceci a constitué une étape révolutionnaire en reconnaissance automatique de la parole. La reconnaissance de mots enchaînés par la programmation dynamique consiste à comparer l'image inconnue à la concaténation des formes de réferences des mots isolés. A ce sujet trois méthodes sont très connues:

- l'algorithme à *two level* de Sakoe [Sakoe 79] et
- l'algorithme à *level building* de Myers et Rabiner [Myers 81].
- l'algorithmes en une passe de Bridle [Bridle 82] et Nakagawa [Nakagawa 83].

Récemment, l'étude sur l'utilisation de la programmation dynamique en reconnaissance de parole est menée aux problèmes suivants:

- l'utilisation de la quantification vectorielle, de l'analyse clustering [Rabiner 77,Boyer 87] et des références multiples de chaque mot du vocabulaire a permis à la programmation dynamique d'avoir une bonne performance en multi-locuteurs.
- la modélisation de l'effet de coarticulation entre les mots successifs dans des petits vocabulaires spécifiques [Martino 84,Boyer 87].

- l'autorisation des frontières floues du début et de la fin de la parole [Rabiner 78a, Rabiner 81,Martino 84,Boyer 87].
- l'introduction de la syntaxe comme contrainte supplémentaire à l'intérieur d'un mot [Boyer 87] et au niveau de la phrase [Ney 87,Bourlard 85,Nakagawa 87].

3.2 Approche probabiliste

Cette approche consiste à mémoriser, sous forme de probabilité de transition d'état ou de coefficients de pondération éventuellement, tous les phénomènes produits par toutes les séquences de combinaison des symboles pouvant avoir lieu en reconnaissance [Bahl 83, Sakoe 71,Myers 81,Lowerre 76,Baker 75a,Kopec 85]. Le progrès de cette approche par rapport à l'approche de la programmation dynamique est que la connaissance statistique a priori sur la propriété de la parole est explicitement modélisée. L'information contextuelle est représentée stochastiquement par des probabilités de transition. Beaucoup de techniques de cette approche permettent d'avoir un apprentissage automatisé du système de reconnaissance pour s'adapter à une nouvelle application [Rabiner 77,Levinson 83b]. Si l'apprentissage du système est convenable et suffisant, le taux de reconnaissance peut être très élevé. En général, pour que l'apprentissage soit bien fait sur un seul locuteur, environ 20 minutes de parole doivent être traitées [Hoskins 85].

3.3 Approche cognitive (I.A.)

L'objectif de l'interprétation de la parole est de reconnaître la parole continue de façon indépendante du locuteur sur un vocabulaire important. Ce qui nécessite:

- la modélisation des connaissances de tous les niveaux afin de pouvoir traiter la variabilité correctement et efficacement,
- l'introduction d'une stratégie spécifique à chaque niveau d'interprétation,
- l'utilisation immédiate de l'information sur la parole en cours d'analyse dans le processus d'interprétation.

L'approche cognitive considère que de différentes sources de connaissances intervenant dans la production et la perception peuvent être modélisées et réutilisées explicitement [Zue 85,Klatt 77,Steven 80,DeMori 85a,Lesser 77,Pierrel 81]. Les travaux dans cette approche consacrent donc des efforts sur la modélisation du processus de l'interprétation de la parole par un modèle fonctionnel. Or, les deux approches présentées précédemment sont des modèles comportementaux, et donc insuffisants. Beaucoup de propositions dans cette voie existent, mais pas de théorie générale [Lesser 77,Erman 80,Pierrel 87,DeMori 85a].

Par ailleurs, les modèles précédents disposent d'une stratégie de recherche simple qui ne consiste qu'à énumérer toutes les solutions possibles, un ensemble fini, connues a priori par le système, et retenir la meilleure suivant un critère. Lorsque le nombre d'éléments dans cet ensemble devient grand, ces modèles ne sont plus raisonnables, sinon inopérants, même si des stratégies heuristiques de réduction de calcul sont utilisées. Dans ce cas une

utilisation de la structure plus profonde du signal est necessaire. L'approche I.A. permet d'utiliser des connaissances diverses et des heuristiques dans la perception humaine afin de construire l'interprétation finale en réduisant l'espace de recherche des solutions. A l'inverse des autres approches, les connaissances, les données (interprétation partielle) et la structure de contrôle sont séparées explicitement. Ceci constitue la caractéristique essentiel de la stratégie de solution de l'approche I.A.

Contrairement aux approches précédentes, l'I.A permet de trouver des résultats d'interprétation en faisant une recherche à profondeur variable. Le signal est examiné sur l'axe du temps à des niveaux de détails differents, selon le besoin d'information du processus d'interprétation. Cette propriété donne la possibilité de réduire le calcul et permet, sous contrainte que le temps de calcul soit le même, de rechercher les solutions en utilisant des analyses plus fines.

3.4 Conclusion

La difficulté du problème d'interprétation de la parole est dûe à la variabilité de la parole. L'extraction de l'information commune aux prononciations différentes d'une même unité linguistique reste encore mal connue. Actuellement, les modèles stochastiques *comportementaux* sont largement utilisés pour pallier cette ignorance en mémorisant toutes les variations phonologiques, dans différents contextes de toutes les unités de reconnaissance. Ces modèles peuvent effectivement capter et incorporer la variabilité de la parole, mais, par nature, ils cachent le mécanisme de la production de cette variabilité. Cette technique augmente le nombre de comparaisons nécessaires à la reconnaissance et, lorsque le nombre de locuteurs ou la taille du vocabulaire est important, conduit à une très faible distinction entre les unités de reconnaissance et donc à limiter la performance du système.

Les problèmes pour les techniques non-cognitives sont principalement:

- Le processus de reconnaissance étant fondé sur un modèle mathématique très simplifié, il est impossible d'inclure les connaissances qui expriment comment le locuteur produit le signal de parole ou comment l'auditeur percepte la parole.

- La stratégie utilisée est la recherche exhaustive (recherche en faisceau éventuellement) dans tout l'espace de solutions et donc, théoriquement, cette approche est très lourde. Le calcul de distance entre une forme de test et l'ensemble des formes de référence est en général inévitable. La récompense de ce type de recherche est que la meilleure solution est toujours garantie.

- Il est difficile d'inclure des stratégies de recherche heuristique telles que l'utilisation des îlots de confiance.

- En général, l'algorithme ne fournit qu'une meilleure solution. La deuxième solution, avec éventuellement un score de reconnaissance immédiatement inférieur à celle de la meilleure, est ignorée complétement [Woods 82]. Ce problème est essentiel dans tous les systèmes fondés sur une dissimilarité cumulée. La proposition d'une deuxième solution nécessite presque tous les calculs pour la première solution.

- Une approche purement de reconnaissance et non d'interprétation. L'objectif final est de donner une séquence d'unités de base parmi l'ensemble de séquences acceptables, celle-ci permettamt l'observation de la parole prononcée et minimisant (maximisant) une mesure de similarité (divergence) cumulative entre la séquence reconnue et le signal d'entrée. L'utilisation de contraintes sémantiques et pragmatiques est difficile.

- Le critère d'une distance cumulée est trop simpliste pour mesurer la qualité de reconnaissance. En effet, l'homme attache de différentes importances sur différentes parties d'une phrase, selon des connaissances de nature complexe.

- La base de décision étant le résultat de comparaison, le processus de reconnaissance ne tient aucun compte de la structure et des caractéristiques propres de la parole. Par exemple, en général, les contraintes phonétiques et phonologiques ne sont pas explicitement inclues. Seule la forme acoustique est considérée.

L'approche cognitive tente de résoudre le problème de l'interprétation de la parole par la modélisation *fonctionelle* du processus de la production, de la perception et de la compréhension de la parole. Cette approche conduit à une solution finale et, donc, est prometteuse. Cependant, on sait très peu actuellement sur ce qui se passe réellement à l'intérieur du processus. La plupart des méthodes développées sur cette voie sont ainsi basées sur des hypothèses non vérifiables et parfois loin être vraies.

4 Représentation du signal

4.1 Introduction

L'objectif de la reconnaissance automatique de la parole est de convertir le signal de parole en une représentation linguistique structurée des messages émis. La représentation paramétrique de la parole est une composante dans le système de reconnaissance et une liaison directe du système avec le monde réel. L'entrée de la partie de représentation paramétrique est le signal acoustique de la parole. La sortie est une suite de vecteurs dont la valeur change en fonction du temps. La réalisation de cette représentation est assurée par un système de traitement du signal. Cette partie du système sert à produire une représentation vectorielle non structurée de la parole utilisable par des niveaux supérieurs. Cette composante est considérée comme le sous-système de plus bas niveau d'abstraction de l'information traitée.

L'objectif initial de la représentation est de caractériser une zone de signal de parole par un vecteur de dimension réduite qui, pour une unité symbolique telle que le phonème, ne varie pas en fonction du locuteur. On constate qu'il est impossible de réaliser un tel système uniquement basé sur l'information locale. Il n'existe en fait pas de méthode systématique permettant de déterminer un tel meilleur vecteur et le choix nécessite l'ingéniosité humaine [Chen 87].

Nous exigeons plutôt que le rôle essentiel de cette représentation soit:

- La réduction de données redondantes sans dégrader le contenu du message;

- L'extraction d'un vecteur caractéristique, discriminant et fidèle qui distingue différents phénomènes dans la parole;

On désire que le vecteur de représentation vérifie que:

- l'espace de représentation soit bijectif avec l'espace original du signal de la parole, c'est à dire que le vecteur puisse être utilisé pour reproduire le signal
- les composantes soient orthogonales
- les normes des composantes soient non-corrélées
- le vecteur ait un sens physique
- le calcul puisse s'effectuer dans un temps raisonnable.

Le but final d'une représentation est de permettre la reconnaissance. Ainsi, pour chaque représentation il faut définir une mesure de dissimilarité entre deux formes. Chaque représentation doit s'associer à une mesure de similarité spécifique [Nocerino 85]. La distance Euclidienne est convenable pour mesurer la dissimilarité sous une représentation dérivée d'une base orthogonale, mais pour certaines représentations, la dissimilarité obtenue ne correspond pas du tout à celle estimée par la perception humaine.

4.2 Les paramètres

En reconnaissance de la parole, le banc de filtres linéaires, la transformée de Fourier discrète (DFT), la prédiction linéaire (LPC) et l'analyse cepstrale sont souvent utilisées [Rabiner 75a]. Des études sont réalisées pour comparer la performance de ces techniques dans différents systèmes de reconnaissance. Plus généralement, la comparaison théorique de l'efficacité des critères de la sélection d'une représentation a été extensivement étudiée et on peut se référer aux articles de Kailath [Kailath 67], Fu [Fu 70] et Devijver [Devijver 74].

Nous commentons brièvement les différentes méthodes d'analyse [Flanagan 72,Rabiner 75a, Rabiner 78b,Schafer 79].

- Banc de filtres: On calcule l'énergie moyenne des composantes fréquentielles du signal dans certaines bandes de fréquence. S'il sont bien conçus, les filtre peuvent conserver mieux l'information sur les transitions du signal que les méthodes à fenêtre d'analyse.
- Transformée de Fourier: Dans cette représentation, l'information sur le signal original est complète. Si le signal est stationnaire, on peut reconstituer exactement le signal à partir de sa représentation. Selon la largeur de la fenêtre d'analyse, le spectre du signal obtenu par la transformée de Fourier contient des harmoniques dûes à la source de l'excitation de la parole qui peuvent perturber l'identification de l'information relativement indépendante de locuteur. Par exemple, l'identification des formants est difficile par cette représentation.
- Analyse homomorphique: C'est une technique permettant de séparer les informations de l'excitation et du conduit vocal.

- Analyse par prédiction linéaire: Dans cette analyse, le nombre de formants exprimables est prédéterminé. Si p est correct, le nombre, la largeur de bande et la position des formants sont très précis. La prédiction linéaire représente mal les creux dans le spectre que des sons nasalisés possèdent généralement.

- Les modèles pôle-zéro (ARMA) [Atal 78.Song 83.Yegnanarayana 81,Morikawa 82,Steiglitz 77, Linggard 82]: Ces modèles apportent de la précision sur la modélisation du spectre du signal. En particulier, le signal de parole bruitée et les sons nasalisés sont mieux représentés par les modèles à pôles et zéros. Nous pensons que, du point de vue de la précision de l'analyse, les techniques précedentes sont déjà suffisantes. Les modèles à pôle-zéro ne pourront pas améliorer radicalement la reconnaissance. Cet argument se justifie par l'expérience de l'introduction de la quantification vectorielle, qui représente des vecteurs indénombrables dans l'espace R^n par quelques centaines de vecteurs représentatifs. Cette modélisation de la parole ne diminue pas sensiblement le taux de reconnaissance.

- Modèles non-stationnaires AR [Hall 83] ou ARMA [Grenier 83]: On tente de modéliser le signal de la parole avec une grande précision, même dans le cas où le signal n'est pas stationnaire. Des travaux restent à faire pour utiliser les coefficients de ces modèles en reconnaissance de la parole, notamment pour la comparaison des formes.

Afin de réduire la dimension du vecteur et de conserver l'information discriminante, des transformations mathématiques peuvent être appliquées. Parmi ces transformations, on peut citer la transformée de Karhunen-Loeve, la transformée de Foley-Sammon [Foley 75], et la transformation non linéaire de l'espace caractéristique [Young 74]. L'inconvénient de ces transformations est que le vecteur résultant ne comporte plus de sens physique. Peu de systèmes de reconnaissance de la parole utilisent réellement ce type de transformation.

4.3 Conclusion

Comme dans n'importe quel système de reconnaissance de formes, un système d'extraction des paramètres est indispensable pour la reconnaissance automatique de la parole. L'information perdue à ce niveau de traitement est irrécupérable aux niveaux supérieurs et risque par conséquent de provoquer des erreurs de reconnaissance, Quelle est la meilleure représentation paramétrique pour la parole? On ne sait pas. Les cinq conditions énumérées dans l'introduction ne sont pas toutes vérifiées en général. On essaie de trouver une solution et non la solution. Actuellement, les représentations basées sur des notions spectrales (obtenues par T.F.D, LPC, cepstre, etc) sont les plus utilisées. Ces représentations semblent mieux adaptées car certains comportements du système auditif humain peuvent être expliqués par l'analyse spectrale.

5 Un Modèle de phonème à profils du signal

5.1 Introduction

En interprétation du signal, le raisonnement doit se faire sur une représentation symbolique du signal de la parole. Il est donc nécessaire de définir un convertisseur signal-symbole. Nous avons indiqué dans les chapitres 4 et 5 qu'une telle conversion est difficile. Nous n'avons pas l'ambition de réaliser un convertisseur signal-symbole sans erreur, car nous pensons que, localement, le signal est incertain et donc un tel convertisseur n'est possible que pour des cas particuliers. Nous demandons à la conversion de fournir à chaque instant une liste de propositions avec des coefficients de vraisemblance associés. Ces coefficients sont aussi obtenus en fonction des mesures locales et, par conséquent, n'indiquent qu'un jugement du niveau de conversion.

5.2 Unité de reconnaissance

La parole est observée sous forme d'une variation continue. Or, pour réaliser la communication, certaines catégories perceptuelles correspondant à un vocabulaire devraient être transmises. Autrment dit, le signal doit être identifié en une suite finie d'élements discrets. Actuellement la "taille" de ces élements et la manière dont ils interviennent à la perception ne sont pas encore connues [Flanagan 72]. Dans l'objectif de l'interprétation de la parole continue, nous allons analyser différentes unités élémentaires et nous ferons le choix de l'unité pour notre système.

L'unité **mot** vient directement des travaux d'analyse du texte écrit où un séparateur est obligatoirement présent entre deux mots successifs. En parole continue, un mot n'est pas du tout une unité l'articulation et un tel séparateur n'existe pas. Donc le "mot" est utile pour construire l'idée, mais pas adapté pour être l'unité élémentaire de la reconnaissance. Un autre problème lié au "mot" est la complexité de calcul exigée. Dans un système de reconnaissance de la parole continue à base de mots, lorsque la taille N du vocabulaire à reconnaître dépasse une centaine, l'utilisation du mot comme unité de reconnaissance n'est plus raisonnable du point de vue de la quantité d'information à assimiler à l'apprentissage:

- Si on ignore l'effet de coarticulation, le calcul de l'apprentissage ou de la reconnaissance est déjà proportionel à N.
- Si on admet que la coarticulation est prise en compte par le système alors que ce calcul est proportionel à N^2 dans le cas où seule l'influence contextuelle entre deux mots consécutifs est considérée.

Lorsque le vocabulaire grandit, ce schéma devient rapidement prohibitif car il faut présenter au système plusieurs répétitions toutes les combinaisons de phonèmnes pouvant avoir lieu à la reconnaissance.

Pour réduire les difficultés rencontrées lorsque le mot est utilisé comme la plus petite unité de reconnaissance, l'unité **diphone** a été proposée [Klatt 80]. Un diphone est une unité phonétique composée de la deuxième moitié d'un phonème suivie de la première moitié

du phonème suivant. Chaque diphone est représenté par 3 ou 4 profils spectraux. Ce modèle inclut l'information sur la transition entre les phonèmes, mais présente les inconvénients suivants:

- Le nombre des instances du modèle doit être élevé pour reconnaître toutes les combinaisons phonétiques (environ 2500);
- L'information dans la partie centrale d'un phonème, où l'influence de contexte est relativement petite n'est pas suffisamment utilisée.

On peut encore utiliser comme unités élémentaires de reconnaissance la **syllable** [Hunt 80] ou la **demisyllable** [Rosenberg 83]. L'effet contextuel est dans tous les cas réduit par rapport au mot, mais le nombre de ces unités à mémoriser pour la reconnaissance est encore trop grand – plus de 10000 pour la syllable et plus de 1000 pour la demisyllable. Cela nécessite de plus un effort considérable lors de l'apprentissage.

Dans notre système, nous utilisons des **phonèmes** comme unités élémentaires. L'avantage d'utiliser le phonème comme la plus petite unité de reconnaissance dans un système de reconnaissance de parole se résume par les points suivants:

- conceptuellement, la parole est représentée par une séquence d'unités phonétiques, lien avec la linguistique.
- l'indépendance des tâches; l'apprentissage se fait une fois pour toutes les applications.
- le faible nombre d'unités à reconnaître (< 80).
- la variation phonétique à l'intérieur d'un mot est prévisible par des règles phonologiques. Ces règles sont souvent définies sur des unités phonétiques.
- la facilité d'inclure des traits acoustico-phonétiques, e.g. durée, formants, burst, dans la reconnaissance.
- la possibilité de tenir compte des difficultée de la reconnaissance de phonèmes différents par l'attachement d'une importance différente dans le score total. Ceci permet d'améliorer par exemple la reconnaissance des lettres *e-set* ($= \{b, c, d, e, g, p, t, v, z\}$) en anglais.

5.3 Quelques modèles existants

Pour réaliser la conversion signal-symbole, plusieurs modèles existent, mais nous allons constater que le problème de modélisation n'est pas encore résolu de façon satisfaisante.

Formes de réference consécutives

On modélise l'unité de reconnaissance directement par son image acoustique dans un espace de représentation. La reconnaissance consiste à effectuer la programmation dynamique entre les formes inconnues et les formes de référence. Pour comparer deux formes par programmation dynamique, il est nécessaire de calculer toutes les distances entre chaque profil

de la forme de référence et chaque profil de la forme de test. A la vitesse de déplacement de la fenêtre d'analyse généralement utilisée, pour des formes stationnaires et ayant une longue durée, il y a de la redondance dans le calcul. En revanche, pour des formes qui ont une transition brutale entre les profils successifs, des détails sont insuffisamment inclus.

Modèle de Markov caché

Les modèles de Markov cachés sont actuellement utilisés dans la reconnaissance de la parole à mots isolés, enchaînés et la parole continue. Selon le niveau où les modèles sont définis, ils peuvent être utilisés pour représenter une phrase, un mot, un phonème [Schwartz 85] ou d'autres unités de la parole [Bahl 83].

- Ces modèles présentent un grand nombre d'avantages:
  - Base mathématique solide pour comprendre le fonctionnement.
  - Bon modèle de la variabilité de la parole.
  - Le recalage temporel incorporé systématiquement.
  - Prise en compte de l'ordre d'apparition des évènements dans la séquence de vecteurs.
  - Optimisation d'un critère de performance; e.g. minimisation de la probabilité d'erreurs.
  - Existence des méthodes d'apprentissage automatique et d'adaptation automatique au locuteur. Etant donnés le signal de la parole segmentée et la séquence de symboles correspondant aux segments, l'algorithme d'apprentissage de Baum [Baum 72] ajuste les paramètres des modèles afin d'augmenter la probabilité que chaque modèle produise la donnée associée.
  - Reconnaissance réalisée par un simple calcul de la probabilité cumulée. L'algorithme de *Viterbi* [Jr 73] permet de la calculer efficacement. Cet algorithme peut se formuler en terme de la programmation dynamique [Bourlard 85],
  - Séparation nette entre données et algorithme.
  - Décision globale sans obligation d'utiliser des seuils.
- Parmi les inconvénients:
  - Ignorance complète de la durée relative des évènements dans le signal; contraintes faibles sur la structure temporelle du signal.
  - Supposition de la dépendance du premier ordre pas toujours vérifiée.
  - Dégradation de performance si l'apprentissage n'est pas suffisant.
  - Le choix a priori de la topologie des modèles (le nombre d'états, les transitions autorisées et les règles de transition) limite la souplesse des modèles.
  - Difficulté d'inclure des connaissances explicites sur la production et la perception de la parole.
  - Ce n'est qu'un modèle comportemental et non fonctionel. Il n'y a aucun lien explicite du modèle à la production et la perception de la parole.

Machines connexionistes

Une machine connexioniste [Hinton 84,Rumelhart 86] est un ensemble d'unités de traitement que l'on appelle neurones connectées par une fonction spécifiée par des coefficients de pondération. Ces machines sont aussi appelées les réseaux neuro-mimétiques. L'intéret des machines connexionistes en reconnaissance phonémique de la parole est lié à leur capacité d'acquérir de la connaissance sur la relation de conversion signal-symbole par un aprentissage automatique et de reconnaître des formes similaires à celles présentées dans l'ensemble d'exemples d'apprentissage. En ajustant les coefficients, ces machines peuvent mémoriser et généraliser des contraintes sur la structure des formes d'ordre élevé en n'imposant aucune hypothèse sur les formes à reconnaître. La procédure d'apprentissage [Rumelhart 86] consiste simplement à minimiser une erreur quadratique globale au niveau de la machine dans l'espace des coefficients de pondération de façon que la relation entrée-sortie définie par l'ensemble des exemples d'apprentissage soit maintenue au mieux. Les machines connexionistes sont appliquées en synthèse de fonctions booléennes, jeux, reconnaissance des caractères, système expert [Soulie 86], synthèse texte-parole [Sejnowsky 86] et en reconnaissance de la parole [Prager 86].

Les *perceptrons à niveaux multiples* (Multilayer Perceptrons, MLP) constituent une classe de machines connexionistes. Un MLP réalise tout simplemenmt une application d'une forme d'entrée à n_i-dimensions vers n_o symboles de sortie, à travers plusieurs niveaux intermédiaires. Chaque niveau se compose d'un certain nombre de neurones. La sortie d'un neurone est une fonction continue non-linéaire, souvent sous la forme

$$y = \frac{1}{1+e^{-x}},$$

de la combinaison linéaire des sorties des neurones du niveau inférieur. Les connexions à l'intérieur du même niveau, venant d'un niveau supérieur ou entre des niveaux non successifs sont interdites.

A la reconnaissance, la forme inconnue est présentée aux neurones du plus bas niveau. Les sorties des neurones du niveau le plus supérieur peuvent être calculées en une seule passe.

Il existe des algorithmes d'apprentissage pour obtenir automatiquement les coefficients du réseau. On utilise souvent l'algorithme de rétro-propagation des erreur *error back propagation algorithm* [Rumelhart 86,Sejnowsky 86], qui est itératif et consiste à minimiser une erreur quadratique:

- présenter des vecteurs d'entrée et calculer le vecteur de sortie;
- propager l'erreur de reconnaissance et corriger les coefficients de pondération. La correction peut être réalisée par la méthode du gradient descendant [Peeling 86].

Il est démontré [Minsky 69] que grâce à l'introduction des niveaux intermédiaires et de la fonction non-linéaire, à la sortie de chaque neurone, un MLP peut réaliser n'importe quelle sorte d'application non-linéaire entre son entrée et sa sortie à condition que le nombre de niveaux soit suffisant. Cette propriété distingue les MLP des perceptrons à deux niveaux proposés et étudiés dans les années 60 [Minsky 69], où il est impossible de trouver une

configuration des coefficients de pondération pour produire certaines relations de projection entrée-sortie [Hinton 85], et assure la puissance de ce modèle.

Les modèles à base de "mémoire-comparaison" obéissent à une relation entre le nombre de formes stockées et le temps de recherche. Ceci limite l'application en temps-réel de ces modèles. Un grand avantage de MLP au niveau de l'efficacité du calcul, par rapport à ces modèles est que la vitesse de la classification de forme, qui ne dépend que de la complexité du réseau, est indépendante du nombre de formes utilisées à l'apprentissage.

Les MLP ont été utilisés en reconnaissance automatique de la parole. Les résultats obtenus récemment pour la reconnaissance monolocuteur sont impressionants [Bourlard 87, Peeling 86].

Cependant, les machines connexionnistes ne sont que des mémoires associatives, concises et suffisamment performantes pour stocker les relations complexes entre deux domaines. En effet, on a déjà constaté que si le corpus d'apprentissage contient trop de formes ou si le nombre des neurones au niveau d'entrée ou de sortie est très grand, la précision de classification d'un tel système se dégrade [Kosko 87]. Beaucoup de travail reste à faire afin que ces modèles puissent traiter convenablement à la fois la déformation des formes à cause de l'influence contextuelle et le recalage temporel – un problème majeur en reconnaissance de la parole.

Conclusion

Les deux premières techniques effectuent un recalage temporel de façon que la différence entre la forme de test et la forme de référence soit minimisée. Cependant en même temps, la proportion de durée des évènements phonétiques n'est pas conservée. La dernière technique a des difficultés à tenir compte en même temps de l'ordre des évènements du signal et du recalage temporel. Or, du point de vue phonétique, les durées relatives des phonèmes portent des traits informatifs et peuvent aider à discriminer certains phonèmes [Vaissiere 83]. D'où l'inconvénient commun que les proportions temporelles des formes qui aident à la distinction des unités différentes sont supprimées.

5.4 Un modèle des phonèmes

Introduction

Nous proposons un modèle comportemental des phonèmes et les algorithmes d'exploitation associés.

On peut imaginer une équivalence de la perception de l'oreille humaine comme étant un ensemble de cellules présentées en parallèle à l'entrée du signal. Chaque cellule est spécialisée dans la capture d'une séquence ou d'une forme d'excitation phonatoire particulière. Excitée par le signal, chaque cellule produit une réponse de valeur réelle qui est d'autant plus grande que la variation du signal est plus proche de la forme spécialisée. Pour connaître le signal à chaque instant nous mesurons et trions selon la valeur de réponse à l'excitation de façon décroissante les sorties de ces cellules. La probabilité que la variation du signal appartienne à la forme de variation représentée par une cellule est donc d'autant plus grande que la cellule

est proche de la tête de la liste de sortie triée.

L'excitation des cellules peut être une représentation paramétrique du signal telle que le spectre ou les coefficients de prédiction linéaire. La spécialisation des cellules peut être automatisée.

Nous donnons comme résultats de classification phonémique les étiquettes et les probabilités associées d'un certain nombre de cellules, celles qui fournissent les plus grandes réponses à l'excitation. Le nombre dépend de l'incertitude du signal à chaque instant. Il n'est pas raisonnable d'espérer une fiabilité des résultats très élevée parce que

- l'homme ne fait pas de décisions séparées sur chaque phonème dans le flux de parole [Flanagan 72];
- l'effet de l'influence contextuel sur les phonèmes est non négligeable;
- l'unité phonème est trop fragile pour que la décision soit fiable.

Hypothèses et modèle

Nos hypothèses sur le signal de la parole sont

- En tenant compte des phénomènes phonologiques, l'image, ou la réalisation d'un phonème, est mesurable dans le signal de la parole;
- La variation paramétrique non-linéaire à l'intérieur d'une image de phonème est utile pour discriminer des phonèmes différents et doit être conservée lors de la comparaison entre le signal et les références;
- Pour comparer des formes de longueur différente, un recalage linéaire peut être utilisé;
- L'influence contextuelle peut être modélisée par des références multiples pour chaque symbole de phonème.

Notre modèle phonémique pour un phonème se compose de 2 à 5 profils du signal, obtenus par un échantillonnage uniforme de la séquence de vecteurs paramétriques correspondant à l'image du phonème dans le signal de la parole (c.f. figure 8.1). La durée du temps occupée par un modèle est de 10 ms (pour certains plosives) à 140 ms (pour certaines diphtongues). La moyenne et l'écart type de la durée sont aussi inclues. Ce modèle contient non seulement l'information sur la répartition de spectre en fonction du temps, mais aussi l'information sur la position et la durée de chaque évènement de la suite d'évènements de la parole dans l'image du phonème. Notre modèle inclut la corrélation entre les vecteurs paramétriques adjacents. Cette corrélation a permis d'améliorer le score de reconnaissance dans de différents travaux [Bocchieri 86,Furui 86].

Mesure de similarité

La similarité locale que nous utilisons est le rapport de vraisemblance entre le vecteur de test et le vecteur de référence. La similarité entre le signal de parole et la séquence de profils

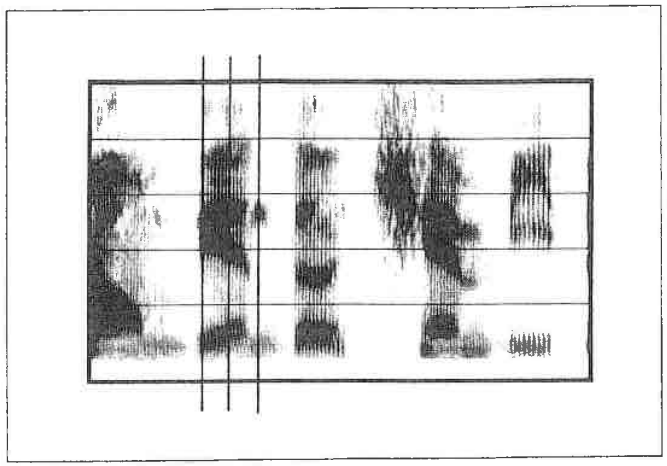


Figure 8.1: exemple d'extraction des profils pour un son

de référence à un instant donné, est obtenue par une optimisation de la somme des similarités locales, tout en respectant l'écart temporel de chaque profil dans la référence du phonème. L'optimisation est faite en faisant varier la durée du phonème de la valeur minimale à la valeur maximale définies par la référence.

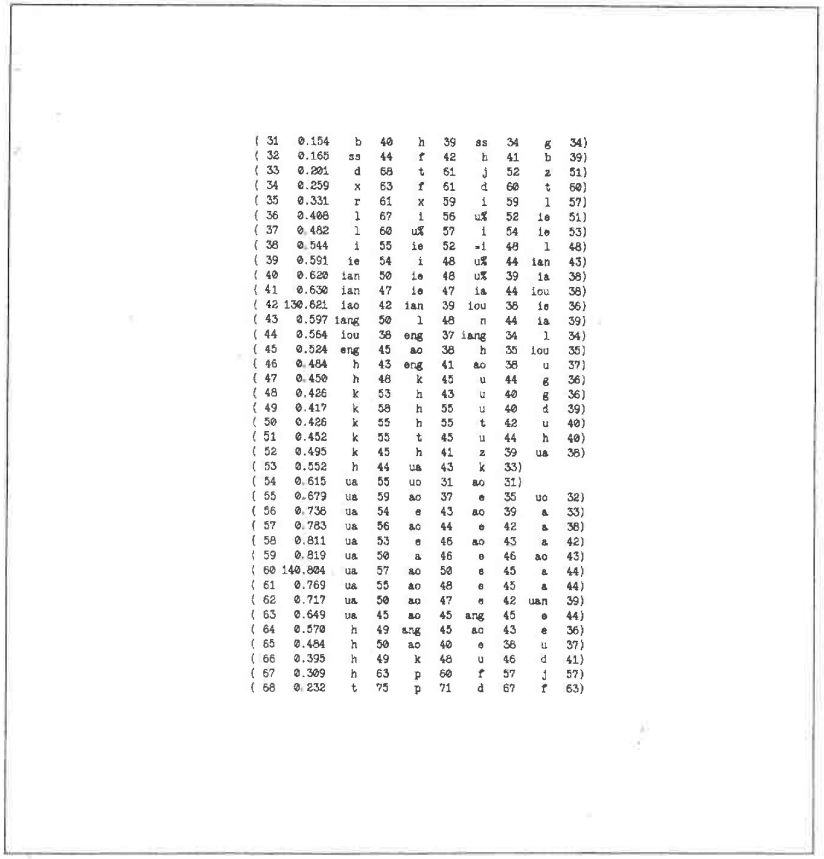
A chaque instant, si la similarité du signal à un modèle de référence est grande, on obtient à ce moment une valeur proche de 1, et si la similarité est très faible, on obtient une valeur proche de 0. Un modèle peut donc être considéré comme un filtre du signal qui accepte la suite de vecteurs paramétriques et qui donne en sortie une fonction temporelle indiquant la vraisemblance du signal à une référence donnée.

A la reconnaissance, le signal est présenté en parallèle devant un banc de tels filtres, chacun identifiant un phonème. Nous trions ensuite, à chaque instant, les sorties de ces filtres et sélectionnons les meilleures comme candidats de la reconnaissance.

Expansion du rapport de vraisemblance

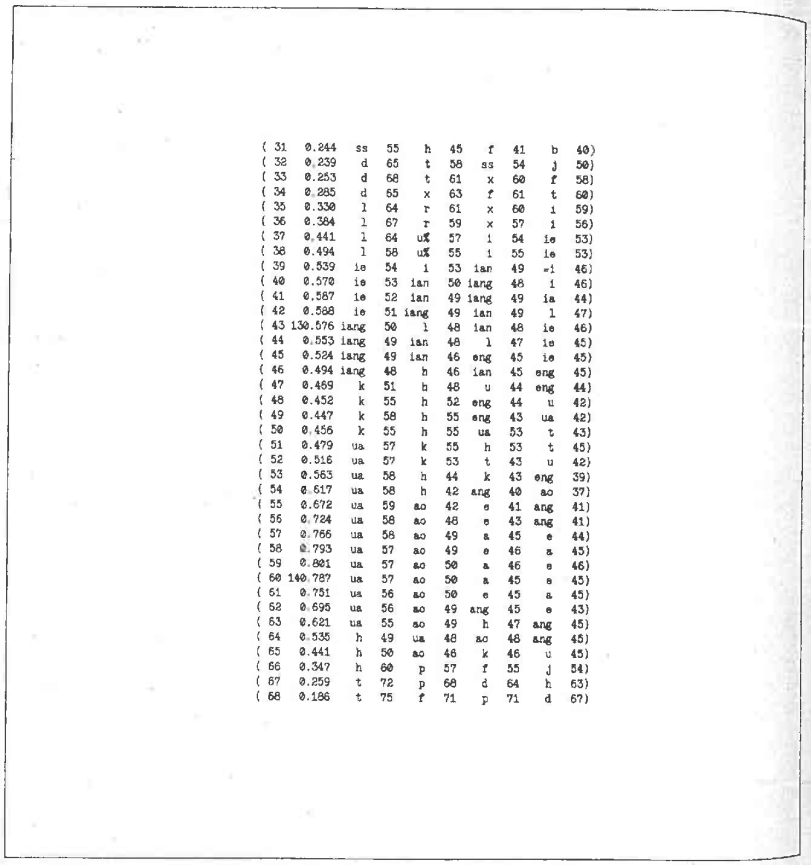
Si on compare une référence de phonème avec le signal de la parole duquel elle est extraite, on constate qu'à la sortie du filtre, à l'instant correspondant au centre de la référence, il y a un point qui est exactement égal à 1. Cette valeur signifie que le signal à l'instant est identique à la référence. En fait, non seulement à cet instant le filtre doit fournir la valeur 1 mais au voisinage aussi, car du début jusqu'à la fin de la zone où la référence a été prise, le signal est étiqueté par le même symbole phonémique. Nous tenons compte de ce fait par l'expansion du rapport de vraisemblance dans la sortie du filtre du centre jusqu'aux deux extrêmités de la référence, définies par le processus de mesure de similarité. La figure 8.2

et la figure 8.3 montrent respectivement le résultat de reconnaissance de phonèmes sans ou avec expansion.



(31 0.154 b 40 h 39 ss 34 g 34)
(32 0.165 ss 44 f 42 h 41 b 39)
(33 0.201 d 68 t 61 j 52 z 51)
(34 0.259 x 63 f 61 d 60 t 60)
(35 0.331 r 61 x 59 i 59 l 57)
(36 0.406 l 67 i 56 u% 52 ie 51)
(37 0.462 l 60 u% 57 i 54 ie 53)
(38 0.544 i 55 ie 52 -i 48 l 48)
(39 0.591 ie 54 i 48 u% 44 ian 43)
(40 0.620 ian 50 ie 48 u% 39 ia 38)
(41 0.638 ian 47 ie 47 ia 44 iou 38)
(42 130.621 iao 42 ian 39 iou 36 ie 36)
(43 0.597 iang 50 l 48 n 44 ia 39)
(44 0.564 iou 38 eng 37 iang 34 l 34)
(45 0.524 eng 45 ao 38 h 35 iou 35)
(46 0.484 h 43 eng 41 ao 38 u 37)
(47 0.450 h 46 k 45 u 44 g 36)
(48 0.426 k 53 h 43 u 40 g 36)
(49 0.417 k 58 h 55 u 40 d 39)
(50 0.426 k 55 h 55 t 42 u 40)
(51 0.452 k 55 t 45 u 44 h 40)
(52 0.495 k 45 h 41 z 39 ua 38)
(53 0.552 h 44 ua 43 k 33)
(54 0.615 ua 55 uo 31 ao 31)
(55 0.679 ua 59 ao 37 e 35 uo 32)
(56 0.736 ua 54 e 43 ao 39 a 33)
(57 0.783 ua 56 ao 44 e 42 a 38)
(58 0.811 ua 53 e 46 ao 43 a 42)
(59 0.819 ua 50 a 46 e 46 ao 43)
(60 140.804 ua 57 ao 50 e 45 a 44)
(61 0.769 ua 55 ao 48 e 45 a 44)
(62 0.717 ua 50 ao 47 e 42 uan 39)
(63 0.649 ua 45 ao 45 ang 45 e 44)
(64 0.570 h 49 ang 45 ao 43 e 36)
(65 0.484 h 50 ao 40 e 36 u 37)
(66 0.395 h 49 k 48 u 46 d 41)
(67 0.309 h 63 p 60 f 57 j 57)
(68 0.232 t 75 p 71 d 67 f 63)

Figure 8.2: le résultat de classification sans expansion



```
( 31   0.244   ss   55    h   45    f   41    b   40)
( 32   0.239    d   65    t   58   ss   54    j   50)
( 33   0.253    d   68    t   61    x   60    f   58)
( 34   0.285    d   65    x   63    f   61    t   60)
( 35   0.330    l   64    r   61    x   60    i   59)
( 36   0.364    l   67    r   59    x   57    i   56)
( 37   0.441    l   64   u%   57    i   54   ie   53)
( 38   0.494    l   58   u%   55    i   55   ie   53)
( 39   0.539   ie   54    i   53  ian   49   -i   46)
( 40   0.570   ie   53  ian   50 iang   48    i   46)
( 41   0.587   ie   52  ian   49 iang   49   ia   44)
( 42   0.588   ie   51 iang   49  ian   49    l   47)
( 43 130.576 iang   50    l   48  ian   48   ie   46)
( 44   0.553 iang   49  ian   48    l   47   ie   45)
( 45   0.524 iang   49  ian   46  eng   45   ie   45)
( 46   0.494 iang   48    h   46  ian   45  eng   45)
( 47   0.469    k   51    h   48    u   44  eng   44)
( 48   0.452    k   55    h   52  eng   44    u   42)
( 49   0.447    k   58    h   55  eng   43   ua   42)
( 50   0.456    k   55    h   55   ua   53    t   43)
( 51   0.479   ua   57    k   55    h   53    t   45)
( 52   0.516   ua   57    k   53    t   43    u   42)
( 53   0.563   ua   58    h   44    k   43  eng   39)
( 54   0.617   ua   58    h   42  ang   40   ao   37)
( 55   0.672   ua   59   ao   42    e   41  ang   41)
( 56   0.724   ua   58   ao   48    e   43  ang   41)
( 57   0.766   ua   58   ao   49    a   45    e   44)
( 58   0.793   ua   57   ao   49    e   46    a   45)
( 59   0.801   ua   57   ao   50    a   46    e   46)
( 60 140.787   ua   57   ao   50    a   45    e   45)
( 61   0.751   ua   56   ao   50    e   45    a   45)
( 62   0.695   ua   56   ao   49  ang   45    e   43)
( 63   0.621   ua   55   ao   49    h   47  ang   45)
( 64   0.535    h   49   ua   48   ao   48  ang   45)
( 65   0.441    h   50   ao   46    k   46    u   45)
( 66   0.347    h   60    p   57    f   55    j   54)
( 67   0.259    t   72    p   68    d   64    h   63)
( 68   0.186    t   75    f   71    p   71    d   67)
```

Figure 8.2: le résultat de classification avec expansion

5.5 Algorithmes

Strcture de données

Une référence de phonème est donnée par le produit cartésien

$$< symbole, longueur, pourcent, nombre, profil >$$

dont nous précisons chaque composante:

symbole: Le nom donné au phonème. Il peut avoir plusieurs références ayant le même symbole.

longueur: Le nombre de profils (ou le nombre de prélèvements de l'analyse paramétrique) représenté par la référence. Il définit la durée de la référence.

pourcent: Le pourcentage de variation de durée autorisé. Cette variation dépend du phonème et peut valoir jusqu'à $\pm$ 50%.

nombre: Le nombre de profils réellement utilisés pour caractériser le phonème. Typiquement la valeur est comprise entre 2 et 5.

profil: Les vecteurs de paramètres échantillonnés à partir des prélèvements d'analyse du phonème.

Algorithmes

A la reconnaissance, le signal de la parole est soumis en parallèle à un banc de filtres de référence. Les sorties de filtres $LR_{n,i}$, exprimées par le rapport de vraisemblance entre la phrase inconnue S à l'instant n, S_n, et la référence i et normalisées au nombre de profils dans la référence, sont triées et les meilleures étiquettes phonétiques associées à leur vraisemblance sont stockées sous forme de treillis de phonèmes pour un traitement ultérieur. Dans la totalisation des rapports de vraisemblance de chaque profil, un recalage linéaire est utilisé. Ce type de recalage est reconnu comme adéquat pour des unités phonémiques [Vaissiere 83].

$LR_{n,i}$ est calculé en deux phases. Pendant la première phase, nous calculons le rapport de vraisemblance entre le signal à chaque instant et chaque référence par la formule suivante:

$$LR'_{n,i} = \frac{1}{M_i} \max_{dmin_i \leq d \leq dmax_i} L(S_{n-d/2}, P_i, M_i, d) \qquad 0 \leq n < N, 1 \leq i \leq N_r$$

et

$$L(S_k, p, M, d) = \sum_{j=0}^{M} l(S_{k+jd/M}, p_j)$$

où M_i est le nombre de profils dans la référence i, $dmin_i$ et $dmax_i$ sont le minimum et le maximum de durée du phonème représenté, les P_i sont les profils échantillonnés du phonème, $l(v_1, v_2)$ est le rapport de vraisemblance de v_1 à v_2, N est le nombre de profils constituant la phrase prononcée et N_r est le nombre de références dans le banc de filtres.

Dans la seconde phase, nous faisons une expansion de rapport de vraisemblance $LR'_{n,i}$ à chaque instant n, vers la gauche et la droite. La largeur de l'expansion est la durée optimale obtenue par le recalage temporel de référence:

$$LR_{n,i} = \max_{k=-Dopt_{n,i}/2}^{Dopt_{n,i}/2} LR'_{n+k,i} \times W(k, Dopt_{n,i}) \quad 0 \leq n < N, 0 \leq i < N_r$$

où $Dopt_{n,i}$ est la durée optimale de la référence i pour obtenir la meilleure vraisemblance à l'instant n et la fonction de pondération $W(k,d) \rightarrow [0,1]$ est définie par la formule suivante.

$$W(k,d) = 1 - |k| \times \frac{1.0 - MINV}{d/2} \quad |k| \leq d/2$$

où MINV est une constante entre [0,1]. La constante MINV permet d'indiquer le centre du point de départ de l'expension et prend une valeur typique de 0.7.

Les deux phases de calcul sont mêlées lors de l'implantation de l'algorithme pour des raisons de stokage et d'efficacité. Les distances entre les profils de la phrase prononcée et chaque référence sont tabulées afin d'éviter de les recalculer inutilement. Nous résumons la méthode de reconnaissance basée sur notre modèle de phonème par les algorithmes qui suivent.

La fonction Comparaison calcule, pour chaque référence de phonème i et pour chaque profil du signal inconnu à l'instant n, le rapport de vraisemblance $LR_{n,i}$.

TabScore = le tableau de scores de comparaison: TabScoreType

```
Comparaison(R,S,Nr,N) → TabScoreType
    Pour i de 0 à Nr − 1 Faire
        TabLlr = RemplirDistTab(R[i],S,N);
        dvar = Longueur(R[i]) × PourcentVarLong(R[i]);
        dmin = Max(Nombre(R[i]),Longueur(R[i])-dvar);
        dmax = Longueur(R[i]) - dvar;
        Pour n de 0 à N-1 Faire
            (LR',dopt) = CompaRéfSig(TabLlr,dmin,dmax,Nombre(R[i]),n);
            TabScore = Expansion(TabScore,LR',dopt/2,n,i);
        FinPour;
    FinPour;
```

La fonction CompaRéfSig compare une référence avec le signal à l'instant n. Le résultat est le couple formé du meilleur score, sous la forme du rapport de vraisemblance, et de la durée optimale de la référence.

NBINTERV = le nombre qui divise l'intervalle de la variation de durée: constante

```
CompaRéfSig(TabLlr,dmin,dmax,nombre,n) → (ScoreType,DuréeType)
    ChaquePas = Max(1, (dmax-dmin+1)/NBINTERV);
    DuréeOpt = VarDurée = dmin;
    MaxV = 0;
    Tantque VarDurée ≤ dmax Faire
        Somme = 0;
        ChaquePart = VarDurée/nombre;
        Début = n - (VarDurée/2);
        Pour c de 0 à nombre-1 Faire
            Somme += TabLlr[c][Début+c× ChaquePart];
        FinPour;
        Si MaxV < Somme
        Alors MaxV = Somme; DuréeOpt = VarDurée;
        FinSi;
    FinTantque;
    (MaxV/nombre,DuréeOpt);
```

Tab = le tableau de score de comparaison
LR = la valeur du rapport de vraisemblance à l'instant n
d_{opt2} = la moitié de la durée optimale
n = l'instant du signal
i = le numéro de la référence

```
Expansion(Tab,LR,dopt2,n,i)
    Diff = (1.0 - MINV)/dopt2;
    W = 1;
    Tab[n][i] = Max(Tab[n][i],LR);
    Pour k de 1 à dopt2 Faire
        W -= diff;
        LrPondéré = W × LR;
        Tab[n+k][i] = Max(Tab[n+k][i],LR);
        Tab[n-k][i] = Max(Tab[n-k][i],LR);
    FinPour;
```

Apprentissage

L'apprentissage est le processus d'acquisition et d'assimilation des connaissances d'un système. Ce processus réalise un changement adaptatif à l'intérieur du système dont la conséquence est une amélioration de l'efficacité du système lorsque il est présenté avec la même donnée [Simin 83]. Sous ce nom, deux types de problèmes existent: l'apprentissage numérique et l'apprentissage symbolique. L'apprentissage numérique consiste à instancier un ensemble de modèles paramétriques de façon qu'un certain critère soit optimisé. L'apprentissage symbolique consiste entre autre à trouver et à exprimer des relations conceptuelles entre des objets donnés en exemple de façon explicite [Michalski 84,Kodratoff 86]. La différence

fondamentale se trouve dans le type d'instanciation des liens dans le modèle: numérique ou semantique.

Dans le cas de notre modèle de phonèmes, le processus d'apprentissage est du type numérique; il consiste à instancier un modèle pour chaque symbole de phonème. A l'apprentissage, les paramètres des modèles sont ajustés afin de produire au mieux la relation de conversion signal-symbole demandée.

Nous avons fait l'hypothèse que c'est le centre d'une séquence de variations représentées par un modèle de phonème qui contient le plus d'information discriminante. Il est évident que sous cette hypothèse, les réponses à l'excitation deviennent de plus en plus faibles lorsque la variabilité de la parole augmente car les formes dévient de plus en plus de leur centre. Mais nous nous contentons du fait que l'ensemble des modèles phonétiques se comporte de façon proche du comportement de perception humaine, - lorsque la vitesse d'articulation augmente, l'homme a de plus en plus de difficulté pour comprendre.

Cette hypothèse nous permet de faire l'apprentissage des modèles en n'utilisant que les centres des images des phonèmes pris dans un sous-ensemble de combinaisons phonème-phonème.

5.6 Introduction de la quantification vectorielle

Notre modèle reconnaît des phonèmes déformés par le contexte par l'utilisation des références multiples. Lorsque le nombre de références est important, il peut avoir des profils du signal semblables qui sont réutilisés dans l'ensemble de références. Dans l'objectif de la réduction le calcul lors de reconnaissance, nous effectuons une quantification vectorielle sur les références par les étapes suivantes:

- construction d'un *codebook* à partir d'un corpus de parole par une classification non-supervisée;
- codage des références par le codebook;
- création d'un nouveau codebook dans lequel tous les vecteurs qui ne sont pas utilisés pour coder les références sont supprimés;
- les références, sous forme codée, et ce nouveau codebook sont utilisés pour la reconnaissance.

La figure 8.4 montre la circulation d'information de ces étapes.

Cette technique permet de réduire sensiblement la quantité de calcul lors de la création de la matrice de distances locales et fournit une propriété très intéressante: Quelque soit le nombre de références, le calcul de la matrice de distances locales est toujours le même.

Dans un test informel, nous avons fait varier le nombre de codes dans le codebook initial et nous constatons qu'il y a très peu de dégradation de la qualité de reconnaissance par rapport au cas où la quantification vectorielle n'est pas utilisée, lorsque le codebook est de taille 512. Notre système de compréhension de parole donne encore de résultats corrects, même si la taille est diminuée jusqu'à 32.

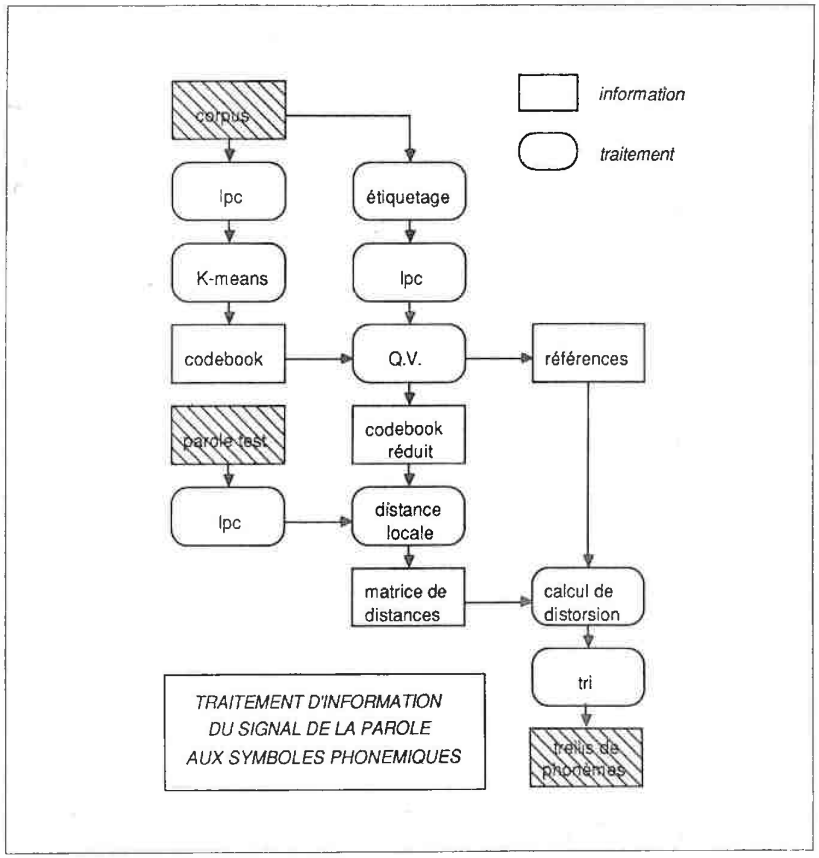


Figure 8.3: Le rôle de la quantification vectorielle dans notre modèle de phonème

5.7 Un exemple

Nous donnons un exemple d'une représentation graphique du résultat du modèle dans le figure 8.5. Dans cette figure, un phonème est d'autant plus bien reconnu à un instant donné que la trace est plus foncée.

350 références ont été introduites pour représenter 52 phonèmes. La durée de la parole de test *ba.3 dian.4 hua.4 fang.4 dao.4 shu.1 de.0 hou.4 bian.1* est de 1.91 secondes. Il a falu 19 secondes de temps de CPU d'un *masscomp* pour réaliser le calcul d'une matrice de distances de taille 544 × 191 et d'autres opérations de la reconnaissance.

5.8 Conclusion

Notre modèle de phonèmes a les caractéristiques suivantes:

- L'utilisation de l'unité phonétique permet un apprentissage pour environ 60 symboles.
- Les informations sur la durée d'un évènement phonémique, ou les contraintes temporelles, sont suffisamment inclues.
- Le fait que dans la parole les profils successifs ne sont pas indépendants est modélisé.
- Un recalage temporel est inclut, ce qui permet de tenir compte des différentes vitesses d'élocution.
- A la fin de la reconnaissance, le début et la fin de la meilleure reconnaissance de chaque unité sont spécifiés et peuvent être utilisés pour la localisation ("spotting") d'un phonème.
- Il est possible d'inclure dans le modèle des traits acoustico-phonétiques.
- Aucun seuil n'est utilisé dans le modèle.
- Le calcul est simple et la reconnaissance est rapide.
- Un grand corpus d'apprentissage n'est pas exigé.
- L'unité élémentaire étant le phonème, l'apprentissage est peu dépendant du texte de corpus.

Il reste encore des travaux à mener sur l'automatisation du processus d'apprentissage et sur la recherche d'une représentation des modèles plus compacte.

6 Reconnaissance sans présegmentation

6.1 Segmentation

Nous définisons la segmentation de la parole comme la partition du signal de la parole suivant l'axe des temps en zones non-superposées de telle manière que chaque zone partitionnée corresponde à une unité de reconnaissance élémentaire linguistiquement significative.

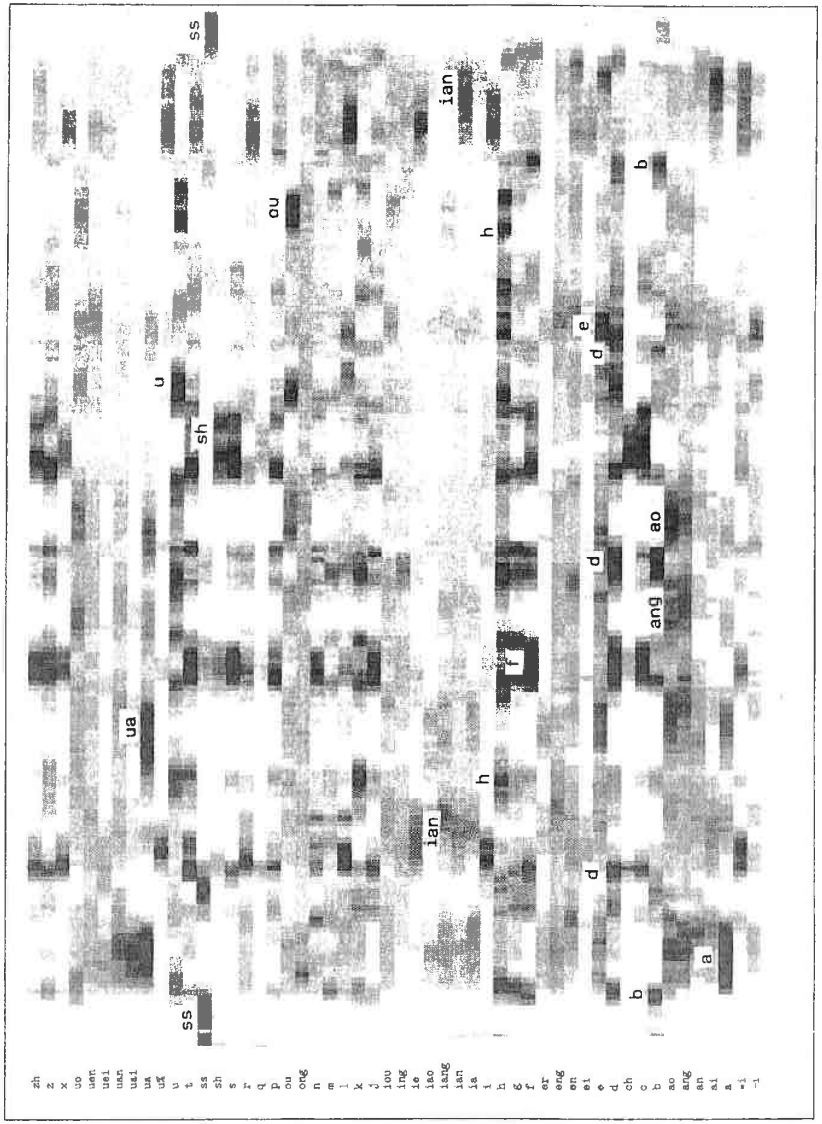


Figure 8.5: le résultat de reconnaissance phonémique de la phrase: ba.3 dian.4 hua.4 fang.4 dao.4 shu.1 de.0 hou.4 bian.1. (*Deplacer le télephone vers derrière le livre.*) Verticalement: temps; horizontalement: symboles phonémiques triés alphabétiquement. 52 phonèmes sont utilisés.

Cette unité peut être choisie parmi les suivantes: phonème, diphone, demi-syllabe, syllabe, mot ou phrase.

La reconnaissance de la parole continue par des méthodes sans présegmentation de la parole en unités de base, telles que la programmation dynamique [Sakoe 71] ou des modèles probabilistes [Bahl 83], a obtenu un taux de reconnaissance très élevé. Cependant, avec ces méthodes il est difficile d'inclure des connaissances explicites de phonétique et de phonologie. Par conséquent, l'apprentissage et la reconnaissance des systèmes basés sur ces méthodes nécessitent un effort de calcul considérable.

D'un autre côté, la reconnaissance de la parole basée sur la segmentation peut efficacement réduire la quantité de données traitées par les niveaux supérieurs mais des erreurs de segmentation inévitables provoquent des insersions et des suppressions de primitives. Ces erreurs imprévisibles obligent l'analyse grammaticale à refuser la phrase en cours d'analyse ou à commettre des erreurs catastophiques au niveau de la reconnaissance.

Récemment, la reconnaissance de la parole continue par des méthodes combinant les deux approches a été étudiée [Brassard 85].

6.2 Difficultés de la segmentation

A cause de l'inertie de l'appareil phonatoire et de la tendance du locuteur à articuler plus librement, la réalisation continue d'une suite de symboles est déformée par rapport à la concaténation des réalisations individuelles de ces symboles. Les images des symboles successifs sont superposées l'une à l'autre. La figure 8.6 illustre cet aspect.

Ce phénomène rend le positionnement d'une marque entre deux symboles successifs extrèmement difficile, sinon impossible.

Le processus humain de production de la parole, qui convertit une séquence de symboles en parole continue est toujours plus compliqué que ce que l'on pense. En parole réelle, il n'existe pas suffisamment d'évidence en bas niveau pour retrouver la correspondance des marges entre les symboles initiaux. Il n'y a pas toujours de silence entre les mots [Mariani 87]. L'intention de segmenter directement (sans passer par des niveaux supérieurs) la parole en unités correspondant aux mots ou aux phonèmes conduit immédiatement à des fausses décisions et entraîne souvent une erreur catastrophique dans l'interprétation finale. Des chercheurs ont constaté que si l'on peut trouver une méthode qui retarde ou évite complétement les décisions de segmentation, la performance du système pourra être améliorée [Klatt 80]. En effet, une segmentation parfaite ne peut être obtenue qu'après avoir identifié toute les unités lexicales. Puisque l'identification des unités lexicales est à son tour dépendante de la reconnaissance de la parole, il est logique qu'une segmentation doive s'effectuer en parallèle avec la reconnaissance de la structure de phrase. En réalité, des systèmes fondés sur la programmation dynamique ou sur des modèles probabilistes, dont le fonctionnement ne nécessite pas une segmentation préalable, ont obtenu des résultats de reconnaissance bien meilleurs que ceux qui segmentent explicitement la parole [Bahl 83]. Cependant, les systèmes sans segmentation demandent un calcul numérique important.

La segmentation est définie sur l'unité de reconnaissance. On se pose alors une question fondamentale: comment arriver à une partition correcte sans que les unités de reconnaissance

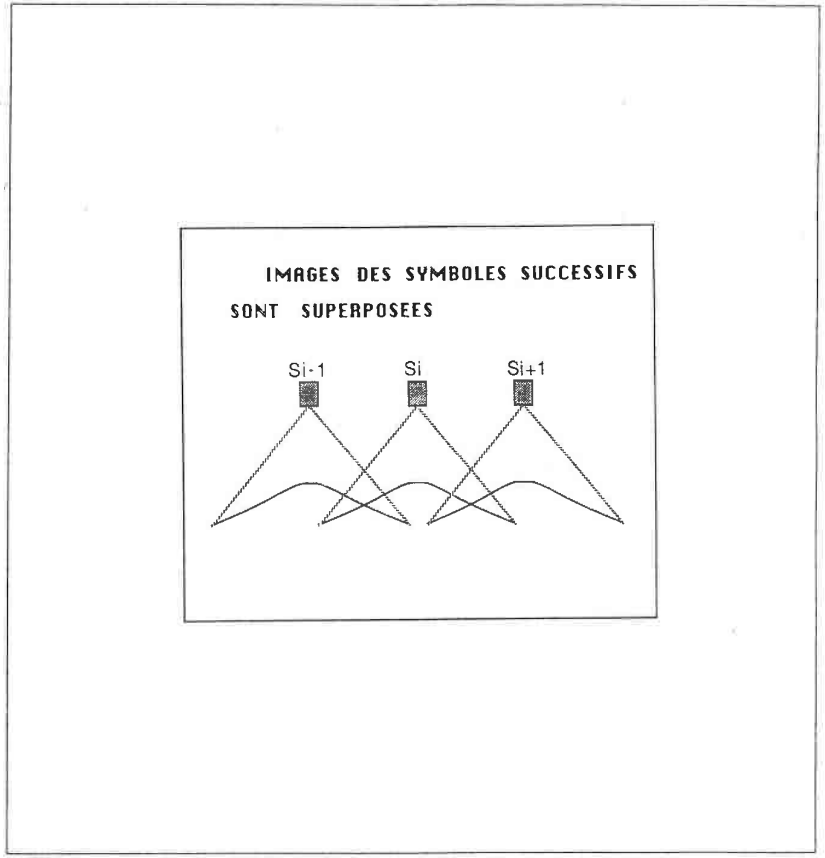


Figure 8.6: les images des symboles sont superposées

ne soient reconnues? En fait, pour obtenir une segmentation fiable, ces unités doivent être reconnues. Nous indiquons que la segmentation et la reconnaissance des unités ne doivent pas être considérées comme deux étapes entièrement distinctes dans l'interprétation de la parole. Par contre, elles sont reliées et doivent être traitées comme un processus.

6.3 Approches de segmentation

Segmentation par marquage des marges

La première approche, classique et traditionnelle, est fondée sur l'hypothèse suivante: entre les segments d'image consécutifs de deux unités, il existe une transition d'un certain ensemble de paramètres indépendante des combinaisons des unités. Cette hypothèse conduit naturellement à la méthode de segmentation par la détection des transitions. Pour résoudre le problème de la segmentation, beaucoup de travaux ont fait dans les directions suivantes:

- trouver un ensemble de paramètres et une décision de segmentation associée [Haton 79a],
- appliquer des principes de l'estimation statistique, [Zelinski 83] ou
- adapter la programmation dynamique [Svendsen 87].

Le défaut essentiel de cette approche est que la transition à l'aide de laquelle on pose une marque de segmentation n'existe pas toujours. En particulier si la différence entre la fin de l'image du symbole précédant et le début de l'image du symbole courant est petite ou nulle, la segmentation correcte est difficile ou même impossible. Par exemple, il est presque impossible de placer correctement les marques de segment phonétique pour la phrase anglaise *"we were away a year ago"*. Cette difficulté est illustrée par la figure 8.7.

Segmentation par localisation du centre

Notre proposition consiste à changer complètement la vue sur la segmentation. Alors que les marges de l'image d'un symbole sont indéfinies, il existe un point au centre de l'image d'un symbole où le signal est le plus proche de la référence de la reconnaissance. Nos hypothèses sont que le centre de l'image d'une unité de reconnaissance:

- doit recevoir le moins d'influence contextuelle, notamment l'effet de coarticulation (figure 8.8),
- peut être localisé approximativement et
- peut être utilisé pour guider la reconnaissance et la localisation des unités voisines.

Sous ces hypothèses, notre segmentation est réalisée par la reconnaissance des unités élémentaires et pendant le processus de reconnaissance.

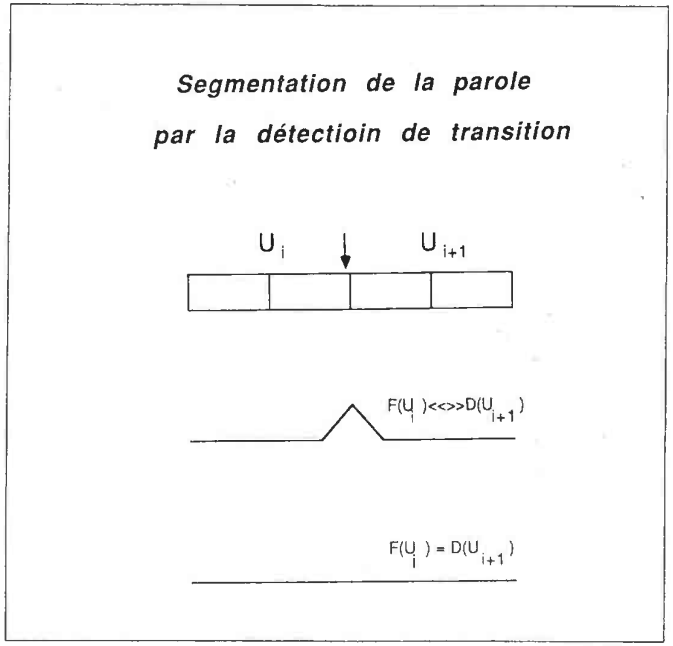


Figure 8.7: La segmentation préalable parfaite n'est qu'une vue de l'esprit

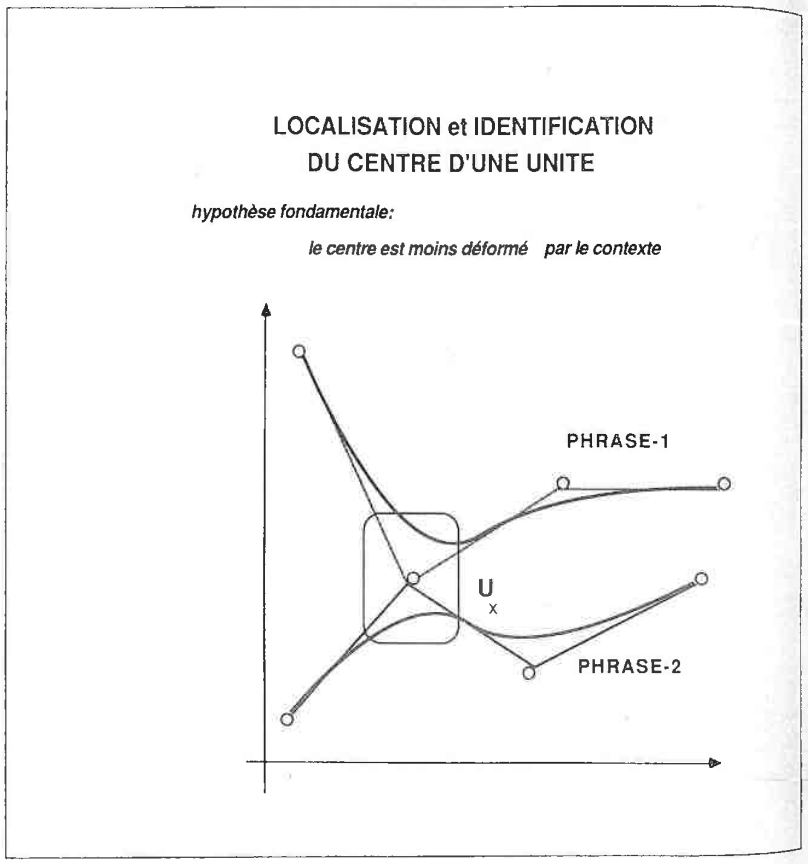


Figure 8.8: localisation et identification des unités

6.4 Reconnaissance sans pré-segmentation

Notre approche de reconnaissance sans segmentation préalable utilise le phonème comme l'unité de reconnaissance élémentaire. Un mot est décomposé en une liste de syllabes puis en une liste de phonèmes. Le processus de reconnaissance commence par construire quelques arbres d'interprétation partielle, liés directement avec les zones du signal où la reconnaissance phonétique est relativement sûre. Ces zones sont générées par les étapes suivantes:

- Dans le treillis des phonèmes, on cherche une liste de phonèmes qui ont un rapport de vraisemblance élevé. Ces phonèmes sont proposés à différents endroits temporels du signal.
- A chaque endroit proposé, pour chaque phonème dans la liste, on ajoute dans cette liste les phonèmes dont l'image est proche du phonème.
- On accède au dictionnaire lexical avec ces phonèmes comme entrée, et on obtient tous les mots contenant au moins une occurrence d'un phonème proposé.
- L'existence de ces mots est ensuite vérifiée dans le treillis de phonèmes et les survivants sont proposés comme les racines du processus de construction.

L'extension de ces arbres est ensuite effectuée par émission d'hypothèses, localisation et vérification des phonèmes voisins, en utilisant des contraintes de différents niveaux d'abstraction sur la parole. A chaque fois qu'il existe une incertitude sur l'identification ou sur la position d'un phonème, l'arbre d'interprétation concerné est développé en parallèle. Les arbres ayant une qualité d'interprétation faible sont supprimés au cours de la construction.

La méthode que nous utilisons est basée sur la localisation des syllabes. Nous supposons qu'une syllabe peut être localisée approximativement par la détection du centre de la voyelle et que le centre d'une voyelle est relativement stable malgré l'effet du contexte.

Pour localiser des syllabes, nous utilisons le résultat de la reconnaissance de phonèmes par la méthode développée dans 5.4. A chaque instant n nous distingons, dans la liste de candidats issues de la reconnaissance de phonème, des phonèmes voyelles et des phonèmes consonants. Nous définissons la valeur à l'instant n d'une fonction temporelle $S(n)$ par la formule suivante:

$$S(n) = \mathcal{L}(\frac{V_n}{V_n + C_n})$$

où V_n et C_n sont respectivement la moyenne des rapports de vraisemblance des premiers M candidats voyelles et N candidats consonee dans la liste de phonèmes. $\mathcal{L}$ est un filtre linéaire passe-bas. Ce filtre est destiné à supprimer les composantes fréquentielles qui sont inférieures à la fréquence moyenne des noyaux syllabiques dans la parole. M et N sont déterminés expérimentalement. En général, ils sont de l'ordre de 2 et 5 respectivement.

La propriété de la fonction $S(n)$ est la suivante:

- si le signal à l'instant n est reconnu très probablement comme voyelle, alors $V_n \simeq 1$ et $C_n \simeq 0$, $S(n)$ a une valeur proche de 1.

- si le signal à l'instant n est reconnu très probablement comme consone, alors $V_n \simeq 0$ et $C_n \simeq 1$, $S(n)$ a une valeur proche de 0.

Ainsi, les pics de la fonction $S(n)$ indiquent le centre des syllabes et ces pics sont utilisés pour la localisation des mots.

7 Reconnaissance des mots

7.1 Reconnaissance des phonèmes

Avec le modèle phonémique nous obtenons un treillis de phonèmes à chaque instant du signal de parole. C'est à dire qu'on sait à chaque instant aux tels rapports de vraisembla ce que le signal est étiqueté en telles classes de symboles, mais on ne connaît pas explicitement le début et la fin de chaque phonème. L'obtention de cette connaissance – la segmentation – est encore un autre type de problème de conversion signal-symbole.

Nous avons montré dans 6 que la variation du signal et par conséquent la fonction d'appartenance du signal à une classe de symboles est essentiellement *continue* et qu'une décision de segmentation n'est possible que dans des cas particuliers. Or, l'interprétation doit fournir une liste des mots et par conséquent des phonèmes *discrets*. Une discrétisation est donc inévitable. La discrétisation nous oblige à faire une approximation: de forcer à étiqueter le signal continu avec des symbole discrets.

Le treillis de phonèmes peut être utilisé de deux façons: la *proposition* des mots, et la *vérification* des mots. Nous présentons d'abord le lien entre un mot et les phonèmes, puis nous détaillons les deux utilisations.

7.2 Décomposition d'un mot en syllabes

Un mot sous forme orthographique peut être décomposé en syllabes, puis en phonèmes. Nous avons indiqué dans 5.4 que la variabilité et l'influence contextuelle de la parole sont modélisées au niveau phonème par des références multiples. En fait, cette modélisation n'est pas suffisante pour couvrir tous les cas, surtout en parole continue. Au niveau du mot, les variations de plus grand ordre sont modélisées symboliquement, sous formes d'insertion, de suppression et de substitution. Cette modélisatioon est réalisée par un ensemble de règles phonologiques au niveau du mot.

7.3 Proposition de mots

La proposition est de générer une liste de mots candidats que l'on peut associer à une zone de signal de la parole observé. les connaissances utilisables sont la liste de treillis, une liste de modèles de mots donnant l'ensemble de possibilités et parfois par une position temporelle approximative. En général, il faut comparer le signal préclassé à la liste de modèles de mots. Le temps de calcul est très important. Pour avoir une probabilité suffisante que le mot correct soit présent, le nombre de candidats proposés est souvent considérablement grand.

7.4 Vérification d'un mot

La vérification d'un mot consiste à calculer la valeur de la fonction d'appartenance du signal au symbole du mot puis confirmer l'existence du mot dans le signal. Par rapport à la proposition d'un mot, la phase de vérification dispose de plus de connaissances sur le mot à vérifier: la liste des phonèmes qui le constituent, les propriétés de ces phonèmes, le contexte phonologique, la position temporelle prévue approximativement (début, fin et centre) du mot par rapport à ses voisins, etc. Par conséquent, on peut espérer une précision plus élevée en faisant très peu de comparaisons car le symbole est connu.

7.5 Mesure de qualité d'un mot

En conséquence de notre modèle de phonèmes présenté dans 5.4, nous pouvons indiquer comment évaluer la qualité d'une interprétation.

A chaque instant les cellules fournissent une liste de symboles phonémiques dont chacun est associé à un coefficient de vraisemblance. Il est naturel que l'interprétation correcte du signal soit spécifiée par le chemin dans la séquence de ces listes qui réalise un coefficient de vraisemblance cumulé maximum. Nous utilisons notamment la masse de qualité des phonèmes composant le mot. La formule suivante calcule la qualité du mot M, $Q_{mot}(M)$ selon ce principe:

$$Q_{mot}(M) = \frac{1}{\sum_{p_i \in ph(M)} d(p_i)} \sum_{p_i \in ph(M)} d(p_i) \times Q_p(p_i)$$

où la fonction $ph(M)$ calcule la liste de phonèmes composant le mot M, $d(p)$ est la durée du phonème p et $Q_p(p)$ est la qualité de classification du phonèmes p:

$$Q_p(p) = \frac{1}{t_b - t_a} \int_{t_a(p)}^{t_b(p)} \mu_p(t) dt$$

7.6 Reconnaissance des phrases

Nous avons présenté dans ce chapitre la partie de reconnaissance de phonèmes, la localisation des syllabes et la reconnaissances des mots qui fournissent les symboles terminaux pour l'analyse syntaxique. Nous avons présenté également notre approche de la reconnaissance sans segmentation préalable qui permet d'effectuer la reconstruction de la structure syntaxique, présentée dans le chapitre 6, de la phrase et la segmentation de la parole dans le même processus.

Ces différentes composantes de notre système d'interprétation de parole continue sont organisées sous l'architecture *société de spécialistes*, présentée dans le chapitre 7, afin que la coopération des différents niveaux de traitement soit convenablement réalisée. Des aspects concrets sur le système seront données au chapitre 9 où nous présentons une application en interprétation du chinois parlé.

8 Conclusion

L'interprétation du signal de la parole continue nécessite des connaissances réparties dans toute la chaîne, de la production à la perception, du processus humain de la communication parlée. La conception d'une machine capable d'interpréter automatiquement la parole demande donc des études multi-disciplinaires. Les modèles actuellement utilisés pour représenter le processus d'interprétation de la parole sont presque tous des modèles comportementaux et non des modèles fonctionnels. On constate des progrès remarquables dans le domaine mais le problème est loin d'être résolu.

Le signal de la parole est un processus essentiellement continu. L'association d'un symbole à une zone de signal avec précision n'est faisable que sur des cas particuliers. Cette association introduit une discrétisation. Traditionnellement, on considère qu'une analyse syntaxique est un traitement discret-discret – la discrétisation étant faite avant l'analyse. Ainsi des erreurs irrécupérables sont introduites notamment sur la segmentation à cause de la perte d'information importante.

Nous considérons que la segmentation, la reconnaissance des mots et l'analyseur de structure syntaxique du signal doivent tous participer à la discrétisation, c'est à dire que le signal est discrétisé progressivement pendant l'interprétation. De cette manière les connaissances de haut niveau peuvent intervenir dans les décisions de la discrétisation et l'incertitude peut être ainsi diminuée progressivement au cours de l'interprétation.

Chapitre 9

Interprétation du Chinois Parlé

Nous présentons une réalisation de notre approche d'interprétation de la parole continue en interprétation du chinois parlé. L'interprétation du chinois parlé est difficile parce que cette langue est monosyllabique, tonale, avec un nombre de voyelles important, hautement non-stationnaire. Nous discutons dans ce chapitre notre démarche d'interprétation où l'ensemble de méthodes et techniques développées dans les chapitres antérieurs est appliqué. Nous discutons l'organisation et la représentation de connaissances de différents niveaux. Le système dont l'unité élémentaire de reconnaissance est le phonème est capable d'interpréter les symboles fournis par un décodeur acoustico-phonétique de qualité médiocre. Utilisant actuellement une grammaire du type hors-contexte comportant 250 règles et 250 terminaux, ce système a obtenu un taux de reconnaissance au niveau de la phrase supérieur à 90%. Ce système est opérationnel sur une machine du type Sun.

1 Introduction

Le chinois parlé est d'abord une langue monosyllabique. Toutes les prononciations de mots sont composées d'une voyelle (yun) précédée optionnellement par une consonne (sheng). L'ensemble de ces combinaisons donne environ 400 formes, insuffisantes pour définir les concepts de bases de la communication. Cette insuffisance a entaîné deux phénomènes particuliers en langue chinoise:

- l'utilisation du ton lexical, – à chaque syllable un ton parmi cinq catégories est associé pour désigner des concepts différents.
- l'augmentation du nombre de voyelles dont la plupart sont obtenues par la variation continue de la forme du conduit vocal à partir de la forme correspondant à une voyelle simple. Cette variation non seulement augmente le nombre de voyelles jusqu'à 37 en comparaison de 16 en français et 21 en anglais mais aussi donne naissance à des diphtongues et des triphtongues.

Remarquons qu'une langue monosyllabique n'implique pas que la segmentation est facile. Une syllable peut très bien être une voyelle seule ou précédée par une consonne semblable à

une semi-voyelle. Dans ce cas la segmentation de cette syllable avec la précédente est souvent impossible.

Nous avons indiqué dans le chapitre 8 les difficultés de l'interprétation de la parole continue que provoque par la variabilité. En interprétation du chinois parlé nous rencontrons des difficultés supplémentaires, classées en trois catégories:

- La première est que dans une langue monosyllabique la différence entre les images acoustiques des mots différents est plus faible par rapport aux langues où les mots se composent de plusieurs syllables. Par conséquent, on est forcé de reconnaître *sans tricher* chaque consonne et chaque voyelle pour identifier un mot. C'est en fait le problème bien connu dans toutes les langues: – les mots courts sont plus difficiles à reconnaître.
- La deuxième est le grand nombre de diphtongues et triphtongues qui oblige la reconnaissance d'un vocabulaire de phonèmes, déjà très difficiles pour l'anglais et le français, de taille plus importante et contenant deux types de sons dont l'identification demande des modèles et des techniques évoluées:
  - les diphtongues et les triphtongues qui sont hautement non-stationnaires.
  - les consonnes combinées. Par exemple des fricatives ayant un début occlusif, semblables à /t ʃ/ en anglais.
- la troisième est la reconnaissance des variations tonales. Les tons sont définis uniquement au niveau linguistique. Comme tous les autres attributs du signal de parole, ils subissent une déformation contextuelle en parole continue.

2 Phonétique expérimentale

2.1 Introduction

La définition et la classification des phonèmes couramment utilisées pour présenter les sons du chiois [71] sont basées sur la perception et le jugement subjectif. Ceci provoque des difficultés, parfois des contradictions, pour la reconnaissance automatique de la parole parce que dans ce contexte la définition et la classification doivent se fonder sur des quantités de la parole physiquement mesurables.

Un phonème est un symbole associé à un élément sonore d'un langage, la définition de cet élément devant vérifier les propriétés suivantes:

- avoir une distinction perceptionnellement significative avec d'autre éléments qui est
- en prononciation isolée, l'évolution de la forme du système phonatoire au cours de l'élocution est stable sous différentes combinaisons des éléments voisins.

2.2 Classification des phonèmes

Historiquement, il existe d'abord les sons et ensuite la classification: – la proposition d'un cadre pour répertorier ces sons. Donc selon des critères il peut y avoir plusieurs possibilités de

classement. Nous proposons, grâce à critères mesurables et non intuitifs, une classification des phonèmes dans la langue du chinois parlé. Nous présentons la classification dans les tableaux 9.1, 9.2, 9.3 et 9.4.

- Les diphtongues sont des combinaisons de deux phonèmes simples. Les triphtongues sont des combinaisons d'un phonème simple et d'une diphtongue. L'effet contextuel ajoute en général une grande différence par rapport aux phonèmes combinés et il est impossible d'utiliser la concaténation de deux phonèmes participant à la combinaison pour représenter une diphtongue ou une triphtongue.
- Les diphtongues et les triphtongues se produisent à partir des voyelles simples et à travers d'une variation continue et progressive de la configuration du système phonatoire. La variation peut aller vers une autre voyelle simple ou vers une nasalisation particulière. En particulier le second type de variation, appellé assimilation régressive [Chen 72], est produit par l'influence d'une consonante nasale sur la voyelle qui lui précède. Cette évolution peut être constatée facilement dans la table 9.3. Par exemple, le phonème /*i*/ donne la liste de variations suivante:

$$i \rightarrow \textit{ia ie in ing iao iou ian iang iong}$$

- Les fricatives occlusives sont formées d'une fricative précédée par une phase occlusive.
- Pour simplifier l'explication des règles de combinaison des sons, nous avons introduit les deux symboles {r,=i}. En effet, ils correspondent aux mêmes sons comme l'indigué le tableau 9.4.

Phonèmes du chinois parlé (putonghua)							
Consonne					Voyelle		
Plosive	Occlusive	Fricative	Nasale	Liquide	Simple	Diphtongue	Triphtongue

Table 9.1: Tableau des phonèmes du chinois parlé (putonghua)

Consonne																			
Plosive						Fricative										Nasal		Liquide	
Non-voisé			Voisé			Occlusive						Stationaire				m	n	l	r
p	t	k	b	d	g	z	c	j	q	zh	ch	f	s	x	sh	h				

Table 9.2: Tableau des consonnes

Voyelle													
Simple	F_1(Hz)	F_2(Hz)	F_3(Hz)	Diphtongue				Triphtongue					
-i	380	1300	2400										
=i	320	1600	2400										
o				ou	ong								
e	480	1500	2400	er	en	eng							
a	780	1300	2400	ai	ao	an	ang						
ü	300	2000	2500	üe	ün			üan					
i	300	2200	3400	ia	ie	in	ing	iao	iou	ian	iang	iong	
u	350	1000	2400	ua	uo			uai	uei	uan	uen	uang	ueng

Table 9.3: Tableau des voyelles

Phonèmes équivalents	
r	=i

Table 9.4: Tableau des phonèmes équivalents

2.3 Nonstationnarité

Le tableau 9.3 montre que la plus part des voyelles du chinois sont des diphtongues ou des triphtongues. Ceci implique que la plus part des voyelles sont nonstationnaires dans leur réalisation. Nous avons tracé les trajectoires de 250 voylles prononcées par un locuteur masculin dans un plan F_1-F_2 dans la figure 9.1 pour visualiser ce phénomène. Nous constatons que le movement de l'appareil phonatoire lors de l'articulation de diphtongue ou de triphtongue est nonégligeable pour la reconnaissance.

La figure 9.2 montre en plus de détail l'évolution de la diphthongue *ing* dans le plan F_1-F_2. L'articulation commence par la position du phonème *i* et puis, continuellement, vers une position nasalisée. On remarque aussi que le *i* est assez stable puisqu'il y a peu de variations formantiques.

3 Interprétation des tons du chinois

3.1 Introduction

Déformation contextuelle des tons

Un ton est la variation perceptionnellement significative de la fréquence fondamentale de la parole. Dans la langue chinoise, les tons portent l'information lexicale: la même configuration du conduit vocal avec des variations en fréquence d'excitation différentes donne

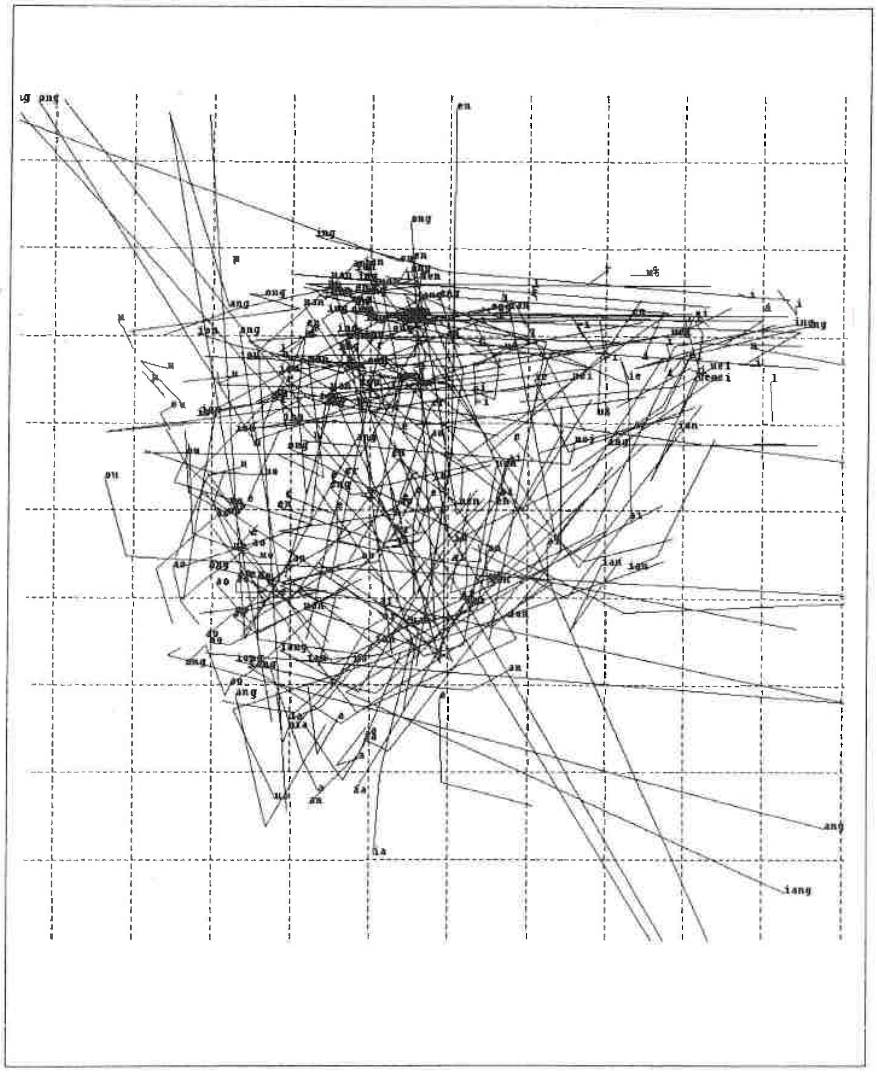


Figure 9.1: Les trajectoires des voyelles, prononcées par un locuteur masculin, dans le plan F_1 (en y, partant de haut) et F_2 (en x, partant de gauche). Les symboles sont imprimés à la fin de chaque trajectoire.

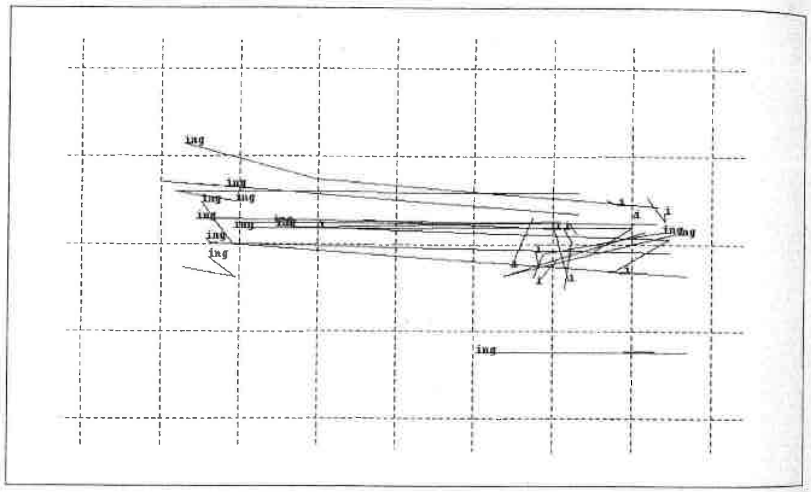


Figure 9.2: Les trajectoires de *ing* et de i, dans le plan F_1 (en y, partant de haut) et F_2 (en x, partant de gauche). La différence entre un phonème nonstationnaire et un phonème stationnaire est notable.

des sons qui ont des sens différents. Autrement dit, pour identifier un mot il est indispensable de reconnaître son ton. L'information tonale peut aussi aider à l'interprétation d'une phrase parlée du chinois.

L'information tonale peut être inclue dans un système de reconnaissance du chinois parlé afin de réduire le nombre d'hypothèses intermédiaires. Par exemple, parmi les 3344 combinaisons consonne-voyelle de la langue, il n'y a que 1200 qui sont réellement utilisées dans dans la prononciation, soit 36%.

Une statistique que nous avons effectuée sur 138 phrases chinoises prises dans un livre d'enseignement du chinois [78] montre encore davantage l'importance des tons. Nous avons relevé les symboles des tons dans les phrases prises. Les phrases trop longues sont coupées en deux, à condition que les deux parties soient toutes deux significatives. Les mêmes phrases sont inclues une seule fois. Le tableau 9.5 donne le résultat, où les chiffres entre 0 et 4 représentent les 5 tons, de T_0 à T_4. dans ce tableau les combinaisons de ton sont toutes différentes. Ceci implique que si on est capable

- de segmenter la parole en unités du mot et
- de reconnaître les tons de chaque segment,

il est possible de construire des systèmes de reconnaissance des phrases chinoises en utilisant uniquement l'information tonale.

Nous indiquons par ailleurs que la répartition des fréquences des tons est relativement équilibrée, comme le tableaux 9.6 montre.

En linguistique on classe les tons en cinq catégories selon la variation perçue de la fréquence fondamentale:

- T_1: haut et plat,
- T_2: montant,
- T_3: descendant puis remontant,
- T_4: descendant et
- T_0: prolongement du ton qui lui précède.

Cependant, la classification n'est qu'au niveau linguistique, comme pour d'autres paramètres de la parole. Au niveau phonétique et phonologique, il existe différentes réalisations d'un ton linguistique selon les tons qui le précèdent et qui le suivent. L'exemple le plus connu est qu'une suite de T_3 se transforme souvent en une suite de T_2 à l'exception du dernier T_3

$$\{T_3\}_n \rightarrow \{T_2\}_{n-1} T_3.$$

la dépendance contextuelle est due en grande partie à la tendance à faciliter la prononciation et à l'inertie du système phonatoire. L'influence de la déformation est tellement importante que même les auditeurs chinois commettent 51% d'erreurs à la reconnaissance des tons pour les mots pris de la parole continue et réarrangés aléatoirement [Lee 86]. La difficulté de la reconnaissance des tons réside principalement dans l'influence du contexte.

10102232141	102424	10324213133413	104013
104040111434	10404113	104124220330	104330
104413	1044240	1044420	104444
110231	11030310	11033044	110441443012
111144240211	11143343	111444143341	1121220434431
11342014042	1134313122243	11402132024	114120211
114124330334	11414042	1142243	1143414
12	1212243	12202410	1220310
12204201	122211213033	12244414244	123043013
130303	13033	13033314	13043
1334430	14300	14320304	14320343
143312123222034	14333420	2	21034
212322444340	21232442443	2130141123014222	2143301022144144312
2212204243	23104413	23314220	240
243	2430424	3	3010223144401
3011043414343	3011214	3012124444414	30123
3021130	3024031222123141411110	3030424	303043433
303043433	303440	303441020122430	3034410201420
3034440	304040	30412020	3041314342220
3041443033	30431443124242	3044131020120	3101114330
31011143340	31034422220	31041304	31042122010010
31111110	31123234401230	3122234144220	31234144220
312440410	314220212224344120242	32013	3204243
32110203014221424	3211133124221441111	322414	3320134304
33244124	3330403	33430412011433	33432042
33444	34	3411020414240231	341430
3420	34210	34221411	3423214244044
34310	4	4001432400	40032341220
403	403320430	4042430203314	4043440
41114320	41230231	41244420	413441142243
42040424114111211	420420444113	4204430	43131
43240	43302411120	4344241104314033	44122334
441324334340	4414310324	442221220431114213	443114413
44330310	4434200	443430	444110
444221	44440		

Table 9.5: combinaisons des tons sur 138 phrases chinoises

ton	T_0	T_1	T_2	T_3	T_4
fréquence	0.158	0.216	0.171	0.200	0.254

Table 9.6: la répartition des tons dans les 138 phrases (1266 tons)

Motivation

Des études ont été menées sur le comportement des tons. Des travaux fondés sur des méthodes purement statistiques et sur la mesure de distance locale ont permis de déterminer des paramètres relativement discriminants et ont donné des résultats intéressants. Ces résultats sont cependant encore non-exploitables réellement par des niveaux de traitement supérieurs à cause des taux d'erreurs trop élevés (30-40%) [Halle 85]. Cependant, comme dans d'autres domaines d'application, l'utilisation de modèles statistiques est inadéquate pour la reconnaissance de structures complexes [Nandahakumar 85]. Les méthodes purement statistiques et locales qui n'incluent pas de contraintes contextuelles sont en particulier insuffisantes en reconnaissance de la parole pour modéliser la dépendance contextuelle, les défauts de détection de fréquence fondamentale et la variation inter- et intra-locuteur. Elles sont ainsi totalement incapables de distinguer différents tons correspondant à des variations de courbes semblables. Le raffinement des techniques statistiques, tel que la recherche des invariants, la définition de différentes distances,... n'améliorent pas sensiblement les résultats.

Pour arriver à un résultat de reconnaissance des tons utilisable dans un système de reconnaissance de parole, il est nécessaire d'exploiter des techniques plus fines, notamment l'introduction de connaissances contextuelles. Ces techniques consistent à donner une interprétation structurelle des formes primitives obtenues par une classification rudimentaire du signal. Dans la résolution des autres problèmes de reconnaissance de formes similaires au nôtre, des méthodes syntaxiques (analyse grammaticale) ou l'utilisation de systèmes experts ont permis d'améliorer considérablement les résultats.

Dans l'état actuel de la recherche sur les tons une certaine expertise a été acquise mais la connaissance reste morcelée et incomplète. On ne connaît pas, par exemple, le nombre de formes nécessaires pour représenter les variations de tons, ni les relations phonologiques entre eux, la dépendance contextuelle des formes, etc. Il est donc nécessaire d'utiliser un système souple, facile à modifier et permettant d'améliorer progressivement nos connaissances. Nous considérons donc qu'une approche d'I.A. par système à base de connaissances sous forme de règles de production est bien adaptée à une telle tâche.

3.2 Présentation générale de la méthode

A partir du signal vocal continu fourni en entrée notre système fournit une suite de symboles correspondant à la variation linguistique de tons de la phrase, assortis de coefficients de vraisemblance. Il s'agit donc de reconnaître à laquelle des cinq classes disjointes précédentes (T_1, T_2, T_3, T_4 et T_0) appartient un segment de parole.

Notre système comporte un niveau d'interprétation (traitement symbolique) et un niveau

d'identification de formes primitives qui fournit les symboles à partir du signal de parole. Les grandes étapes du traitement sont:

- calcul et lissage des contours de pitch et d'intensité à partir du signal de parole;
- segmentation automatique du signal de parole en segments linguistiques par une analyse morphologique;
- modélisation des segments du contour de pitch par des coefficients de n-ième régression;
- classification des segments en formes primitives par une méthode statistique;
- interprétation de ces formes primitives pour reconnaître les tons à l'aide d'un moteur d'inférence raisonnant sur des faits incertains et disposant d'une base de connaissances sous forme de règles de production. Cette base contient l'information contextuelle (et l'information sur d'autres relations), les données fournies par les étapes précédentes constituant la base de faits initiale.

3.3 Pré-traitements

Estimation de la fréquence fondamentale

Le contour de la fréquence fondamentale de la parole est calculé par notre éditeur de signal [Gong 85b] avec indication de non-voisement ou silence. Nous utilisons un détecteur de la fréquence fondamentale basé sur la technique présentée en chapitre 3. Bien que ce détecteur fournisse des résultats très fiables, nous avons inclus une phase de traitement afin de corriger des erreurs dues à l'irrégularité de prononciation et à d'autres causes. Ce traitement consiste d'abord à calculer l'histogramme du contour – la fréquences de chaque valeur dans le contour – et ensuite à remplacer les points dont la valeur est à l'extérieur de l'histogramme, filtré par un filtre passe-bas, par une valeur obtenue en moyennant certains points précédents. Un opérateur médian est enfin appliqué sur le contour ainsi traité pour éliminer le bruit.

Estimation du contour d'intensité

Dans notre étude, la segmentation du contour de pitch est fondée sur le contour de l'intensité du signal de parole. L'intensité est la valeur maximum d'une période de signal de parole voisé estimée statistiquement. Naturellement, elle est différente de l'énergie moyenne d'un segment du signal. Notamment, le contour de l'intensité montre une nette variation pour le changement de voyelles pour lesquelles l'énergie est souvent constante. Ce paramètre n'a pas été utilisé dans le traitement de parole parce que sa détection est difficile. L'algorithme de détection de pitch que nous utilisons permet aussi l'estimation du contour d'intensité. Cet algorithme, utilisant un modèle temporel, est non seulement capable de fournir la valeur de période de l'excitation mais aussi d'indiquer la valeur et la position du maximum (ou minimum) dans chaque periode du signal. Le contour d'intensité est ensuite soumis à un filtre passe-bas. Pour ne pas introduire de déphasages entre les différentes composantes

spectrales et obtenir ainsi une bonne précision lors de la localisation des segments, nous utilisons un filtre à réponse finie.

3.4 Pré-segmentation automatique

Afin de modéliser la variation tonale, nous devons effectuer une segmentation grossière du signal. Dans la parole continue, chaque transition de phonèmes entraîne un changement du signal et provoque en général une variation du contour d'intensité. Notre méthode de segmentation consiste à rechercher sur ce contour tous les creux correspondant aux transitions. Nous avons utilisé le formalisme de la morphologie mathématique pour l'analyse de contours. La morphologie mathématique est une théorie ensembliste initialement appliquée à l'image [Serra 82]. Une image est une fonction

$$f : \Re \times \Re \rightarrow \Re$$

telle que

$$\exists m \in \Re, \forall x, 0 \leq f(x) \leq m$$

c'est donc un signal à deux dimensions. La morphologie mathématique permet de systématiser des idées et des méthodes de traitement d'image et fournit une méthodologie de traitement avec des outils appropriés. L'utilisation d'éléments structurants – objets géométriques pouvant être déplacés sur toute image – et l'analyse à l'aide d'opérations ensemblistes constituent les concepts de base des algorithmes de la morphologie mathématique. Deux opérations élémentaires sont intéréssantes pour détecter les creux dans le contour de l'intensité:

dilatation:

$$(f \oplus B)(x) = \sup\{f(y);\ y \in B_x\} \tag{9.1}$$

Cette opération donne l'ensemble des plus grands éléments dans $f(y)$ lorsque y parcourt l'ensemble B centré en x;

érosion:

$$(f \ominus B)(x) = \inf\{f(y); y \in B_x\} \tag{9.2}$$

Cette opération donne l'ensemble des plus petits éléments dans $f(y)$.

Dans 9.1 et 9.2, B_x représente l'élément structurant symétrique centré en x.

On peut trouver une définition pour le signal temporel à une dimension dans [Slimane 87].

En considérant que le contour d'intensité I est une image particulière, nous composons ces opérations pour obtenir un signal C contenant des pics normalisés correspondant aux creux du contour, la largeur des creux pouvant être sélectionée:

$$C = (I \oplus B) \ominus B - I \tag{9.3}$$

Nouns examinons ensuite les pics fournis par l'analyse morphologique. Si la hauteur d'un pic – la profondeur d'un creux dans le contour d'intensité – est suffisamment grande, le point correspondant est considéré comme une limite de segment candidat. Certains creux sont produits par des petites variations à l'intérieur du signal d'un phonème, par exemple entre

la frontière de la zone instable, où le contour de l'énergie est montant ou descendant, et la zone stable du signal d'un phonème, et donc ne correspondent pas à des limites de segment. Ces points sont éliminés par un critère fondé sur les rapports de la valeur du premier pic à gauche et celui à droite du creux à la valeur du creux du contour d'intensité.

3.5 Modélisation du contour

L'objectif de la modélisation du contour est d'extraire un ensemble restreint de paramètres qui caractérisent la variation de fréquence fondamentale avec une perte d'information aussi faible que possible. Ces paramètres sont utilisés par le niveau d'interprétation si nécessaire. Chaque segment est modélisé par sa $n^{ème}$ régression par rapport au temps. A partir de ce modèle, il est facile de calculer en chaque point la valeur du contour en n multiplications, ainsi que les dérivées première et seconde. Nous avons synthétisé le contour à l'aide des coefficients de régression et constaté que la précision est suffisante lorsque l'ordre du polynôme n est supérieur à 2. Cependant, si les valeurs de pitch des premiers points ou des derniers points d'un segment ne sont pas correctement estimées, le contour du segment synthétisé est déformé de façon notable.

3.6 Classification

Le but de cette étape est de classer les courbes de variation tonales en formes primitives. Pour modéliser tous les phénomènes de variation d'un ton, nous avons utilisé références multiples pour chaque ton. Soit

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, ...\}$$

l'ensemble de classes de tons où

$$\mathcal{C}_i = \{\mathbf{C}_{i,1}, \mathbf{C}_{i,2}, \mathbf{C}_{i,3}, ...\}$$

est l'ensemble des sous-classes pour un ton i donné. Ces ensembles sont disjoints:

$$\mathbf{C}_{i,j} \cap \mathbf{C}_{k,l} = \emptyset \quad \forall i, j, k, l$$

Un vecteur $R_{i,j}$, correspondant à la forme de référence dans l'espace de représentation est associée à chaque sous-classe $\mathbf{C}_{i,j}$. Soit F le vecteur de test. La classification de F consiste à déterminer $\mathbf{C}_{i,j}$ tel que:

$$||F - R_{i,j}|| = \min_{i,j} ||F - R_{i,j}||.$$

Les variations de durée d'élocution entraînent des segments de longueur différente et nécessitent une normalisation temporelle avant la comparaison avec des formes de références. Les frontières de segments étant détectées avec une grande précision, nous avons utilisé un simple réglage temporel linéaire pour que la longueur de la forme de test et de la forme de référence soient égales.

3.7 Interprétation

Introduction

Le phénomène de dépendance contextuelle des tons est très courant en chinois. On observe ainsi des courbes correspondant au même ton et la même forme de courbe pour des tons différents. Citons par exemple: T_4 précédé par T_3 a tendance de devenir T_2; T_3 et T_1 pour différents locuteurs peuvent avoir tendance à être identiques; T_3 précédé par T_1 a tendance à se présenter comme T_4, etc. Nous avons utilisé le terme *tendance* Pour dire que le phénomène est possible mais qu'il y a des exceptions. Un raisonnement approximatif est ainsi nécessaire. Pour la description des tons, nous n'avons pas un vocabulaire suffisant pour exprimer et formuler les connaissances. Ceci a soulevé des difficultés dans la construction de la base de connaissances déclaratives et dans la définition des fonctions fournissant des informations symboliques nécessaires pour le raisonnement.

Système d'interprétation

Dans notre système, le niveau d'interprétation est chargé de corriger les erreurs de reconnaissance de tons introduites par la classification statistique. Cet interprète est implanté sous forme d'un système expert. Le système de correction de déformation contextuelle présenté dans le chapitre 5 est utilisé. Pour chaque segment du signal de parole, les données fournies par les étapes précédentes sont:

- les coefficients de régression. Ces coefficients contiennent toute l'information, sous forme très compacte, sur la forme de courbe du contour de pitch et peuvent être utilisés éventuellement pour calculer des paramètres nécessaires lors de l'interprétation;
- les noms des tons primitifs reconnus par la classification statistique avec les scores de reconnaissance;
- l'emplacement temporel et la longueur du segment;
- la moyenne de fréquence sur une phrase.

Ces informations sont représentées sous forme d'un objet et la donnée de l'interprète est ainsi une liste de ces objets.

Les connaissances sur la déformation contextuelle sont représentées sous forme de règles de production. Nous présentons un extrait des règles utilisées par le système dans l'appendice B. Les identificateurs C, N, P représentent respectivement le segment courant, suivant et précédent du signal.

3.8 Résultats expérimentaux

Le corpus que nous avons utilisé pour ajuster et tester notre méthode contient 156 chiffres (de 0 à 10) en chinois enregistré dans une salle de consoles par un locuteur masculin. Certains chiffres sont prononcés différemment – le même phénomène qu'en anglais où *zero* se prononce *zero* ou *o* – et le corpus contient 15 sons. Pour pouvoir comparer les résultats de

reconnaissance automatiquement, le signal de parole a été préalablement étiqueté manuellement.

Notre méthode de segmentation a permis de localiser correctement tous les segments du corpus. Le résultat de la classification de formes primitives est présenté dans la table suivante (où T_i et tr_i sont des données et des résultats respectivement):

	tr_1	tr_2	tr_3	tr_4	total
T_1	50(83%)	1	0	9	60
T_2	0	7(100%)	0	0	7
T_3	0	0	42(98%)	1	43
T_4	0	0	0	46(100%)	46
total					156(93%)

Table 9.7: Résultats de préclassification à références multiples des tons

Les erreurs de préclassification sont dues aux problèmes suivants:

- la longueur des formes à identifier est normalisée pour pouvoir les comparer avec les références, mais l'information sur la longueur n'est pas utilisée en classification. Ceci est la source principale d'erreurs;
- l'effet de contexte dont nous avons cité quelques exemples antérieurement;
- les erreurs d'estimation de contour de la fréquence fondamentale et de segmentation.

Toutes ces erreurs ont été corrigées par le système expert d'interprétation.

3.9 Conclusion

L'approche règles de production est puissante et souple, elle nous a permis de modifier, d'augmenter et de compléter facilement le système. L'utilisation d'une base de connaissances nous a permis aussi d'introduire l'interaction entre le traitement du signal et le traitement des symboles, les deux étant traditionellement séparés. La méthode est efficace pour inclure des connaissances contextuelles et perceptuelles et des connaissances pour la correction des erreurs de détection de formes primitives. Ces connaissances sont difficiles à modéliser mathématiquement.

Malgré le manque de connaissances phonétiques et phonologiques, les premiers résultats obtenus montrent que la méthode est prometteuse et on obtient facilement d'assez bons résultats sur un petit corpus qui peuvent être utilisés comme l'entrée d'un système de reconnaissance du langage parlé.

Nous pensons que l'évaluation doit continuer pour compléter la base de règles, pour obtenir une statistique fiable sur la performance et pour que le résultat du système soit réellement utilisable dans la reconnaissance de la parole.

4 Représentation des connaissances statiques

Les connaissances statiques sont des relations entres des objets connues a priori et manipulées dans les différents niveaux d'abstraction et constituent une délimination de l'espace de solution qui ne varie pas au cours de l'interprétation. La qualité de ces connaissances ainsi que les mécanismes d'exploration déterminent les performances du système. Pour effectuer des traitements en interprétation, les connaissances statiques doivent être mise en utilisation par la machine. La représentation des connaissances constitue un interface qui nous permet d'exprimer notre compréhension du problème sous une forme assimilable par la machine. Nons exigeons qu'une représentation doit vérifier les points suivants:

- Pour pouvoir maintenir les connaissances la représentation doit être facilement compréhensible par l'homme. Un compilateur peut être utilisé pour convertir en forme plus facilement exploitable par la machine.
- Pour que l'ensemble des connaissances soit cohérent, une connaissance ne doit pas être entrée à plus qu'un endroit.
- La modification doit être facile.

Nous présentons l'ensemble des connaissances utilisées dans notre système d'interprétation du chinois parlé.

4.1 Niveau signal

Au niveau du signal, ce que l'on connaît a priori sur le problème est en général sous forme d'une notion réalisable par des programmes du traitement du signal.

Du côté de production de la parole, on sait que la parole humaine contient des variations dont l'étendue fréquentielle est environ entre 50Hz et 10kHz. Nous utilisons un filtre linéaire appliqué au signal de la parole pour sélectionner cette bande fréquentielle qui porte de l'information.

Du côté perception, des études ont montré que l'échelle fréquencielle *MEL*, qui consiste à effectuer une transformée de fréquence linéaire dans la bande inférieure 1kHz et logarithmique dans la bande supérieure, correspond mieux au comportement de l'oreille humaine qu'une échelle entièrement linéaire [Davis 80]. Dans l'évaluation de notre système cette échelle est utilisée.

L'information sur la variation du contour d'énergie à court terme du signal de la parole est très importante dans la perception humaine. Cette variation est également inclue dans l'ensemble de paramètres caractérisant la parole.

4.2 Niveau signal-symbole

Cet ensemble de connaissances fournit au système d'interprétation des images des symboles primitives et assure donc la connexion du système avec le monde réel.

Nous avons essentiellement trois types de connaissances à ce niveau:

- Les formes de référence, comprenant des profils spectraux, utilisées par notre modèle de phonèmes présenté dans le chapitre 8. En effet, la variation du signal de parole représentée dans R^n a les contraintes suivantes:
  - les vecteurs n'occupent pas tout l'espace mais un sous-espace.
  - la trajectoire d'une séquence de vecteurs correspondant à un son réel n'est pas arbitraire.

  Ces formes de références imposent ces contraintes. Les références sont multiples. C'est à dire qu'on associe à chaque symbole primitif plusieurs images acoustiques possibles. Le tableau 9.8 donne un extrait de notre liste de références de phonèmes.
- Les durées approximatives des unités phonémiques forment une autre sorte de contrainte.
- L'influence du contexte sur la variation tonale est représentée par des règles de production dont nous avons discutées dans 3.7.

4.3 Niveau lexical

Ce niveau de connaissances comprend

- les connaissances sur la qualité du convertisseur signal-symbole du niveau inférieur dans l'objectif du traitement de l'incertitude et
- les connaissances sur la relation des mots terminaux avec les symboles primitifs phonèmes.

Qualité de la conversion signal-symbole

Etant donnés un symbole de phonème et la vraisemblance avec laquelle le signal est produit par ce symbole, nous pouvons estimer les vraisemblances avec lesquelles le signal est produit par d'autres symboles de phonème, à condition de connaître la fonction de transfert signal-symbole du convertisseur. La connaissance sur les performances du décodeur acoustico-phonétique du système est représentée sous forme de règles, comme illustré dans les tableaux 9.9 et 9.10.

Le tableau 9.9 montre la qualité de la conversion signal-symbole des voyelles. Dans la liste de confusion du phonème p sont indiqués les phonèmes que le convertisseur peut proposer à la place de p. Le nombre associé mesure la possibilité de la confusion. Nous constatons par exemple que pour le décodeur spécifique la classification des phonèmes an, ian, ang, eng, e, est très floue car ils sont acoustiquement voisins et donc difficiles à reconnaître. Tandis que le décodeur ne confond pratiquement jamais le phonème i.

Dans le tableau 9.10 nous remarquons qu'en général les consonnes sont plus difficiles à reconnaître que les voyelles. La classification pour p, t, k est en particulier très grossière.

Dans les deux tableaux nous donnons aussi quelques paramètres sur l'image des symboles utilisés par le niveau lexical pour la localisation des mots.

N_f	n_1	n_2	$v(\%)$	n_p	Name	Type
BB	148	149	0	2	c	S
BB	178	179	0	2	i	Y
BB	200	201	0	2	l	S
BB	225	226	0	2	i	Y
BB	347	348	0	2	h	S
BB	353	358	20	3	ua	Y
BD	28	35	30	3	ing	Y
BD	83	86	0	3	u	Y
BD	97	98	0	2	sh	S
BD	100	101	0	2	=i	Y
BF	42	43	0	2	f	S
BF	145	150	30	3	iou	Y
BF	194	195	0	2	x	S
BL	35	37	0	3	l	S
BL	121	130	20	3	ian	Y
BL	153	154	0	2	ch	S
BL	160	167	20	3	ang	Y
BL	444	445	0	2	i	Y

Table 9.8: extrait de la définition des formes de références des phonèmes utilisée pour la conversion signal-phonèmes (Note: N_f est le nom du fichier duquel la référence est extraite, n_1 et n_2 sont respectivement l'instant temporel du début et de la fin de la référence dans le fichier, v indique la variation de la longueur de la référence au moment de comparaison, n_p est le nombre de profils pour représenter l'image du phonème, et Name et Type donne le symbole et son type (sheng ou yun) du phonème, respectivement.)

Table de Description des Voyelles				
sym	l_{moy}	f	l_{min}	liste de confusions
ü	8	0.33	2	(u . 0.7)
er	8	0.32	3	(e . 0.65) (ao . 0.7)
ai	10	0.33	4	(e . 0.5) (an . 0.6) (ei . 0.7)
a	10	0.32	4	(ia . 0.7)(er . 0.68)
uo	10	0.32	4	(ou . 0.9) (u . 0.6)(ao . 0.78)
ou	10	0.32	4	(uo . 0.9) (ao . 0.8) (u . 0.7)
i	10	0.34	2	
ang	13	0.32	4	(eng . 0.75) (ong . 0.8) (iang . 0.75)(an . 0.65)
ian	9	0.32	4	(an . 0.78) (ang . 0.7) (iang . 0.75)(en . 0.65)
ua	8	0.34	4	(a . 0.7) (ao . 0.65)
uan	8	0.30	3	(an . 0.8) (uen . 0.7)
uen	7	0.32	3	(uan . 0.7) (en . 0.7) (uei . 0.6)(eng . 0.7)
uei	8	0.32	4	(ei . 0.8)
uai	7	0.30	3	(er . 0.8) (ai . 0.7)
ie	8	0.32	3	(ei . 0.7) (e . 0.66)
ong	11	0.32	4	(eng . 0.7) (ang . 0.74) (eng . 0.66)
eng	12	0.32	4	(ong . 0.75) (en . 0.75) (ang . 0.7) (an . 0.6) (uen . 0.6)
iang	10	0.32	4	(ang . 0.9) (ian . 0.76) (eng . 0.7)
u	8	0.32	3	(uo . 0.75) (ou . 0.7)
en	8	0.32	4	(eng . 0.9) (uen . 0.75)
an	7	0.32	4	(uan . 0.64) (ai . 0.65) (ian . 0.6) (en . 0.7)(ang . 0.67)
ao	8	0.32	3	(ou . 0.76) (uo . 0.83) (e . 0.66) (er . 0.78)
iou	7	0.32	3	(ou . 0.65) (u . 0.7) (uo . 0.6) (ü . 0.65)
ei	8	0.32	3	(i . 0.6) (ie . 0.7)(ai . 0.7)
=i	4	0.33	2	(e . 0.6) (r . 0.79) (-i . 0.6)
-i	5	0.32	2	(e . 0.65) (=i . 0.8)
e	8	0.32	3	(ao . 0.8) (=i . 0.65) (ai . 0.65) (uo . 0.7)(en . 0.65)
ia	9	0.33	3	(a . 0.76)
in	10	0.33	3	(ing . 0.67)
ing	12	0.32	4	(ian . 0.56)(ang . 0.65)

Table 9.9: Descriptions des symboles des voyelles en terme de la qualité de la conversion signal-symbole et des mesures sur le signal

- l_{moy} et l_{min} sont respectivement la longueur moyenne et minimale de l'image, mesurées en nombre de profils;
- f mesure la fiabilité a priori du décodeur lorsqu'il distribue un symbole particulier.

Table de Description des Consones				
sym	l_{moy}	f	l_{min}	liste de confusions
l	5	0.30	2	(m . 0.65)
ss	10	0.5	5	(s . 0.6) (f . 0.6)
sh	7	0.32	2	(zh . 0.8) (ch . 0.8) (x . 0.7)
s	6	0.33	3	(f . 0.9)(x . 0.7) (z . 0.7) (sh . 0.76) (h . 0.75)
ch	7	0.32	2	(sh . 0.9) (zh . 0.85) (q . 0.66) (z . 0.6)
zh	5	0.32	2	(r . 0.7) (z . 0.8) (c . 0.6) (sh . 0.68) (ch . 0.7) (j . 0.8)
c	5	0.29	3	(s . 0.9) (sh . 0.7) (z . 0.7)
r	9	0.32	3	(zh . 0.8)
j	4	0.30	2	(q . 0.85) (x . 0.85) (z . 0.85) (zh . 0.85) (t . 0.7)
x	7	0.32	3	(q . 0.9)(t . 0.7)(sh . 0.84)(ch . 0.75) (j . 0.76)(s . 0.7)
z	5	0.30	2	(s . 0.87) (zh . 0.75) (c . 0.76) (j . 0.76)
n	6	0.33	2	(m . 0.75)
t	5	0.3	2	(p . 0.9) (k . 0.8) (d . 0.75) (b . 0.66)(x . 0.7)
m	5	0.33	2	(n . 0.7) (l . 0.6)
q	7	0.3	3	(x . 0.9) (j . 0.87) (sh . 0.7) (ch . 0.7)
f	6	0.33	2	(h . 0.95) (s . 0.9)
p	5	0.30	2	(t . 0.9) (k . 0.7)(q . 0.7)
b	4	0.3	1	(d . 0.9) (m . 0.65) (k . 0.7) (g . 0.85) (t . 0.8)
d	5	0.3	1	(b . 0.9) (z . 0.8) (g . 0.7) (t . 0.6) (n . 0.65)
h	5	0.28	2	(f . 0.7) (k . 0.7)(d . 0.65)(g . 0.65)
k	5	0.28	2	(h . 0.7) (p . 0.7) (t . 0.7)
g	5	0.27	2	(d . 0.8) (z . 0.8) (h . 0.6) (b . 0.75)(k . 0.9)

Table 9.10: Descriptions des symboles des consonnes en terme de la qualité de la conversion signal-symbole et des mesures sur le signal

Règles de décomposition du mot

Ce sont des règles de décomposition d'un mot en séquences de symboles de phonèmes. Les types de règles suivants sont formulés dans notre application:

- la conversion de la forme orthographique vers une forme normalisée;
- la généralisation des variations phonologiques possibles d'un mot et d'un phonème dans des différents contextes. Les tableaux 9.11 et 9.12 donnent des exemples;

- la décomposition d'un mot en syllables;
- la décomposition d'une syllable en une liste de phonèmes;
- la décomposition des diphtongues et des triphtongues en phonèmes ou/et diphtongues.

```
(z-i.0 ze.0)
(kuai.4 ker.4)
(me.0 mao.0)
(shen.2 she.2)
(ji.3  j.3)      ; T-3  reduces duration
(qi.3  q.3)
(xi.3  x.3)
(ju.3  j.3)      ; T-3  reduces duration
(qu.3  q.3)
(xu.3  x.3)
(sh=i.2 sh.0)     ; a vowel may be absorbed in a context
(li.2  yi.2)      ; ju.4 li.2
(li.3  yi.3)      ;
(li.4  yi.4)      ;
```

Table 9.11: Extrait des règles des variations phonologiques au niveau mot. Le mot de la partie gauche de chaque couple peut se prononcer comme le mot donné en partie droite.

Le tableau 9.12 donne quelques exemples des variations phonologiques au niveau des phonèmes, en particulier les diphtongues et les triphtongues dont nous avons expliqué la nonstationnarité.

```
(iao  (ao)(i ao))
(iou  (i iou)(iou))
(u%an (u% ian)(e en)(ian))
(u%n  (u% ing))     ; xun 4
(u%e  (u% e)(u% ai)(u% ei))
(uang (u ang)(ang))
(ong (ou ong)(ong))
(iang (i ang)(iang))
(uai  (uo ai)(ai)(uai))
```

Table 9.12: Extrait des règles des variations phonologiques au niveau phonème. A chaque symbole phonémique est associée une liste de variations de prononciation possibles

4.4 Niveau syntaxico-sémantique

Présentation générale

Pour interpréter un message, il est nécessaire d'utiliser aussi trois autres types de connaissances, la syntaxe, la sémantique et la pragmatique.

- Syntaxe: La syntaxe est l'ensemble des règles sur l'ordre d'apparition des mots qui prescrit la norme de transmission des messages.
- Sémantique: La correction syntaxique n'est pas suffisante pour que la phrase soit acceptable (ou pour que la phrase soit compréhensible). La sémantique impose des contraintes sur la combinaison des mots dans une phrase syntaxiquement correcte.

  La solution généralement adoptée en linguistique pour représenter la sémantique consiste à attacher à tous les éléments lexicaux des traits caractéristiques rappelant la nature du signifié. L'introduction des traits permet de définir un système de règles dont le principe est le suivant:

  > On ne peut pas associer deux mots dont les traits sont incomparables [Jayez 82].

- Pragmatique: Un discours naturel doit utiliser les connaissances acquises avant et après la communication. La pragmatique est l'étude sur l'environnement d'une phrase. La pragmatique constitue une contrainte dynamique sur la phrase alors que la syntaxe et la sémantique sont des contraintes statiques.

La classification de ces connaissances est floue, il est très difficile de les distinguer nettement lorsqu'on traite un problème particulier [Pitrat 77]. Par exemple au sens de classification des mots en catégories, la syntaxe et la sémantique sont très proches et sont interchangeables. Dans notre système, le niveau syntaxico-sémantique fournit un modèle linguistique qui constitue un ensemble de contraintes du domaine de la tâche.

Grammaire sémantique

Nous avons adopté la grammaire sémantique [Burton 76] pour représenter les connaissances syntaxiques et sémantiques. La grammaire sémantique utilisée actuellement est du type contexte libre et est constitué de 250 règles de production et 250 terminaux. La grammaire est donnée en appendice C. Cette grammaire autorise un facteur de branchement maximal d'environ 100 et permet de générer des séquences de chiffres de longueur jusqu'à 3.

Cette grammaire décrit deux tâches de façon simplifiée: le contrôle d'un robot dans un bureau et la demande d'information sur la réservation des billets de train.

A chaque production est associée une fonction exécutable paramétrée par la partie droite de la production. Elle réalise la traduction du non-terminal dans la partie gauche et peut être déclenchée par le niveau compréhension.

4.5 Précompilation

Les connaissances symboliques sont compilées avant le processus d'interprétation. Dans cette phase de précompilation nous effectuons les calculs suivants:

- Nous vérifions si le texte de la grammaire respecte la méta-syntaxe de la description de notre grammaire.

- Nous calculons, pour chaque symbole terminal de la grammaire, les listes de réalisation possibles des symboles phonémiques. Les variations phonologiques sont prises en compte. Nous obtenons ainsi un réseau ayant deux types d'entrée permettant de réaliser efficacement
  - l'accès par mot, le résultat est une liste de listes de symboles phonémiques;
  - l'accès par phonème, le résultat est l'ensemble des mots contenant ce phonème.

  Le language *Lisp* nous a facilité énormément la construction du réseau.
- Nous compilons les règles de la grammaire de façon à ce que le temps d'accès des règles à partir d'un symbole quelconque
  - en chaînage avant et
  - en chaînage arrière

  soit réalisé par un seul accès à un tableau, indépendamment du nombre de règles.
- Nous parcourons les règles pour calculer les contraintes positionnelles utilisées par notre constructeur syntaxique décrit dans le chapitre 6. C'est à dire pour chaque symbole grammatical, nous donnons le nombre maximal et le nombre minimal des terminaux qu'il peut avoir dans ses descendants.

5 Expérimentation

5.1 Corpus et apprentissage

Le corpus contient deux parties. La première partie est extraite de journaux, comprenant cinq courts articles d'environ cent phrases. Les images acoustiques de phonèmes sont selectionnées manuellement par l'inspection du spectrogramme et l'écoute du son. Les modèles phonémiques sont instanciés d'abord par ces images. Ensuite nous lançons le système de reconnaissance pour reconnaître des phrases qui vérifient la grammaire et nous examinons les résultats. Chaque fois qu'il y a un phonème mal reconnu, nous ajoutons l'image du phonème dans la liste de références. Il a fallu plusieurs réexamens et réajustements de certaines références déjà sélectionnées pour que l'ensemble des références représente bien la parole. Les techniques du clustering ne sont pas utilisées pour la sélection de références.

Les phrases sont prononcées de façon entièrement continue à une vitesse normale. La parole est échantillonée à 10kHz avec 10bits de précision et puis préaccentuée. Les paramètres utilisés sont les coefficients de prédiction linéaire de l'ordre 12 obtenus par l'application successive d'une fenêtre d'Hamming de 25.6ms avec un décalage de 10ms. Le rapport de vraisemblance pour comparer des vecteurs de paramètre est dérivé de la distance d'Itakura [Itakura 75]:

$$d(T,R) = log(\frac{a_R V_T a_R^t}{a_T V_T a_T^t})$$

où a_R et a_T sont respectivement les vecteurs de coefficients de LPC de référence et de test. V_T est la matrice d'autocorrélation du vecteur des coefficients de test. Cette distance représente

l'intégration du carré de la différence du spectre logarithmique du vecteur de référence et celui du vecteur de test [Gray 76]. Le calcul de cette distance peut se faire en $p+1$ multiplications et additions [Gray 76]. Nous avons modifié la distance de façon que l'énergie à court terme soit prise en compte [Nocerino 85] dans la comparaison. Des expériences ont montré que cette énergie est utile pour distinguer certaines copules de sons dont la séparation aurait été difficile.

Nous avons utilisé 29 symboles pour les voyelles et 20 symboles pour les consonnes. Pour entraîner le système dans un mode mono-locuteur, 300 références sont manuellement sélectionnées et plus de 10 minutes de parole sont examinées.

5.2 Evaluation

Le test corpus contient 200 phrases ne faisant pas partie du corpus d'apprentissage. Ces phrases, de longueurs variables de 5 à 19 mots, ont été enregistrées pendant une période de 5 jours.

90% des phrases sont interprêtées correctement en première proposition. 3% des phrases sont rejetées et le reste sont reconnues dans les cinq premières propositions. Le système a donné un taux de reconnaissance au niveau phonème de 99%, grâce à l'utilisation des contraintes de différents niveaux de traitement.

Nous avons également testé les paramètres MFCC [Davis 80] d'ordre 12. Dans nos conditions, les paramètres de LPC donnent un résultat un peu meilleur que MFCC.

Les erreurs sont dues aux facteurs suivants

- l'insuffisance de l'apprentissage numérique de références de phonème. Ceci nous conduira à développer des méthodes stochastiques pour réaliser un apprentissage automatique.
- l'insuffisance de la précision de la représentation numérique de la parole originale (sur 10bits).
- les confusions entre b et d, f et s, qui sont très difficiles à supprimer, en particulier par l'analyse non-descriptive telle que la notre.

Le système d'interprétation est écrit en Lelisp [Chailloux 84] et en C [Kernighan 78]. Les calculs, confiés à deux machines reliées par réseau local *Ethernet*, sont résumés par le tableau 9.13.

machine	tâche	langage	taille de programme	CPU pr. 1 phrase
masscomp(0.7Mflops)	classification	C	5300 lignes	20 secondes
sun (4Mips)	raisonnement	Lelisp	4100 lignes	100 secondes

Table 9.13: utilisation de deux machines pour l'interprétation

6 Exécution d'une interprétation

Nous donnons la trace d'un processus complet de l'interprétation d'une phrase dans l'appendice D.

7 Conclusion

L'interprétation du chinois parlé est plus difficile que d'autres langues, à cause des particularités de la langue: langue monosyllabique, nombreuses voyelles et consonnes non stationnaires, tons lexicaux. L'étude sur l'interprétation de la langue par machine est encore jeune, peu de connaissances phonétiques et phonologiques étaient disponibles sur le sujet.

Nous avons appliqué les méthodes et les techniques développées dans les chapitres précédents pour résoudre le problème. Nous avons construit un ensemble de connaissances pour l'interprétation de la parole continue. Les résultats expérimentaux obtenus sont tout à fait comparables à ceux obtenus dans le monde pour d'autres langues.

Une caractéristique importante de notre système est que les connaissances et les mécanismes d'exploration sont nettement séparés. Nous pouvons ainsi améliorer le comportement du système en mettant simplement à jour les connaissances au fur et à mesure de l'acquisition de notre compréhension sur le problème, sans modifier les algorithmes d'exploration.

Notre objectif est d'interpréter le signal incertain. Nous n'avons donc pas consacré du temps dans le traitement du bas niveau du système. Cependant l'introduction de plus de connaissances phonétiques permettrait améliorer les performances du système.

Chapitre 10

Conclusion

L'interprétation de signaux incertains reçoit de plus en plus d'attention dans le domaine de reconnaissance de formes et de l'intelligence artificielle. Les chercheurs commencent à comprendre la particularité de ce type d'interprétation. Il existe déja des méthodes et des techniques pour résoudre certains problèmes spécifiques mais une théorie générale est encore loin être formulée.

L'incertitude de la conversion du signal en symboles primitifs constitue la difficulté principale de l'interprétation. En présence d'incertitude, un processus d'interprétation doit laisser cette incertitude se propager tant qu'une information plus robuste ne permet pas de l'éliminer. Les connaissances a priori de différents niveaux d'abstraction sur le problème qui contraignent l'espace de recherche peuvent aider à supprimer l'incertitude. La représentation et l'utilisation de ces connaissances deviennent donc le problème essentiel en interprétation.

L'interprétation du signal incertain nous oblige de mener la recherche dans les directions

- de la compréhension des processus de production et de perception du signal,
- de l'étude des outils de traitement du signal,
- de la définition de modèles paramétriques pertinents du signal,
- de la construction de mécanismes de raisonnement permettant d'enlever progressivement l'incertitude, et
- de la définition d'une architecture et des stratégies de contrôle adaptées pour l'organisation des connaissances et l'interprétation.

Nos efforts de recherche dans ces aspects ont contribué à construire un système d'interprétation du signal de parole continue, – un signal incertain typique. Nos résultats montrent que l'ensemble des méthodes présentées dans cette thèse sont valables en théorie et en pratique.

Nous envisageons de continuer notre étude sur:

- l'interprétation de la parole continue:
  - Exploiter l'information sur la structure profonde de la parole: la sémantique et la pragmatique,

  - Compléter la base de connaissance phonologique du système,
  - Améliorer la qualité de la conversion du signal en symboles phonémiques;
- le modèle de la société de spécialistes:
  - Définir des modes de communication plus souple et naturel entre spécialistes et entre associations,
  - Etudier de nouveaux mécanismes de contrôles du processus d'interprétation,
  - Adapter l'architecture pour d'autres problèmes d'interprétation de signaux incertains.

Les applications potentielles de l'interprétation du signal incertain comprennent toutes les situations où une machine doit traiter les problèmes liés au monde réel ou doit manipuler des objets non-artificiels et sont donc illimitées, mais des résultats plus fiables sont encore nécessaires pour que l'interprétation automatique soit réellement utilisable industriellement.

Bibliographie

[71] —. *Chinois Fondamental.* Volume 1, Imprimerie de Commerce, Bejing, Chine, 1971.

[78] —. *Manuel du Chinois Moderne.* Volume 2, Imprimerie de Commerce, Bejing, Chine, 1978.

[Adda 87] G. Adda, M. Eskénazi, and P. E. Stern. The use of rough spectral features for large vocabulary recognition. In *Proceedings of European Conference on Speech Technology*, pages 171–174, Edinburgh, U.K., September, 1987.

[Aho 72] A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling.* Volume 1, Prentice-Hall, 1972.

[Aiello 81] N. Aiello, C. Bock, H. P. Nii, and W. C. White. *AGE reference manual.* Technical Report, HPP, Computer Science Department, Stanford University, Stanford, CA, 1981.

[Aikins 83] J. S. Aikins. Prototypical knowledge for expert system. *Artificial Intelligence*, 20:163–210, 1983.

[Anderberg 73] M. R. Anderberg. *Cluster Analysis for Applications.* Academic press, New York, 1973.

[Arthur 86] J. D. Arthur. A descriptive/predictive model for menu-based interaction. *Int. J. Man-Machine Studies*, 25:19–32, 1986.

[Atal 72] B. S. Atal. Automatic speaker recognition based on pitch contours. *J. Acoust. Soc. Amer.*, 52:1687–1697, 1972.

[Atal 78] B. S. Atal. Linear prediction analysis of speech based on a pole-zero representation. *J. Acoust. Soc. Amer.*, 64:1310–1318, 1978.

[Backer 78] E. Backer. *Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets.* Delft University Press, Delft, The Netherlands, 1978.

[Bahl 83] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. PAMI.*, PAMI-5(2):179–190, 1983.

[Baker 75a] J. K. Baker. The DRAGON system - An overview. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23(1):24–29, 1975.

[Baker 75b] J. K. Baker. *Stochastic Modeling as a Means of Automatic Speech Recognition*. PhD thesis, Carnegie-Mellon University, April 1975.

[Balzer 80] R. Balzer, L. Erman, P. London, and C. Williams. Hearsay-III a domain independent framework for expert systems. In *Proc. of AAAI 1980*, 1980.

[Barr 82] A Barr and E. A. Feigenbaum. *The Handbook of Artificial Intelligence*. Volume 1, Dept. of Computer Science, St ndford University, 1982.

[Baum 72] L. E. Baum. An inequality and associated maximation technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[Bellman 57] R. Bellman. *Dynamic Programming*. Princeton, N.F., Univ. Press, Princeton, 1957.

[Bentz 85] B. Bentz. An automatic programming system for signal processing applications. *Pattern Recognition*, 18(6):491–495, 1985.

[Bezdeck 86] J. C. Bezdeck, R. J. Hathaway, R. E. Howard, and C. A. Wilson. Coordinate descent and clustering. *Control and Cybernetics*, 15:195–204, 1986.

[Bezdek 81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, 1981.

[Bocchieri 86] E. L. Bocchieri and G. R. Doddington. Frame-specific statistical features for speaker independent speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(4):755–764, August 1986.

[Bourlard 85] H. Bourlard, Y. Kamp, H. Ney, and C. J. Wellekens. Speaker independent connected speech recognition via dynamic programming and statistical method. In M. R. Schroeder, editor, *Speech and Speaker Recognition*, Kerger, 1985.

[Bourlard 87] H. Bourlard and C. J. Wellekens. Multilayer perceptrons and automatic speech recognition. In *Proc. IEEE First Annual Int. Conf. on Neural Networks*, San Diego, California, June 1987.

[Boyer 87] A. Boyer. Application des techniques de programmation dynamique et de quantification vectorielle à la reconnaissance des mots isolés et des mots enchaînés. *Thèse de Doct. Univ. de NANCY 1*, 1987.

[Brassard 85] J. P. Brassard. Integration of segmenting and nonsegmenting approaches in continuous speech recognition. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1985*, pages 1217–1220, 1985.

[Bridle 82] J. S. Bridle, MD. Brown, and R. M. Chamberlain. An algorithm for connected word recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1982*, pages 899–902, Paris, France, 1982.

[Brown 82] J. Brown. Controlling the complexity of menu networks. *Communications of the ACM*, 25:412–418, 1982.

[Brown 83] P. F. Brown, C. H. Lee, and J. C. Spohr. Bayesian adaptation in speech recognition. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1983*, pages 761–764, April 1983.

[Bunke 84] H. Bunke and G. Sagerer. Use and representation of knowledge in image understanding based on semantic networks. In *Proc. 7th ICPR*, pages 1135–1137, Montreal, 1984.

[Burton 76] R. Burton. *Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding System*. Technical Report 3353, BBN, 1976.

[Buzo 80] A. Buzo, A. H. Gray, R. M. Gray, and J. D. Markel. Speech coding based upon vector quantification. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28(5), 1980.

[Cadzow 80] J. A. Cadzow. High performance spectrum estimation - a new method. *IEEE trans. ASSP*, ASSP-28:524–529, October 1980.

[Carbonell 84] N. Carbonell, D. Fohr, J. P. Haton, F. Lonchamp, and J. M. Pierrel. An expert system for the automatic reading of french spectrograms. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1984*, San Diego, 1984.

[Carbonell 85] N. Carbonell, J. P. Damestoy, D. Fohr, J. P. Haton, F. Lonchamp, and J. M. Pierrel. Techniques d'intelligence artificielle en décodage acoustico-phonétique. In *Actes des $14^{ème}$ Journées d'étude sur la parole*, pages 299–303, Paris, 1985.

[Carbonell 86] N. Carbonell, J. P. Haton, D. Fohr, F. Lonchamp, and J. M. Pierrel. APHODEX, design and implementation of an acoustic-phonetic decoding expert system. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, 1986.

[Carre 84] R. Carré, R. Descout, M. Eskénazi, J. Mariani, and M. Rossi. The french language database : defining, planning and recording a large database. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1984*, San Diego, 1984.

[Chailloux 84] J. Chailloux, M. Devin, and J.-M. Hullot. Lelisp: A portable and efficient Lisp system. In *1984 ACM Symposium on Lisp and Functional Programming*, Austin, Texas, 1984.

[Charniak 85] E. Charniak and D. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley Publishing Company. 1985.

[Charpillet 85] F. Charpillet, J. P. Haton, and J. M. Pierrel. Un système de reconnaissance de parole continue pour la saisie de textes lus. In *Actes du 5ème congrès AFCET R.F.I.A.*, Grenoble, 1985.

[Chen 72] M. Chen. *Nasals and Nalization in Chinese: explorations in phonological universals*. PhD thesis, University of California, Berkeley, 1972.

[Chen 73] C. H. Chen. *Statistical Pattern Recognition*. Hayden Book Co., Rochelle Park, N.J., 1973.

[Chen 87] C. H. Chen. Statistical pattern recognition – early development and recent progress. *Int. J. of Pattern Recognition and Artificial Intelligence*, 1(1):43–51, 1987.

[Chien 76] Y. T. Chien. Interactive pattern recognition: techniques and system. *IEEE Trans. Computers*, 9(5), 1976.

[Cohen 82] J. R. Cohen. A pitch measurement algorithm for speech. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1982*, pages 176–179, 1982.

[Cole 83] R. A. Cole, R. M. Stern, M. S. Phillips, S. M. Brill, P. Specker, and A. P. Pilant. Feature-based speaker independent recognition of english letters. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1984*, October 1983.

[Cooley 65] J. W. Cooley. An algorithm for machine computation of complex Fourier series. *Meth. Comput.*, 19:297–301, 1965.

[Davis 80] S. B. Davis and P. Mermelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28(4):357–366, August 1980.

[Demichelis 83] P. Demichelis, R. DeMori, P. Laface, and M. O'Kane. Computer recognition of plosive sounds using contextual information. *IEEE Trans. ASSP*, ASSP-31(2), 1983.

[DeMori 80] R. DeMori and P. Laface. Use of fuzzy algorithms for phonetic and phonemic labelling of continous speech. *IEEE Trans. PAMI.*, PAMI-2:136–148, 1980.

[DeMori 83] R. DeMori. *Computer models of speech using fuzzy algorithms*. Plenum, 1983.

[DeMori 85a] R. DeMori, P. Laface, and Y. Mong. Parallel algorithms for syllable recognition in continous speech. *IEEE. Trans. PAMI.*, PAMI.7(1), 1985.

[DeMori 85b] R. DeMori and D. Probst. Knowledge-based computer recognition of continuous speech. In M. R. Schroeder, editor, *Speech and Speaker Recognition*, Kerger, 1985.

[DeMori 86] R. DeMori and D. Probst. *Computer Recognition of Speech*, chapter 20, pages 499–526. Academic Press, 1986.

[Devijver 74] P. A. Devijver. On a new class of bounds on bayes risk in multihypothesis pattern recognition. *IEEE Trans. Computers*, 23, 1974.

[Diday 76] E. Diday and J. C. Simon. Clustering analysis. In *Digital Pattern Recognition*, Springer-Verlag, 1976.

[Doumeingts 83] G. Doumeingts, D. Breuil, and L. Pun. *La gestion de production assistée par ordinateur*. Hermes Publishing (France), 1983.

[Dove 83] W. Dove, C. Myers, A. Oppenheim, R. Davis, and G. Copec. Knowledge based pitch detection. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1983*, pages 1348–1351, Boston, 1983.

[Dubes 81] R. Dubes and A. K. Jain. Clustering methodologies in exploratory data analysis. In M. C. Yovits, editor, *Advances in Computers*, pages 113–228, Academic press, New York, 1981.

[Dubnowski 76] J. J. Dubnowski, R. W. Schafer, and R. W. Rabiner. Real-time digital hardware pitch detector. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:2–8, Feb. 1976.

[Duda 73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, 1973.

[Duda 76] R. Duda, P. Hart, and N. Nilsson. *Subjective bayesian methods for rule-based inference systems*. Technical Report 124, SRI International, Menlo Park, 1976.

[Duifhuis 82] H. Duifhuis, L. F. Willems, and R. J. Sluyter. Measurement of pitch in speech: an implementation of Goldstein's theory of pitch perception. *J. Acoust. Soc. Am.*, 1568–1580, June. 1982.

[Dunn 73] J. C. Dunn. A fuzzy relative of the ISODATA processes and its use in detecting compact, well-separated clusters. *J. Cybernet*, 3:32–57, 1973.

[Durfee 87] E. H. Durfee, V. R. Lesser, and D. D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Trans. on Computers*, C-36(11):1275–1291, 1987.

[Erman 80] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. *Computer Surveys*, 12(2):213–253, 1980.

[Erman 81] L. D. Erman, P. E. London, and S. F. Fickas. The design and an example use of hearsay-III. In *Proc. of 7th IJCAI*, pages 409–415, Vancouver, BC, 1981.

[Everitt 74] B. Everitt. *Cluster Analysis*. Wiley, New York, 1974.

[Farreny 85] H. Farreny. *Les Systèmes-experts: principes et exemples*. CEPADUES Editions, 1985.

[Feustel 82] C. D. Feustel and L. G. Shapiro. The nearest neighbor problem in an abstract metric space. *Pattern Recognition Letters*, 1, December 1982.

[Findler 79] N. V. Findler. *Associative Networks: Representation and Use of knowledge by computer*. Academic Press, New York, 1979.

[Flanagan 72] J. L. Flanagan. *Speech Analysis, Synthesis and Perception*. Springer-Verlag, 2nd ed, New York, 1972.

[Foglein 86] J. Foglein. Objective functions in fuzzy clustering. In *Proc. Int. Conf. on Pattern Recognition 1986*, Paris, France, 1986.

[Fohr 86] D. Fohr. APHODEX : Un système expert en décodage acoustico-phonétique de la parole continue. *Thèse de Doct. Univ. de NANCY 1*, 1986.

[Foley 75] D. H. Foley and W. Sammon. An optimal set of discriminant vectors. *IEEE Trans. on Computers*, March 1975.

[FornyJr 73] G. D. FornyJr. The Viterbi algorithm. In *Proc. IEEE*, pages 268–278, Mar 1973.

[Francopoulo 86] G. Francopoulo. Introduction of grammar rules. In *Proc. of 2nd International Conference on Artificial Intelligence*, IIRIAM, Marseille, France, 1986.

[Friedman 77] D. H. Friedman. Pseudo maximum likelihood pitch extraction. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-25:213–221, June. 1977.

[Fu 70] K. S. Fu, P. J. Min, and T. J. Li. Feature selection in pattern recognition. *IEEE Trans. System Science and Cybernetics*, 6(1), 1970.

[Fu 77] K. S. Fu. Error-correcting parsing for syntactic pattern recognition. In A. Klinger, K. S. Fu, and T. L. Kunii, editors, *Data Structures, Computer Graphics, and Pattern Recognition*, pages 449–492, Academic Press, Inc., 1977.

[Fu 80] K. S. Fu and Y. S. Yu. *Statistical Pattern Classification Using Contextual Information*. Research Studies Press-Wiley, 1980.

[Fu 86] K. S. Fu. *Syntactic Pattern Recognition*, chapter 4, pages 85–117. Academic Press, 1986.

[Fukunaca 75] K. Fukunaca and P. Narendra. A branch and bound algorithm for computing K-nearest neighbors. *IEEE Trans. on Computers*, July 1975.

[Furui 86] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(1):53–59, 1986.

[Gersho 82] A. Gersho. On the structure of vector quantizers. *IEEE Trans. on Information Theory*, IT-28, Mar 1982.

[Goldstein 73] J. L. Goldstein. An optimum processor for the central formation of pitch of complex tones. *J. Acoust. Soc. Am.*, 54:1496–1516, 1973.

[Goldstein 78] J. L. Goldstein, A. Gerson, P. Srulovicz, and M. Fursr. Verification of the optimal probabilistic basis of aural processing in pitch of complex tones. *J. Acoust. Soc. Am.*, 63:486–497, Feb. 1978.

[Gong 83] Y. Gong. *Conception et Réalisation d'un Système de Transformée de Fourier Rapide (FFT)*. Rapport du DEA Génie Electrique et Instrumentation, Département d'Electronique, Université de Pierre et Marie CURIE (Paris VI), Oct. 1983.

[Gong 85a] Y. Gong. *Introduction au Traitement du Signal pour la Représentation Paramétrique en Reconnaissance Automatique de la Parole*. Cours DEA Informatique, Département Mathématique Appliquée et Informatique, Université de Nancy I, 1985.

[Gong 85b] Y. Gong and J. P. Haton. Assia, Un éditeur "intelligent" pour la manipulation et l'analyse du signal vocal. In *Actes des* $14^{ème}$ *Journées d'Etudes sur la Parole*, Paris, 1985.

[Gong 85c] Y. Gong and J. P. Haton. Manipulation et analyse de signaux sous Unix : l'éditeur Assia. In *Une Achitecture Nouvelle et ses Applications, Actes des Journées Sm90*, Agence de l'Informatique, Paris, 1985.

[Gong 86a] Y. Gong and J. P. Haton. A knowledge based system for contextually deformed pattern interpretation applied to chinese tone recognition. In *Proc. of 2nd International Conference on Artificial Intelligence*, pages 521–530, IIRIAM, Marseille, France, 1986.

[Gong 86b] Y. Gong and J. P. Haton. Un système à base de connaissances pour la reconnaissance automatique des tons du chinois. In *Actes des* 15^{eme} *Journées d'Etudes sur la Parole*, Aix en Provence, 1986.

[Gong 87a] Y. Gong and J. P. Haton. Multiple level specialist society for signal interpretation (in French). In *Actes du* $6^{ème}$ *congrès Reconnaissance*

des Formes et Intelligence Artificielle, pages 245–258, AFCET, INRIA, Antibes, France, Nov. 1987.

[Gong 87b] Y. Gong and J. P. Haton. Phoneme based continuous speech recognition without pre-segmentation. In *Proceedings of European Conference on Speech Technology*, pages 121–124, Edinburgh, U.K., September, 1987.

[Gong 87c] Y. Gong and J. P. Haton. Time domain harmonic matching pitch estimation using time-dependent speech modeling. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35(10):1386–1400, October 1987.

[Gong 88] Y. Gong and J. P. Haton. A specialist society for continuous speech understanding. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1988*, New York City, April 1988.

[Gosling 81] J. Gosling. *Unix Emacs - Emacs user's Manual.* Technical Report, Carnegie-Mellon University, December 1981.

[Gray 76] A. H. Jr. Gray and J. D. Markel. Distance measures for speech processing. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:380–391, 1976.

[Gray 84] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, April 1984.

[Grenier 83] Y. Grenier. Time-dependent arma modeling of non-stationary signals. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-31(4):899–911, August 1983.

[Gubrynowicz 82] R. Gubrynowicz. Application de la théorie des sous-ensembles flous à l'analyse et la reconnaissance automatique de la parole. *Rapport DT/LAA/TSS/RCP/101*, 1982.

[Gubrynowicz 86] R. Gubrynowicz, K. Marasek, and W. W. Wiezlak. Reconnaissance de mots isolés par la méthode descriptive de traits phonétiques. *Actes des 15ème Journées d'Etudes sur la Parole*, 235–238, 1986.

[Hall 73] P. A. V. Hall. Equivalence between and/or graphs and context-free grammars. *Commun. Assoc. Comput. Machinery*, 16:444–445, july 1973.

[Hall 83] M. G. Hall, A. V. Oppenheim, and A. S. Willsky. Time-varying parametric modeling of speech. *Signal Processing*, 5:267–285, 1983.

[Halle 85] P. Hallé. Les tons du chinois de Pékin, leur comportement en parole continue. In *Actes des 14ème Journées d'Etudes sur la Parole*. GALF, PARIS, 1985.

[Hanson 78] A. Hanson and E. Riseman. Vision: A computer system for interpreting scenes. In Hanson Riseman, editor, *Computer Vision Systems*, New York, 1978.

[Haton 76] J. P. Haton and R. Mohr. A new parsing algorithm for imperfect patterns and its applications. In *3th Int. Jt. Conf. Pattern Recognition*, San Diego, USA, 1976.

[Haton 79a] J. P. Haton and O. Morel. Automatic recognition of connected digits sequences by means of segmentation and dynamic programming. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1979*, Washington DC., April 1979.

[Haton 79b] M. C. Haton and J. P. Haton. Sirene, a system for speech training of deaf people. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1979*, Washington DC, 1979.

[Haton 84] J. P. Haton. Knowledge-based and expert systems in automatic speech recognition. In R. DeMori, editor, *New Systems and Architectures for Automatic Speech Recognition and Synthesis*, D. Reidel, 1984.

[Haton 85] J. P. Haton. Intelligence artificielle en compréhension de la parole : État des recherches et comparaison avec la vision par ordinateur. *TSI*, 4(3):265–287, 1985.

[HayesRoth 79] B. Hayes-Roth, F. Hayes-Roth, S. Rosenschein, and S. Cammarata. Modeling planning as an incremental opportunistic process. *Proc. 6th Int. Jt. Conf. Artificial Intelligence*, 375–383, 1979.

[HayesRoth 83] B. Hayes-Roth. *The blackboard structure: A general framework for problem solving?* HPP-83-30, Knowledge system laboratory, computer science dept. Standford University, 1983.

[HayesRoth 84] B. Hayes-Roth. *BB1: An Architecture for Blackboard Systems, that control, explain and learn about their own behavior.* Technical Report HPP-84-16, Stanford University, Stanford, CA, 1984.

[HayesRoth 85] B. Hayes-Roth. Blackboard structure for control. *J. of Artificial Intelligence*, 26:251–321, 1985.

[HayesRoth 86] B. Hayes-Roth, A. Garvey, M. Vaughan, J. Jr., and M. Hewett. *A Layered Environment for Reasoning about Action.* Technical Report KSL 86-38, Knowledge System Laboratory, Stanford University, Stanford, CA, August 1986.

[Hess 83] W. J. Hess. *Pitch Determination of Speech Signals - Algorithms and Devices.* Springer Berlin, 1983.

[Hinton 84] G. H. Hinton, T. J. Sejnowsky, and D. H. Ackley. *Boltzmann Machines: Constraint Satisfaction Networks that learn.* CMU-CS-84-119, Carnegie-Mellon University, 1984.

[Hinton 85] G. E. Hinton. Learning in parallel networks. *Byte*, 265–273, April 1985.

[Hoskins 85] J. Hoskins. Large vocabulary speech recognition - Today at IBM. *Speech Technology*, 3(1):16–21, August-September 1985.

[Hunt 80] M. J. Hunt, M. Lenning, and P. Mermelstein. Experiments in syllable-based recognition of continuous speech. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1980*, pages 880–883, April 1980.

[Itakura 75] F. Itakura. Minimum production residual principle applied to speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23:67–72, February 1975.

[Jain 86] A. K. Jain. *Cluster Analysis*, chapter 2, pages 33–57. Academic Press, 1986.

[Jambu 78] M. Jambu. *Classification Automatique pour l'analyse des données.* Volume I : Méthodes et Algorithmes, Dunod, 1978.

[Jayez 82] J. H. Jayez. *Compréhension Automatique du Langage Naturel - le cas du groupe nominal en français.* Masson, 1982.

[Jelinek 76] F. Jelinek, R. L. Mercer, and L. R. Bahl. Continuous speech recognition: statistical methods. *C.S.R. group, IBM T.J.*, 1976.

[Johnson 84] D. H. Johnson. Signal processing software tools. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1984*, page 8.6, 1984.

[Kailath 67] T. Kailath. The divergence and bahattachayya distance mesures in signal selection. *IEEE Trans. Communication Technology*, 15, 1967.

[KamgarParsi 85] B. Kamgar-Parsi and L. N. Kanal. An improved branch and bound algorithm for computing K-nearest neighbors. *Pattern Recognition Letters*, 3, Jan. 1985.

[Kanal 86] L. N. Kanal and G. R. Dattayreya. *Problem solving methods for pattern recognition*, chapter 6, pages 143–165. Academic Press, 1986.

[Kay 85] M. Kay. *An overview of Natural Language Understanding.* Xerox Palo Alto Research Center and CSLI Stanford University, 1985.

[Kayser 84] D. Kayser. Représentation les connaissances: pourquoi? comment? In *Actes du séminaire "Dialogue homme-machine à composante orale"*, pages 155–174, 1984.

[Kernighan 78] B. W. Kernighan and D. M. Richie. *The C Programming Language.* Prentice Hall, Inc., 1978.

[Kim 84] J. H. Kim, D. W. Payton, and K. E. Olin. An expert system for object recognition in natural scenes. *Proc. 1st Conf. on Application of A.I.*, 1984.

[Kittler 85] J. Kittler and D. Pairman. Contextual pattern recognition applied to cloud detection and identification. *IEEE Trans. Geoscience and Remote Sensing*, 26(6):855–863, 1985.

[Klatt 77] Klatt. Review of the ARPA speech understanding project. *J. Acoust. Soc. Amer.*, 62:1345–1366, 1977.

[Klatt 80] D. H. Klatt. Speech perception: A model of acoustic-phonetic and lexical access. In R. A. Cole, editor, *Perception and Production of Fluent Speech*, L. Erlbaum Assoc., Hillsdale, 1980.

[Kodratoff 86] Y. Kodratoff. *Leçons d'apprentissage symbolique automatique.* Cepadues - Editions, France, 1986.

[Kohonen 86] T. Kohonen. Dynamically expanding context, with application to the correction of symbol strings in the recognition of continuous speech. In *Proc. Int. Conf. on Pattern Recognition 1986*, pages 1148–1151, France, 1986.

[Kohonen 87] T. Kohonen. Micro-processor implementation of a large vocabulary speech recognizer and phonetic typewriter for Finnish and Japanese. In *Proceedings of European Conference on Speech Technology*, Edinburgh, U.K., September, 1987.

[Kopec 84] G. E. Kopec. The integrated signal processing system ISP. *IEEE Acoust., Speech, Signal Processing*, ASSP-32:842–851, August 1984.

[Kopec 85] G. Kopec and M. A. Bush. Network-based isolated digit recognition using vector quantification. *IEEE Acoust., Speech, Signal Processing*, ASSP-33(4), 1985.

[Kordi 87] K. Kordi, C. Xydeas, and M. Holt. Printed character recognition using Markov models. In *Actes du 11ème Colloque GRETSI*, pages 507–509, Nice, Juin 1987.

[Kosko 87] B. Kosko. Constructing an associative memory. *Byte*, 137–144, September 1987.

[Kunt 80] M. Kunt. *Traitement Numérique des Signaux.* Edition Georgi, 1980.

[Laasri 87] H. Lâasri, B. Maître, and J. P. Haton. Atome: another tool for multi-expert sysyems (in French). In *Actes du 6ème congrès Reconnaissance des Formes et Intelligence Artificielle*, pages 749–759, AFCET, INRIA, Antibes, France, Nov. 1987.

[Larousse 66] Larousse. *Dictionnaire Encyclopédique - 3 Volumes en Couleur.* Librairie Larousse, Paris, France, 1966.

[Laver 82] J. Laver, S. Hiller, and R. Hanson. Comparative performance of pitch detection algorithms on dysphonic voices. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1982*, pages 192–195, 1982.

[Lea 75] W. A. Lea, M. F. Medress, and T. E. Skinner. A prosodically guided speech understanding strategy. *IEEE. Acoust., Speech, Signal Processing*, ASSP-23(1):30–38, Feb. 1975.

[Lee 86] P. W. Chi Lee and M. H. Peron. La perception des tons du chinois par des francophones et par des chinois. *Actes des 15ème Journées d'Etudes sur la Parole*, 143–145, 1986.

[Lesser 75] V. R. Lesser and et al. Organisation of the Hearsay-II speech understanding system. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23:11–23, 1975.

[Lesser 77] V. R. Lesser and L. D. Erman. A retrospective view of the Hearsay-II architecture. *Proc. 5th Int. Jt. Conf. on Artif. Intell.*, 1977.

[Levinson 83a] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction of the theory of probabilistic functions of a Markov process to automatic speech recognition. *BSTJ*, 62(4), 1983.

[Levinson 83b] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. Speaker independent isolated digit recognition using hidden Markov models. *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1049–1052, 1983.

[Lieberman 63] P. Lieberman. Some acoustic measures of normal and pathologic larynges. *J. Acoust. Soc. Am.*, 35:344–353, 1963.

[Linde 80] Y. Linde, A. Buzo, and R. M. Gary. An algorithm for the vector quantizer design. *IEEE. Trans. on Communication*, com. 28(1), Jan. 1980.

[Lindsay 80] R. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* New York, McGraw-Hill, 1980.

[Linggard 8?] R. Linggard and F. J. Owens. Pole/zero modeling of speech spectra. *Speech Communication*, 1:125–133, 1982.

[Liu 85] Hsi-Ho Liu. A rule-based system for automatic seismic discrimination. *Pattern Recognition*, 8(6):459–463, 1985.

[Lockwood 85] Ph. Lockwood. Accès rapide dans un dictionnaire de mots. In 5ème *Congrès Reconnaissance des Formes et Intelligence Artificielle*, pages 975–982, Grenoble, Nov 1985.

[Lockwood 86] P. Lockwood. A low cost DTW-based discrete utterance recogniser. In 8th *Int. Conf. on Pattern Recognition 1986*, pages 467–469, Paris, October 1986.

[Lonchamp 84] F. Lonchamp. *Les Sons du Français — Analyse acoustique descriptive.* Cours de Phonnètique. Institut de Phonétique, Université de Nancy II, 1984.

[Lowerre 76] B. T. Lowerre. *The Harpy Speech Recognition System.* PhD thesis, Dep. Computer Science, Carnegie-Mellon University, Pittsburg PA, 1976.

[Lowerre 77] B. T. Lowerre. Dynamic speaker adaptation in the HARPY speech recognition system. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1977*, pages 788–790, Hartford, 1977.

[Lowerre 80] B. Lowerre and R. Reddy. The HARPY speech understanding system. In W. A. Lea, editor, *Trends in speech recognition*, pages 340–360, Prentice-Hall Signal processing series, 1980.

[Maksym 83] J. N. Maksym, A. J. Bonner, C. N. Dent, and G. L. Hemphill. Machine analysis of acoustical signals. *Pattern Recognition*, 16(6):615–625, 1983.

[Mann 79] W. C. Mann. Design for dialogue comprehension. *17th Annu. Meeting Assoc. Computational Linguistics*, 1979.

[Mari 79] J. F. Mari. Contribution à l'analyse syntaxique et à la recherche lexicale en reconnaissance du discours continu. *Thèse de 3ème cycle, Université de Nancy 1*, 1979.

[Mariani 87] J. Mariani. Les technologies de reconnaissance automatique de la parole. In Rocquencourt INRIA, editor, *Research and developpment in language processing*, pages 115–143, Institute National de Recherche en Informatique et en Automatique, Paris, France, 1987.

[Markel 71] J. D. Markel. FFT pruning. *IEEE Trans. Audio Electroacoustic.*, Au-19(4):305–311, December 1971.

[Markoul 73] J. Markoul. Spectral analysis of speech by linear prediction. *IEEE Trans. on Audio Electroacoustic*, AU-21(3):140–148, June 1973.

[Martin 82] Ph. Martin. Comparison of pitch detection by cepstrum and spectral comb analysis. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1982*, pages 180–183, 1982.

[Martino 84] J. Di Martino. Contribution à la reconnaissance globale de la parole : mots isolés et mots enchaînés. *Thèse de 3ème cycle, Université de Nancy 1*, 1984.

[Maser 86] K. R. Maser. Automatical interpretation of sensor data for evaluating in-site conditions. In *Proc. 1st Int. Conf. on Applic. of A.I. in Engg. Problems*, pages 861–886, 1986.

[Masini 85] G. Masini, E. Thirion, and R. Mohr. Stratégie de perception pour un modèle hiérarchique. In *Actes du 5ème congrès RFIA, Tome-2*, pages 631–640, AFCET-INRIA, Grenoble, France, Novembre 1985.

[McClelland 81] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception part-I, an account of basic findings. *Psychological Review*, 88:375–407, 1981.

[Michalski 84] R. S. Michalski and R. E. Stepp. Learning from observation: conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning - An Artificial Intelligence Approach*, Spring-Verlag, 1984.

[Miclet 83] L. Miclet and M. Dabouz. Approximative fast NN recognition. *Pattern Recognition Letters*, 1, 1983.

[Miclet 84] L. Miclet and M. Dabouz. *Un Vocodeur à classification : transmission de parole à très faible débit par quantification vectorielle du spectre.* Technical Report 84D003, Département Système et Communication, ENST, Paris, France, Nov 1984.

[Miller 70] R. L. Miller. Performance characteristics of an experimental harmonic identification pitch extraction (HIPEX) system. *J. Acoust. Soc. Am.*, 47(6):1593-1601, 1970.

[Miller 74] P. L. Miller. A locally organised parser for spoken input. *Communication of ACM*, 17(11):621–630, 1974.

[Miller 81] J. L. Miller. Effect of speaking rate on segmental distinctions. In P. D. Eimas and J. L. Miller, editors, *Perspective on the Study of Speech*, Lawrence Erlbaum Associates, publishiers, 1981.

[Minsky 69] M. L. Minsky and S. Papert. *Perceptrons.* MIT press, 1969.

[Morikawa 82] H. Morikawa and H. Fujisaki. Adaptive analysis of speech based on a pole-zero representation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-30(1):77–87, 1982.

[Myers 81] C. Myers and L. R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-29(2):284, April 1981.

[Mylopoulos 83] John Mylopoulos and Hector Levesque. An overview of knowledge representation. In M. L. Brodie, J. Mylopoulos, and J. V. Schmidt, editors, *On Concepual Modeling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, Springer-Verlag, 1983.

[Nagao 79] M. Nagao, T. Matsuyama, and H. Mori. Structured analysis of complex photographs. *6-th Int. Jt. Conf. on A.I.*, 610–616, 1979.

[Nagao 84] M. Nagao. Control strategies in pattern analysis. *Pattern Recognition*, 17:45–56, 1984.

[Nakagawa 83] S. Nakagawa. A connected spoken word recognition method by $o(n)$ dynamic programming pattern matching algorithm. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. pages 296–299, Boston, 1983.

[Nakagawa 87] S. Nakagawa. Spoken sentence recognition by time-synchrous parsing algorithm of context-free grammar. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1987*, pages 829–832, 1987.

[Nandahakumar 85] N. Nandahakumar and J. K. Aggarwal. The artificial intelligence approach to pattern recognition - a perspective and an overview. *Pattern Recognition*, 18(6):383–389, 1985.

[Naylor 85] Chris Naylor. How to build an inference engine. In R. Forsyth, editor, *Expert Systems*, London, 1985.

[Nazif 84] A. M. Nazif and M.D. Levine. Low level image segmentation: an expert system. *IEEE Tr. PAMI*, PAMI-6(5):555–577, 1984.

[Ney 83] H. Ney. A dynamic programmation algorithm for nonlinear smoothing. *Signal Processing*, 5:163–173, 1983.

[Ney 87] H. Ney. Dynamic programming speech recognition using a context free grammar. In *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1987*, 1987.

[Niederjohn 75] R. J. Niederjohn. A mathematical formulation and comparison of zero-crossing analysis techniques which have been applied to automatic speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23(4):373–379, 1975.

[Nii 82] H. P. Nii, E. A. Feigenbaum, J. J. Anton, and A. J. Rockmore. Signal-to-symbol transformation: HASP/SIAP case study. *The AI Magazine*, 23–35, Spring 1982.

[Nii 86a] H. P. Nii. Blackboard systems: Blackboard application systems, blackboard systems from a knowledge engineering perspective. *The AI Magazine*, 82–106, August 1986.

[Nii 86b] H. P. Nii. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. *The AI Magazine*, 38–53, summer 1986.

[Nocerino 85] N Nocerino, F. K. Soong, L. R. Rabiner, and D. H. Klatt. Comparative study of several distorsion measures for speech recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1985*, pages 25–28, 1985.

[Noll 64] A. M. Noll. Short-time spectrum and "cepstrum" techniques for vocal pitch detection. *J. Acoust. Soc. Am.*, 36:269–302, Feb. 1964.

[Ong 86] K. K. Ong, R. E. Seviora, and P. Dasiewicz. Knowledge-based position estimation for a multisensor house robot. In *Proc. 1st Int. Conf. on Applic. of A.I. in Engg. Problems*, pages 119–130, 1986.

[Oppenheim 69a] A. V. Oppenheim. *Generalized Linear Filtering*, chapter 8. McGraw-Hill Book Company, New York, 1969.

[Oppenheim 69b] A. V. Oppenheim. Speech analysis–synthesis system based upon homomorphic filtering. *J. Acoust. Soc. Amer.*, 45:458–465, 1969.

[Oppenheim 75] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Prentice-Hall, Inc, 1975.

[Paliwal 82] K. K. Paliwal. On the performance of frequence-weighted cepstral coefficients in vowel recognition. *Speech Communication*, 1:151–154, may 1982.

[Paliwal 83] K. K. Paliwal and P. V. S. Rao. A synthesis-based method for pitch extraction. *Speech Communication*, 2:37–45, 1983.

[Peeling 86] S. M. Peeling, R. K. Moore, and M. J. Tomlinson. The multilayer perceptrons as a tool for speech pattern processing research. In *Proc. I.o.A. Autumn Conf. on Speech and Hearing*, 1986.

[Pierrel 81] J. M. Pierrel. Etude et mise en oeuvre de contraintes linguistiques en compréhension automatique du discours continu. *Thèse de Doct. Etat, Université de Nancy 1*, 1981.

[Pierrel 82] J. M. Pierrel. Utilisation des contraintes linguistiques en compréhension de la parole continue : le système myrtille II. *TSI*, 1(5):403–421, 1982.

[Pierrel 87] J. M. Pierrel. *Le Dialogue Oral Homme-Machine : Connaissances linguistiques, stratégies et architectures des systèmes.* Collection Langages, Calcul et Raisonnement, Meridien-Klincksieck, 1987.

[Pitrat 77] J. Pitrat. Formalism for text analysis. In *Actes du Séminaire International sur les Systèmes Intelligents de Question-Réponse et Grandes Banques de Données*, pages 11–34, IRIA (Bonas), 1977.

[Prade 87] H. Prade. Problématiques et méthodes en raisonnement approché (conférence invitée). In *Actes du 6ème congrès Reconnaissance des Formes et Intelligence Artificielle*, AFCET, INRIA, Antibes, France, Nov. 1987.

[Prager 86] R. W. Prager and T. D. Harrison. *Bolzmann machines for speech recognition*, chapter 1, pages 3–27. -, 1986.

[Pramanik 86] S. K. Pramanik. Fuzzy measures in determining seed points in clustering. *Pattern Recognition Letters*, 4(3):159–164, 1986.

[Quinton 82] P. Quinton. Utilisation des contraintes syntaxiques pour la reconnaissance de la parole continue. *TSI*, 1(3):233–248, 1982.

[Rabiner 75a] L. R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, N.J., 1975.

[Rabiner 75b] L. R. Rabiner, M. R. Sambur, and C. E. Schmidt. Applications of a nonlinear smoothing algorithm to speech processing. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23:552–557, December 1975.

[Rabiner 76] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal. A comparative study of several pitch detection algorithms. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:399–418, Oct. 1976.

[Rabiner 77] L. R. Rabiner, S. E. Levinson, A. E. Rosengerg, and J. G. Wilpon. Speaker independent recognition of isolated words using clustering techniques. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-27(4):336–349, 1977.

[Rabiner 78a] L. R. Rabiner and A. E. Rosenberg. Considerations in dynamic time warping algorithm for discrete word recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-26(6), december 1978.

[Rabiner 78b] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech.* Prentice-Hall, Englewood Cliffs, N.J., 1978.

[Rabiner 79] L. R. Rabiner. On the use of symmetry in FFT computation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-27(3):233–239, 1979.

[Rabiner 81] L. R. Rabiner and S. E. Levinson. Isolated and connected word recognition theory and selected applications. *IEEE Trans. Communication*, 29(5):621–659, 1981.

[Ramsey 86] C. L. Ramsey, J. A. Reggia, D. S. Nau, and A. Ferrentino. A comparative analysis of methods for expert systems. *Int. J. Man-Machine Studies*, 24:475–499, 1986.

[Reddy 74] R. Reddy and A. Newell. Knowledge and its representation in a speech understanding system. In Lee W. Gregg, editor. *Knowledge and Cognition*, pages 253–285, Lawrence Erlbaum Associates, publishers, 1974.

[Richie 83] G. Richie. *The implementation of a PIDGIN interpretation*, page . Ellis Horwood, 1983.

[Robson 86] D. Robson. Object-oriented software systems. *Byte*, 74–86, August 1986.

[Rochette 87] D. Rochette and A. Albarello. Transmission de la parole à très faible débit : Réalisation d'un vocodeur 800 bit/s. In *Actes du 11ème Colloque GRETSI*, pages 427–430, Nice. Juin 1987.

[Rosenberg 83] A. E. Rosenberg, L. R. Rabiner, J. Wilpon, and D. Kahn. Demisyllable-based isolated word recognition system. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-31(3):713–726, June 1983.

[Ross 74] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. L. Manley. Average magnitude difference function pitch extraction. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-22:353–362, Oct. 1974.

[Rumelhart 82] D. E. Rumelhart and J. L. McClelland. An interactive activation model of context effects in letter perception part-II, the enhancement effect and some tests and extensions to the model. *Psychological Review*, 89:60–94, 1982.

[Rumelhart 86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representation by Error Propagation*, page . Volume 1: Foundations, MIT Press, 1986.

[Ruspini 70] E. Ruspini. A new approach to clustering. *Information and Control*, 15:22–32, 1970.

[Ruspini 73] E. R. Ruspini. New experimental results in fuzzy clustering. *Information Science*, 6:273–284, 1973.

[Saitta 83] L. Saitta. Experiments in evidence composition in a speech understanding system. *Int. J. Man-Machine Studies*, 19:19–31, 1983.

[Sakoe 71] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Budapest, Hungary, 1971.

[Sakoe 79] H. Sakoe. Two-level DP-matching – a dynamic programming – based pattern matching algorithm for connected word recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-27(6):588, 1979.

[Sanchez 84] C. Sanchez, P. Hanser, and J. P. Haton. Environnement logiciel et matériel pour l'analyse acoustico-phonétique de la parole. In *Actes des 13ème Journées d'Etudes sur la Parole*, Bruxelles, 1984.

[Scaglola 85] C. Scaglola. Language models and search algorithms for real-time speech recognition. *Int. J. Man-Machine Studies*, 22:523–547, 1985.

[Schafer 79] R. W. Schafer and J. D. Markel. *Speech Analysis*. IEEE press, New York, 1979.

[Schroeder 68] M. R. Schroeder. Period histogram and product spectrum: new methods for fundamental frequency measurement. *J. Acoust. Soc. Am.*, 43:829–834. April 1968.

[Schwartz 85] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1985*, pages 1205–1208, 1985.

[Sejnowsky 86] T. J. Sejnowsky and C. R. Rosenberg. *NETtalk: A Parallel Network that Learn to Read Aloud*. JHU/EESC-86, JHU, 1986.

[Selim 84] S. Z. Selim and M. A. Ismail. Soft clustering of multidimensional data: A semi-fuzzy approach. *Pattern Recognition*, 17(5):559–568, 1984.

[Seneff 78] S. Seneff. Real-time harmonic pitch detection. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-26:358–365, August 1978.

[Serra 82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[Shikano 86] K. Shikano, K. Lee, and D. R. Reddy. Speaker adaptation via vector quantization. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1986*, April 1986.

[Shipman 82] D. W. Shipman and V. W. Zue. Properties of large lexicons: implementation for advanced isolated word recognition systems. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1982*, pages 546–549, 1982.

[Shortliffe 76] E. H. Shortliffe. *Computer-based Medical consulation: MYCIN*. American Elsevier, New York, 1976.

[Simin 83] H. A. Simin. Why should machines learn? In R. S. Michalski, editor, *Machine learning I*, Palo Alto, CA: Tioga publishing Co., 1983.

[Singleton 69] R. C. Singleton. An algorithm for computing the mixed radix fast Fourier transform. *IEEE Trans. Audio Electroacoustic.*, AU-17:93–103, June 1969.

[Slimane 87] A. Ben Slimane and B. Zouabi. Première approche de segmentation par filtrage morphologique. In *Actes des 16ème Journées d'Etudes sur la Parole*, pages 200–203, Hammamet, Tunisie, Octombre 1987.

[Sluyter 80] R. J. Sluyter, H. J. Kotmans, and A. V. Leeuwaarden. A novel method for pitch extraction from speech and a hardware model applicable to vocoder systems. *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1980*, 45–48, 1980.

[Sluyter 82] R. J. Sluyter, H. J. Kotmans, and T.A.C.M. Claasen. Improvements of the harmonic-sieve pitch extraction scheme and an appropriate method for voiced-unvoiced detection. *Proc. of Int. Conf. Acoust., Speech, Signal Processing 1982*, 188–191, 1982.

[Song 83] K. H. Song and C. K. Un. Pole-zero modeling of speech based on high-order pole model fitting and decomposition method. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-31(6):1556–1565, 1983.

[Soulie 86] F. Fogelman Soulie, P. Gallinari, S. Thiria, and Y. Le Cun. Learning in automate networks. In *Proc. of 2nd International Conference on Artificial Intelligence*, pages 347–356, IIRIAM, Marseille, France, 1986.

[Steiglitz 77] K. Steiglitz. On the simultaneous estimation of poles and zeros in speech analysis. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSp-25(3):229–234, 1977.

[Stern 83] R. M. Stern and M. J. Lasry. Dynamic speaker adaptation for isolated letter recognition using map estimation. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1983*, April 1983.

[Steven 80] K. N. Steven. Acoustic correlates of some phonetic categories. *J. Acoust. Soc. Amer.*, 68:836–842, 1980.

[Suen 82] C. Y. Suen and R. DeMori, editors. *Computer Analysis and Perception of Visual and Auditory Signals*. CRC Press, Boca Raton, FL, 1982.

[Svendsen 87] T. Svendsen and F. K. Soong. On the automatic segmentation of speech signals. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 1987*, pages 77–80, 1987.

[Tappert 77] C. C. Tappert. A Markov model acoustic component for automatic speech recognition. *Int. J. Man-Machine Studies*, 9:363–373, 1977.

[Technology 84] Signal Technology. *Summary of ILS Program*. Technical Report, Signal Technology, Inc., 1984.

[Terhardt 82] E. Terhardt, G. Stoll, and M. Seewann. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *J. Acoust. Soc. Am.*, 71(3):679–688, March 1982.

[Thomason 86] M. G. Thomason. *Syntactic Pattern Recognition: Stochastic Languages*, chapter 5, pages 119–141. Academic Press, 1986.

[Tohkura 87] Y. Tohkura. A weighted cepstral distance measure for speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35(10):1414–1422, October 1987.

[Tou 74] J. T. Tou and R. C. Gonzales. *Pattern Recognition Principles*. Addison-Wesley, 1974.

[Trivedi 86] M. M. Trivedi and J. C. Bezdeck. Low-level segmentation of aerial images with fuzzy clustering. *IEEE Trans. Systems, Man and Cybernetics*, 16(4):589–598, 1986.

[Tseng 87] H. P. Tseng, M. J. Sabin, and E. A. Lee. Fuzzy vector quantization applied to hidden Markov modeling. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1987*, pages 641–644, 1987.

[Vaissiere 83] J. Vaissière. Speech recognition – a tutorial. In F. Fallside and W. A. Woods, editors, *Computer Speech Processing*, pages 191–236, Printice Hall Int., 1983.

[Vintsyuk 68] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika (Cybernatics)*, 4(1), 1968.

[Welch 71] J. R. Welch and K. G. Salter. A context algorithm for pattern recognition and image interpretation. *IEEE Trans. Systems, Man and Cybernics*, 1:24–30, 1971.

[Willems 72] Y. Willems. *The Use of Prosodies in the Automatic Recognition of Spoken English Numbers*. PhD thesis, MIT, Department of Electrical Engineering, 1972.

[Winston 84] P. H. Winston. *Artificial Intelligence (2nd ed.)*. Addison-Wesley, Reading, Mass, 1984.

[Wise 76] J. D. Wise, J. R. Caprio, and T. W. Parks. Maximum likelihood pitch estimation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24:419–423, Oct. 1976.

[Wolf 80] J. J. Wolf and W. A. Woods. The HWIM speech understanding system. In W. A. Lea, editor, *Trends in speech recognition*, pages 316–339, Prentice-Hall Signal processing series, 1980.

[Woods 72] W. A. Woods. *An experimental parsing system for transition network grammars*, pages 113–154. Algorithmisc Press, New York, 1972.

[Woods 79] W. A. Woods. Control of syntax and semantics in continous speech understanding. In J. C. Simon, editor, *Spoken Language Generation and Understanding*, pages 337–364, D. Reidel Publishing Company, Bonas, France, 1979.

[Woods 82] W. A. Woods. Optimal search strategies for speech understanding control. *Artificial Intelligence*, 18:295–326, 1982.

[Wu 83] Y. S. Wu. A common operational software (ACOS) approach to a signal processing development system. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing 1983*, pages 1172–1175, 1983.

[Yegnanarayana 81] B. Yegnanarayana. Speech analysis by pole-zero decomposition of short-time spectra. *Signal Processing*, 3:5–17, 1981.

[Young 74] T. Y. Young and T. W. Calvert. *Classification, Estimation and Pattern Recognition*. American Elsevier, New York, 1974.

[Zadeh 65] Lotfi A. Zadeh. Fuzzy set. *Information and Control*, 8:338–353, 1965.

[Zadeh 77] L. A. Zadeh. *Fuzzy sets and their applications to pattern classification and clustering analysis*, pages 251–299. Academic Press, 1977.

[Zadeh 78] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1:3–28, 1978.

[Zelinski 83] R. Zelinski and F. Class. A segmentation algorithm for connected word recognition based on estimation principles. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-31(4):818–827, August 1983.

[Zue 82] W. V. Zue. Acoustic phonetic knowledge representation : Implications from spectrogram reading experiments. In J. P. Haton, editor, *Automatic Speech Analysis and Recognition*, D. Reidel,, 1982.

[Zue 85] V. W. Zue. The use of speech knowledge. In *Automatic Speech Recognition*, pages 1602–1615, IEEE proceedings, 1985.

Index

Appendice A

Exemples des menus dans ASSIA

Editor	Analyser	File	Macro-on	Macro-off	Run-macro	!command	Others
Mode:Set	Help	Repeat	Chdir		Refresh	Quit	Process

Figure A.1: le menu principale du système ASSIA

Buffer	Blocks	Scale	Baseline	Relate	Window Size	Polarity	Shift Size
	5	512	0	1	256	+ -	1
Open	Set Mark	Close	Label	Next	Previous	Insert	Save
Scroll	Time	Show	Window	Collect	Replace	Exit	Value
Jump	Color	Move	Adjust	Create	Kill	Shift	Search

Figure A.2: le menu editeur et son menu de paramètres par défaut (en haut)

Object		Window	DFT-N	LPC-order	Derive	Compare	Low-time
		On	512	12	On	Off	32
Fetch	Spect-DFT	Spect-LPC	Spect-CPS	Sonogram	Energy	Overlay	Save
		Print	Cepstrum	Pitch	Redraw	Exit	
Filter	Create	Move	Erase	Autoc	DFT		Others

Figure A.3: le menu analyseur et son menu de paramètres (en haut)

Add	Sub	Scale	Normalize	Derive	Energy	Pitch	Absolut
Sinus	Noise	Pulses	Filter	Add-noise	Extract	Exit	Slope
Constant							

Figure A.4: le menu du traitement de fichier

Appendice B

Extrait des règles d'interprétation des tons

```
(setq ALL-RULES
'(
;   RULE 001
; ------------------------------------------------------------------------
;  T1 with leading fricative and short duration.
;
(IF   ((in-class C "T4")
       (is-great (interval P C) 15 25)
       (is-sup (average-df C "all-part") -1.8)
       (is-inf (ratio (duration C) (average-len SIGNAL)) 0.53)
       (is-sup (ratio (pitch-value C) (average-pitch SIGNAL))   .9)
       (is-negative (average-df C "last-part")))
 THEN ((is-a C "T1"))
 ATTE 0.95
)
;   RULE 002
; ------------------------------------------------------------------------
;  large duration [24]
;
(IF  ((in-class C "T4")
      (is-sup (average-df C "middle-part") .1)
      (is-great (interval P C) 12 20)        ; the previous is not T3
      (is-between (average-df C "middle-part") -0.6 0.6)
      (is-sup (ratio (duration C) (average-len SIGNAL)) 0.9)
      (is-sup (ratio (pitch-value C) (average-pitch SIGNAL))   1.))
 THEN ((is-a C "T1"))
 ATTE 1
)
;   RULE 003
; ------------------------------------------------------------------------
; over-segmented: T1 -> T1 + T1 [28]
;
(IF ((in-class C "T1")
     (in-class N "T1")
     (is-inf (interval C N) 10)
     (is-sup (ratio (pitch-value N) (average-pitch SIGNAL)) 1)
     (is-inf (+ (duration C) (duration N)) (* 1.5 (average-len SIGNAL))))
 THEN ((is-a  C "T1"))
 ATTE 0.9
 ACT  ((merge C N))
```

```
)
;   RULE 004
; ------------------------------------------------------------------------
;  over-segmented: T3 -> T3 + T3  [64]
;
(IF     ((in-class P "T3")
         (in-class C "T3")
         (is-inf (duration C) 22)
         (is-inf (interval P C) 10)
;;       (is-inf (intensity C)  60)
         (is-inf (+ (duration C) (duration P)) (* 1.5 (average-len SIGNAL))))
 THEN ((is-a C  "T3"))
 ATTE 0.9
 ACT  ((merge P C)
       (decr POINTER))
)
;   RULE 005
; ------------------------------------------------------------------------
;  under-segmented T3 + T4 -> T2: [77]
;
(IF ((in-class C "T2")
     (is-sup (duration C) (* 1.5 (average-len SIGNAL))))
 THEN ((is-a C "T3"))
 ATTE 0.8
 ACT  ((split C))
)
;   RULE 006
; ------------------------------------------------------------------------
;   pitch detection failure [92]
;
(IF    ((in-class C "T4")
        (is-between (average-df C "middle-part") -0.2 +0.3)
        (is-between (average-df C "last-part")   -0.2 +0.3))
 THEN  ((is-a C "T1"))
 ATTE 0.8
)
;   RULE 007
; ------------------------------------------------------------------------
;  T3 + T4 -> T3 + T1:
;
(IF   ((in-class C "T1")
       (in-class P "T3")
       (is-inf (interval P C) 10)
       (is-positive (average-df C "first-part"))
       (is-inf (+ (duration C) (duration P)) (* 1.5 (average-len SIGNAL))))
 THEN ((is-a C "T4"))
 ATTE  .9
)
;   RULE 008
; ------------------------------------------------------------------------
;  T1 + T3 (small interval) -> T4 + T4: correction of the first. [154]
;  prediction:
(IF   ((in-class C "T4")
       (is-inf (interval C N) 10)
       (is-inf (duration N) 20)
       (is-a N "T3")) ; will be a stacking operation
 THEN ((is-a C "T1"))
 ATTE  0.98
)
;   RULE 009
; ------------------------------------------------------------------------
;  T1 + T3 (small interval) -> T4 + T4: correction of the second. [155]
;
(
```



```
 IF ((in-class C "T4")
     (is-sup (ratio (pitch-value P) (average-pitch SIGNAL)) 1.)
     (is-inf (duration C) 20)
     (is-inf (interval P C) 10))
 THEN ((is-a C "T3"))
 ATTE 0.9
 )
 ;   RULE 010
 ; ------------------------------------------------------------------------
 ;  pitch detection failure [14]
 ;
 (
 IF ((in-class C "T2")
     (is-inf (average-df C "first-part") -0.9))
 THEN ((is-a C "T1"))
 ATTE 0.9
 )
 ;   RULE 011
 ; ------------------------------------------------------------------------
 ;  over-segmented T3 -> T3 + T2 [168]
 ;
 (
 IF ((in-class C "T2")
     (in-class P "T3")
     (is-small (duration C) 10 20)
     (is-small (interval P C) 8 15)
     (is-small (+ (duration C)(duration P)) 50 60))
 THEN ((is-a C "T3"))
 ACT  ((merge P C)
       (decr POINTER))
 ATTE 0.9
 )
 ;  RULE 012
 ; ------------------------------------------------------------------------
 ;  T3 + T4 -> T3 + T1 [169]
 ;
 (
 IF  ((in-class C "T1")
      (in-class P "T3")
      (is-small (interval P C) 8 15)
      (is-great (average-df C "first-part") 1.6 1.8))
 THEN ((is-a C "T4"))
 ATTE 0.9
 )
 ; RULE 013
 ; ------------------------------------------------------------------------
 ; T1 + T3 -> T1 + T4 [173]
 ;
 (
 IF ((in-class C "T4")
     (in-class P "T1")
     (is-small (interval P C) 8 15)
     (is-small (ratio (pitch-value C)(pitch-value P)) 0.85 0.9))
 THEN ((is-a C "T3"))
 ATTE 0.9
 )
 ; RULE 014
 ; ------------------------------------------------------------------------
 ;   [77] split -> [77]+[78]. for the second splited [78]:
 ;
 (IF ((in-class C "T2")
      (is-a P "T3")
      (is-small (interval P C) 8 15))
 THEN ((is-a C "T4"))
```

```
ATTE 0.9
)
))
```

Appendice C

Extrait de la grammaire

```
(
SENTENCE -=- ON PHRASE OFF

ON -=- ss

OFF -=- ss

PHRASE -=-  DESCR -or- DEF -or- TASK -or- QUEST -or- DEPLACE -or- MENAGE
          -or- TRAIN

MENAGE -=- PLEASE VERBMV OBJ-ROOM ;; (VERBMV (OBJ-ROOM))

PLEASE -=- qing.3 ni.3

VERBMV -=- guan.1 shang.4 -or- da.3 kai.1 -or- gou.4 mai.3 -or- xiu.1 li.3
           -or- jian.3  cha.2

OBJ-ROOM -=- chuang.1 z-i.0 -or- leng.3 qi.4 -or- nuan.3 qi.4 -or- da.4 men.2
            -or- zhong.1 duan.1 -or- dian.4 sh=i.4 -or- shou.1 ying.1 ji.1


DESCR -=- R-OBJ SITU  COORD ;; (SITU (COORD) (R-OBJ))

SITU -=- wei.4 yu.2 -or- AT

PLACE -=- OBJ-REL -or- COORD

OBJ-REL -=- R-OBJ DET  DIR  ;; (DIR R-OBJ)

DET -=- de.0

DEF -=-  VERB-DEF VAR-NAME AS OBJECT ;; (VERB-DEF (VAR-NAME) (OBJECT))

AS -=- wei.2

VERB-DEF  -=-  she.4 -or- ding.4 yi.4 -or- zh=i.3 ding.4

VAR-NAME -=-  jia.3 -or- yi.3 -or- bing.3

OBJECT  -=-  R-OBJ -or- R-OBJ1 DIR DET R-OBJ

R-OBJ1 -=- R-OBJ

R-OBJ -=- P-OBJ -or- ADJ P-OBJ  -or- VAR-NAME
```

```
UNDEF -=- wu.4 ti.3

P-OBJ -=- UNDEF -or- zhuo.1 z-i.0 -or- qiu.2 -or- san.1 jiao.3
   -or- fang.1 kuai.4  -or- dian.4 hua.4 -or- shu.1 -or- qian.1 bi.3
    -or- ying.2  guang.1 ping.2 -or- he.2 z-i.0


ADJ -=-  hong.2 -or- lu%.4 -or- huang.2 -or- lan.2 -or- hei.1
-or- chang.2 -or-  duan.3 -or- da.4 -or- xiao.3 -or- bai.2

TASK -=-  PICK  OBJECT PUT TO PLACE  ;; (PUT (OBJECT)(PALCE))
        -or- PICK OBJECT TURN DIGTS DEGRE ;; (TURN (OBJECT) (DIGTS))
 -or- FIND OBJECT DET PARAM  ;; (FIND (PARAM (OBJECT)))
        -or-  EXECUT DIGTS  CODE INST ;; (EXECUT (DIGTS))

EXECUT -=- zh=i.2 xing.2 -or- qi.3 dong.4

CODE -=- hao.4

INST -=- cheng.2 xu.4 -or- zh=i.3 ling.4

TO -=-  dao.4

PICK -=- ba.3

FIND -=- PRINT -or- MEASURE -or- ADJUST

ADJUST -=- tiao.2  zheng.3 -or- kong.4 zh=i.4

PRINT -=-  da.3 ying.4  -or- xian.3 sh=i.4

MEASURE -=- ce.4 liang.2 -or- ji.4 suan.4

PUT -=- yi.2  -or-  fang.4

QUEST -=- OBJECT  WHAT ;; (WHAT (OBJ-REQ))
        -or- OBJECT  DET PARAM HOWMANY ;; (HOWMANY (PARAM (OBJ-REQ)))
        -or- NOW TIME  ;; (TIME)

NOW -=- xian.4 zai.2

TIME -=- ji.3 dian.3 zhong.1

HOWMANY -=- sh=i.4  duo.1 shao.3

WHAT -=- sh=i.4 shen.2 me.0

AT -=- zai.4

COORD -=-  DIGTS -or- DIGT POINT  DIGT1  ;; (DIGTS (POINT (DIGT1)(DIGT)))

POINT -=- dian.3

DIGTS -=-  DIGT -or-  DIGT DIGT1

; recursive DIGTS -=-  DIGT -or-  DIGTS DIGT

DIGT -=-  yi.1 -or- er.4 -or- san.1 -or- s-i.4 -or- wu.3 -or-
         liu.4 -or- qi.1 -or- ba.1 -or-  jiu.3 -or- ling.2

DIGT1 -=- DIGT
```



```
PARAM -=-  zuo.4 biao.1 -or- ti.3  ji.1 -or- chang.2 du.4
          -or- gao.1 du.4 -or- yan.2 se.4 -or- wei.4 zh=i.4

DIR -=-  P-ADV  SIDE -or- DIR-VERTIC

P-ADV -=-  qian.2 -or- hou.4 -or- DIR-LR -or- DIR-VERTIC
          -or- li.3 -or- wai.4

DIR-LR -=- zuo.3 -or- you.4

DIR-VERTIC -=-  shang.4 -or- xia.4

SIDE -=- bian.1 -or-  mian.4

DEPLACE -=- TOWARD P-ADV GO  DIGTS METER ;; (GO (P-DIV)(DIGTS)(METER))
          -or-  TOWARD DIR-LR TURN  COORD DEGRE ;; (TURN (DIR-LR)(DIGTS))

TURN -=- xuan.2 zhuan.3 -or- zhuan.3 dong.4

DEGRE -=- du.4

TOWARD -=- xiang.4

METER -=- mi.3 -or- gong.1 li.3

GO -=- yi.2 dong.4 -or- zou.3

;
; reservation:


TRAIN -=- INFOREQ -or- IN-OUT-TIK -or- GOTOCITY

GOTOCITY -=- I-DESI GOTO CITY

GOTO -=- qu.4 -or- dao.4

CITY -=- bei.3 jing.1 -or- nan.2 jing.1 -or- tian.1 jing.1
        -or- shang.4 hai.3 -or- guang.3 zhou.1 -or- xi.1 an.1
 -or- he.2 fei.2 -or- su.1 zhou.1 -or- cheng.2 du.1
        -or- gui.4 ling.2

INFOREQ -=- I-DESI VREQ SOME INFO -or-  FROM CT-CT -or- CT-CT

FROM -=- cong.2

CT-CT -=- CITY TO CITY1 DET INF-PARAM HOWMANY

INF-PARAM -=- DIST -or- piao.4 jia.4 -or- che.1 c-i.4

DIST -=- lu.4 cheng.2 -or- ju.4 li.2

CITY1 -=- CITY

IN-OUT-TIK -=- I-DESI OPT OBJ-TIK

OBJ-TIK -=- TICKET -or- DATE DET TICKET

TICKET -=- huo.3 che.1 piao.4 -or- che.1 piao.4 -or- zhan.4 tai.2 piao.4
          -or- zh=i.2  kuai.4 -or- pu.3 kuai.4 -or- te.4 kuai.4
          -or- wuo.4 pu.4

DATE -=- DAY HAO -or- MONTH DAY HAO -or- TIME-REL
```

```
TIME-REL -=- DAY-REL -or- DAY-REL IN-DAY-REL

DAY-REL -=- jin.1 tian.1 -or- ming.2 tian.1 -or- hou.4 tian.1
            -or- da.4 hou.4 tian.1 -or- WEEK WEEK-DAYS

IN-DAY-REL -=- zhong.1 wu.3 -or- shang.4 wu.3 -or- xia.4 wu.3
               -or- zao.3 shang.4 -or-  wan.3 shang.4

WEEK -=- xing.1 qi.1

HAO -=- hao.4

MONTH -=- NUM-MONTH MONTHSP

NUM-MONTH -=- DIGT -or- sh=i.2 -or- sh=i.2 yi.1 -or-  sh=i.2 er.4

MONTHSP -=- yue.4

DAY -=- DIGT -or- ONE-THREE DIGT

ONE-THREE -=- sh=i.2 -or- er.4 sh=i.2 -or- san.1 sh=i.2

WEEK-DAYS -=- tian.2 -or- ri.4 -or- yi.1 -or- er.4 -or- san.1 -or- s-i.4
     -or- wu.3 -or- liu.4

OPT -=- mai.3 -or- gou.4 mai.3 -or- ding.4 gou.4 -or- ding.4 -or- tui.4

I-DESI -=- I VAUX

I -=- wuo.3

VAUX -=- yao.4 -or- xiang.3

SOME -=- yi.1 xie.1 -or- dian.3

INFO -=- qing.2 kuang.4

VREQ -=- liao.3 jie.3 -or- wen.4 -or- xun.2 wen.4

)
```

Appendice D

Trace du processus d'interprétation

1 Arbres syntaxiques

L'arbre syntaxique de la phrase est présenté dans la figure D.1. L'ordre de la création des nœuds est présenté dans la figure D.2.

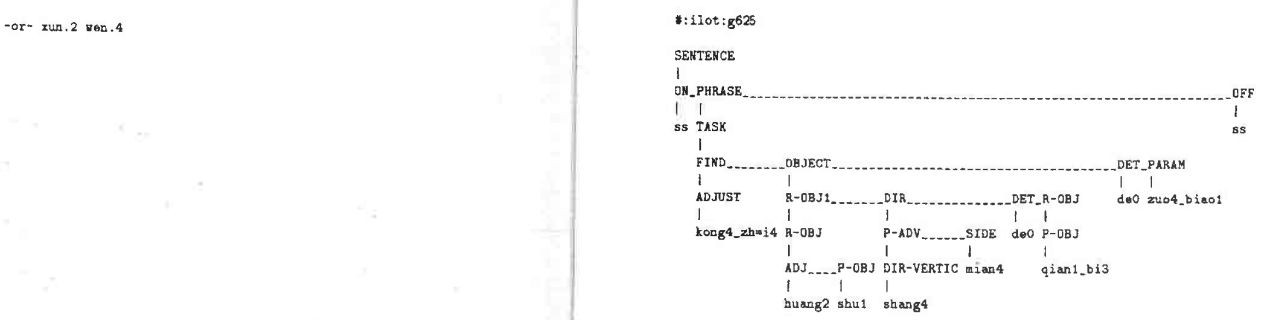


Figure D.1: Arbre syntaxique de la phrase dans l'exemple d'interprétation. (Contrôler les coordonnées du crayon au dessus du livre jaune)

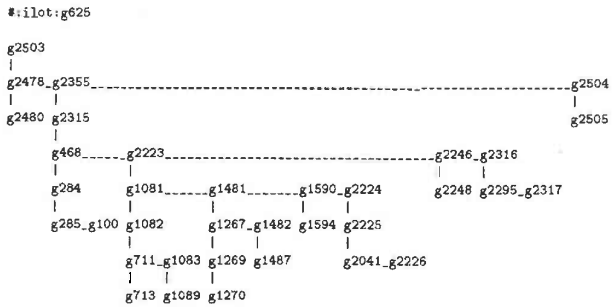


Figure D.2: Arbre syntaxique de la phrase dans l'exemple d'interprétation montrant l'ordre de création des nœuds

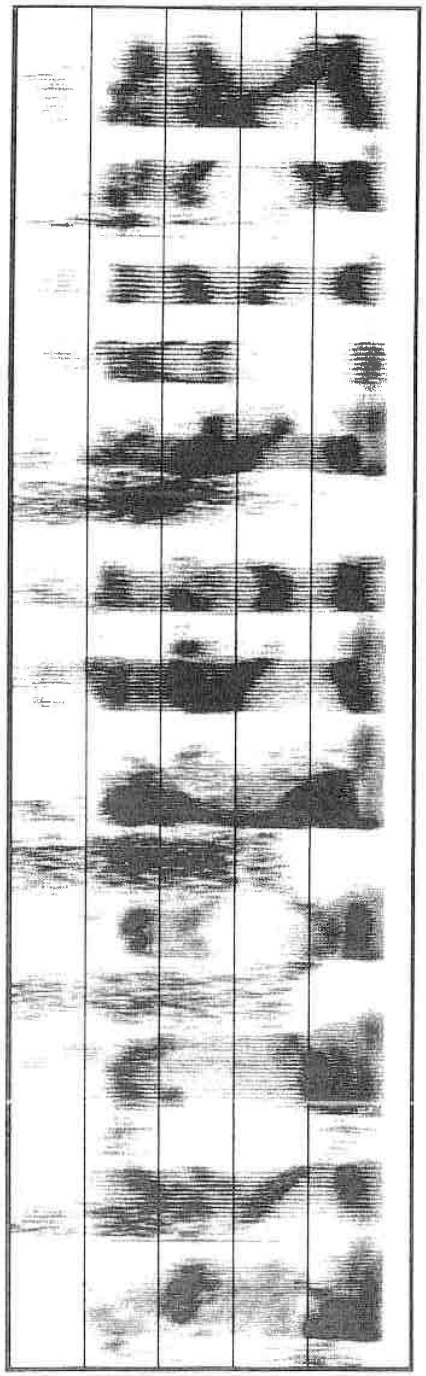
2 Spectrogramme

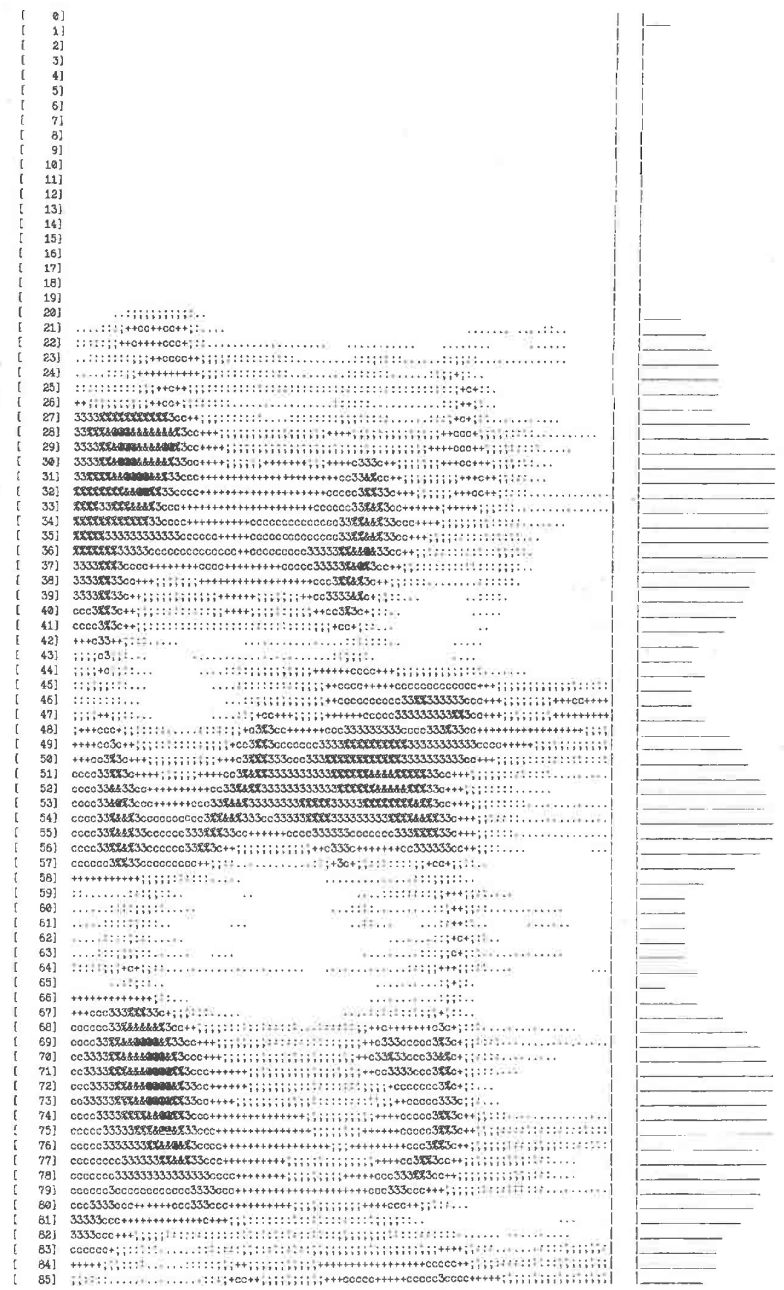
3 Spectres utilisés pour l'interprétation

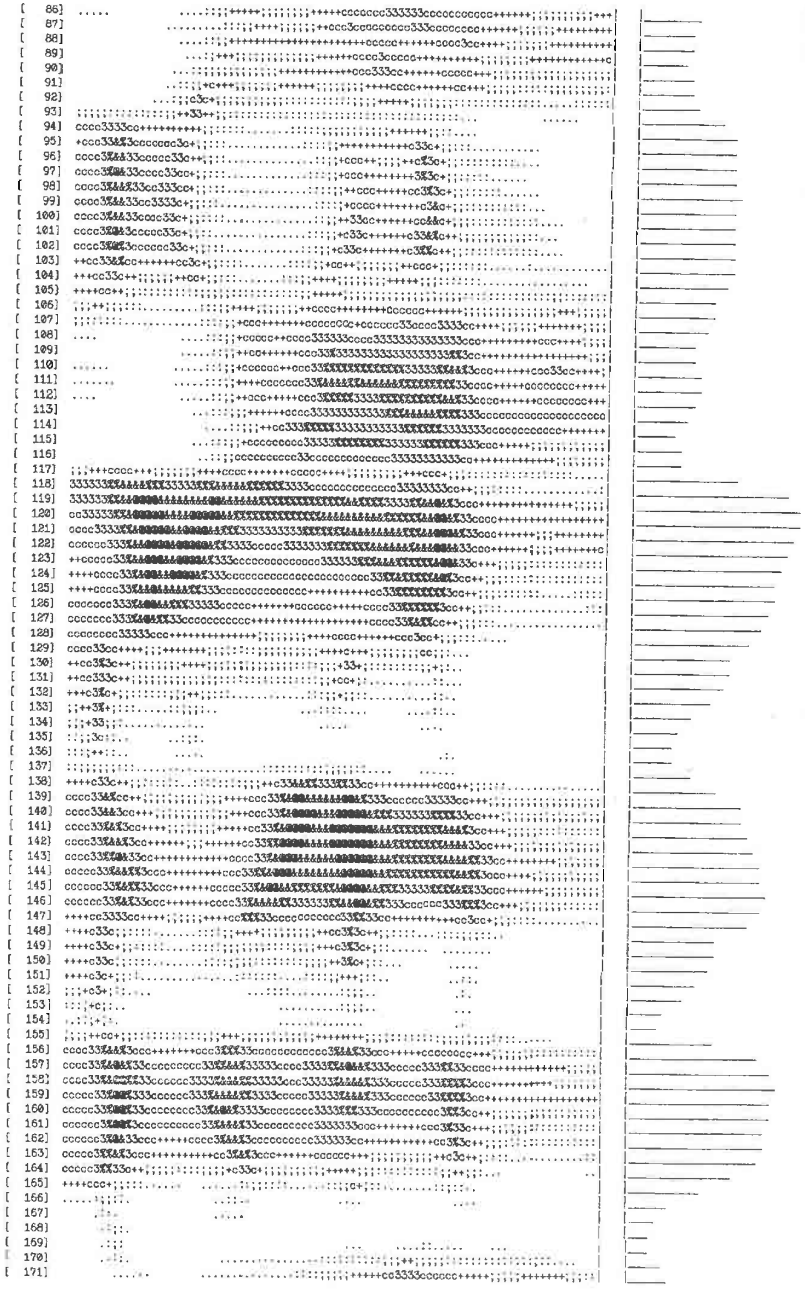
4 Liste de candidats phonémiques

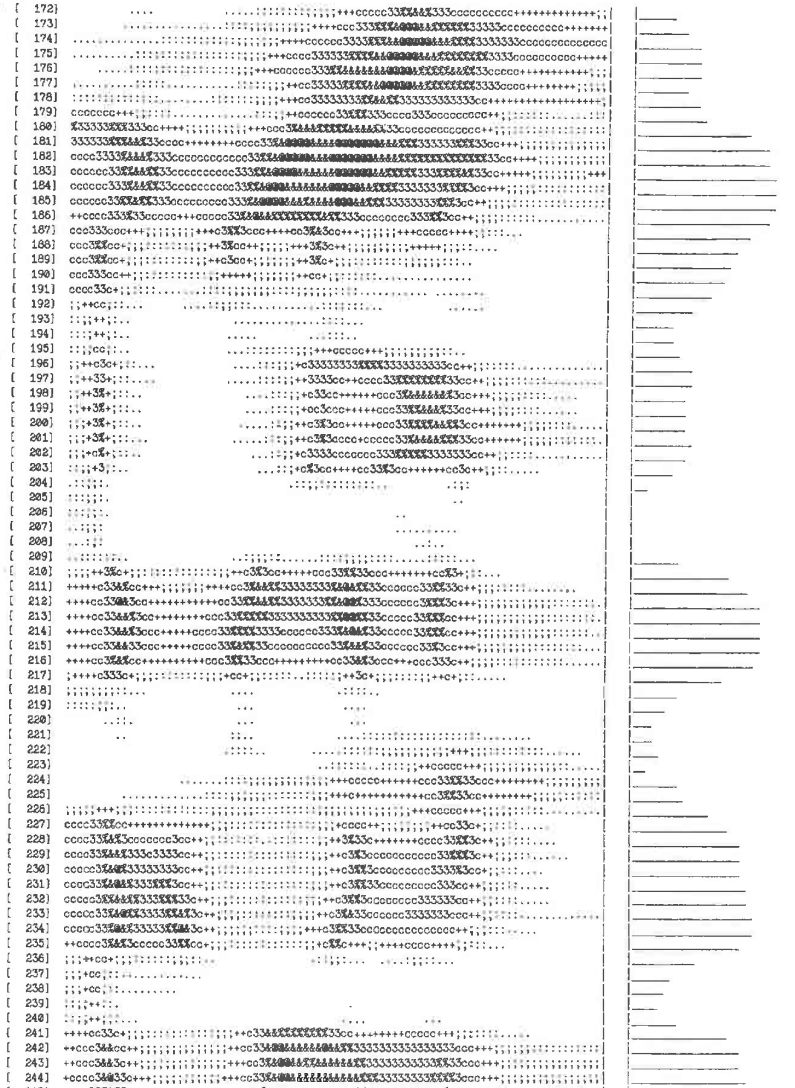
5 Courbe de localisation syllabique

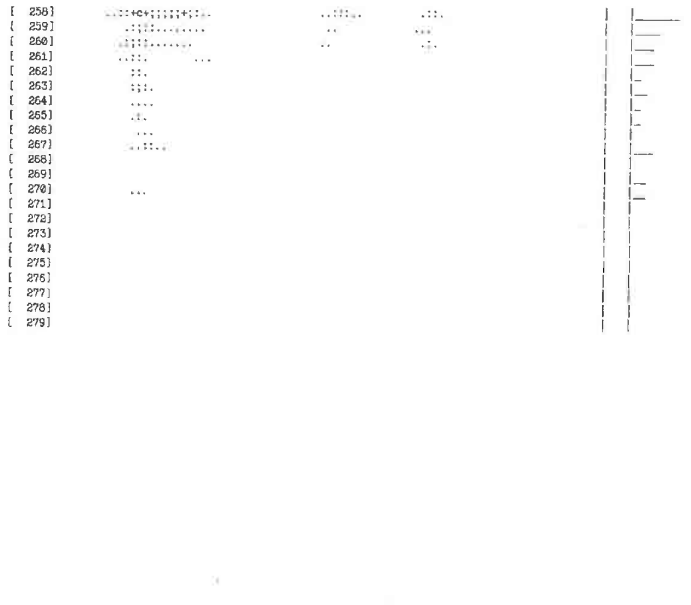
6 Trace d'exécution de la societé de spécialistes





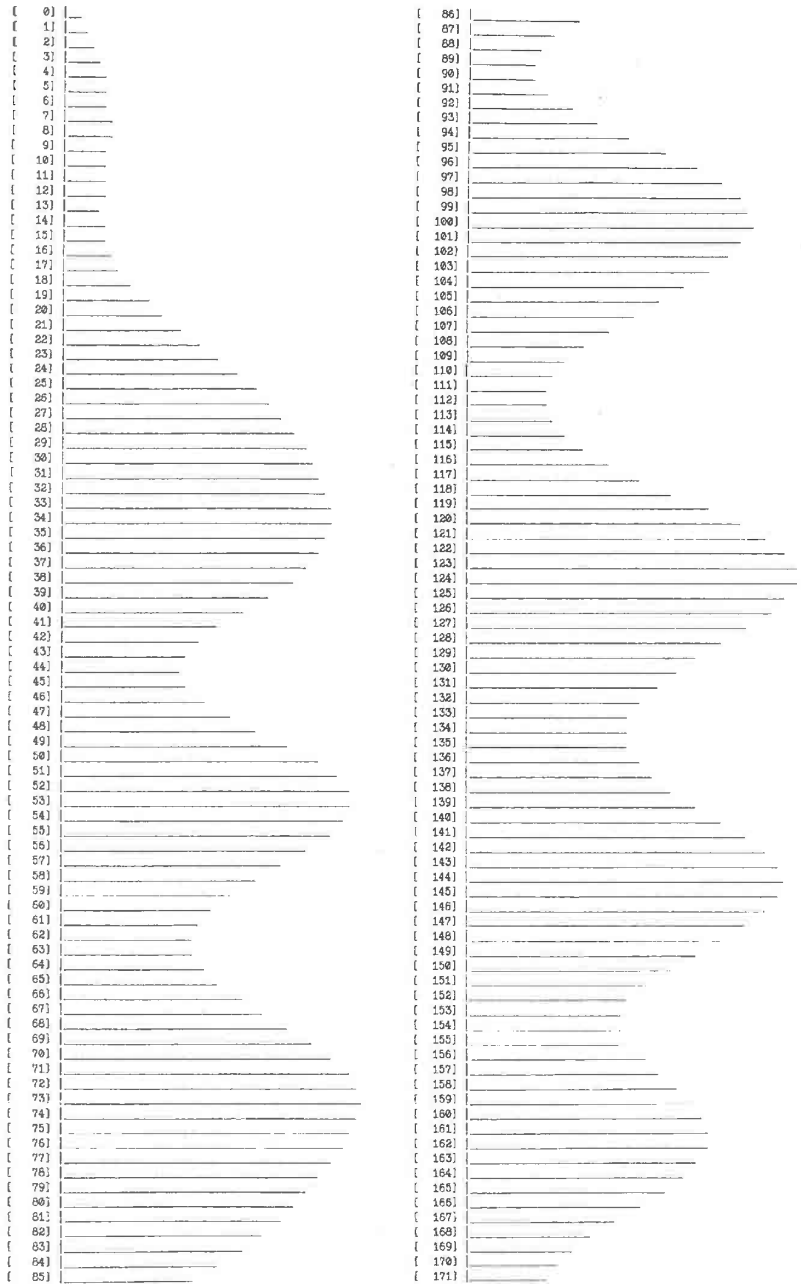


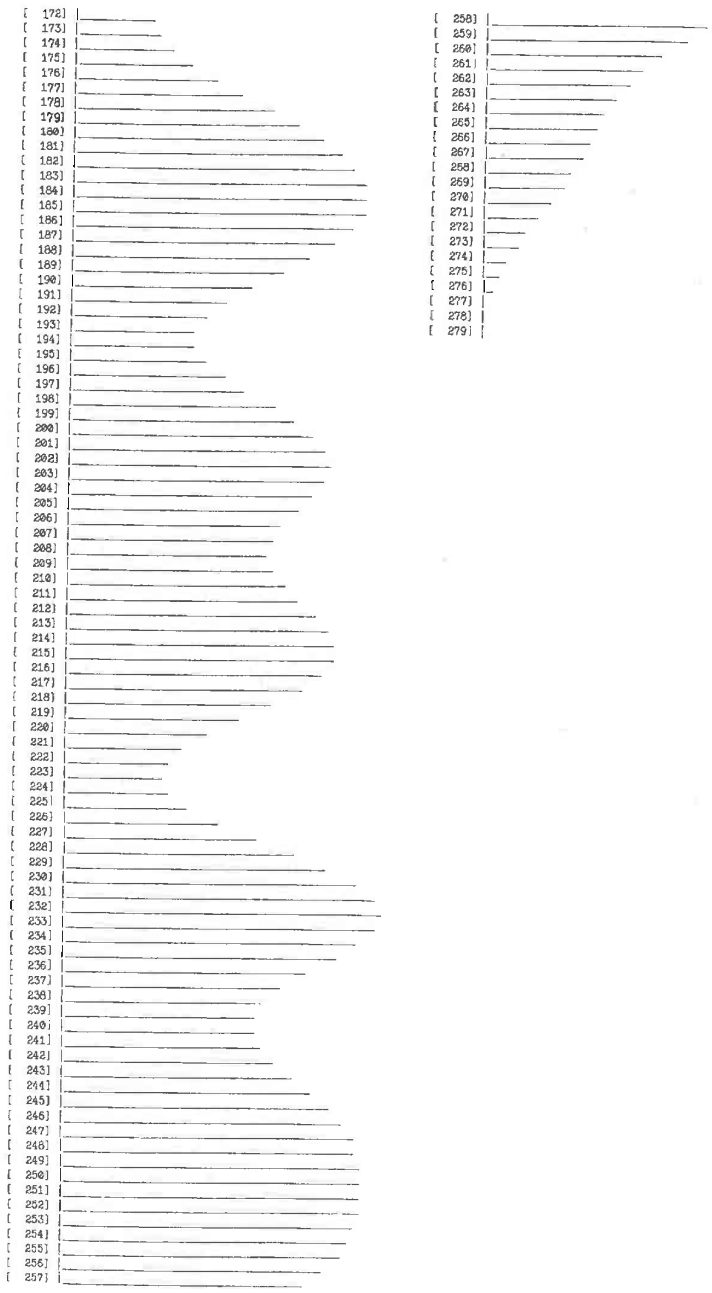





```
; Number of references = 320, maximum profile number = 4
; The recognized phoneme string:
(setq NB-SAMPLE 280 NB-CAND 4)
(setq PH-TAB '(
( 0 110.036   f 64   ss 61    b 54    h 50)
( 1   0.051   f 54   ss 61    b 54    h 50)
( 2   0.067   ss 61    h 45    f 45    b 38)
( 3   0.081   ss 62    h 42    f 35    b 33)
( 4   0.091   ss 69    h 37    b 37    f 34)
( 5   0.100   ss 72    b 43    h 35    d 34)
( 6   0.104   ss 74    b 40    h 38    d 32)
( 7   0.106   ss 78    b 44    h 39    d 33)
( 8   0.106   ss 82    b 48    d 37    h 33)
( 9   0.104   ss 75    b 47    h 38    d 34)
( 10   0.100   ss 82    b 45    h 41    d 38)
( 11   0.095   ss 77    b 49    h 40    d 36)
( 12   0.091   ss 80    b 43    h 36)
( 13   0.089   ss 77    b 50    h 34    d 32)
( 14   0.091   ss 75    b 48    h 41    d 37)
( 15   0.099   ss 78    b 50    h 44    d 37)
( 16   0.113   ss 75    b 48    h 38    d 35)
( 17   0.134   ss 85    b 42    h 35    d 32)
( 18   0.162   ss 76    b 48    h 34    d 34)
( 19   0.197   ss 68    g 48    b 46    h 35)
( 20   0.236   g 68    b 66    ss 59    k 42)
( 21   0.279   g 63    k 60    b 46    u 46)
( 22   0.323   k 69    u 57    h 50    g 44)
( 23   0.368   k 63    u 58    h 56    d 45)
( 24   0.411   h 67    k 53    d 50    u 48)
( 25   0.451   h 52    b 43    k 42    d 38)
( 26   0.487   h 41    k 39    uo 37    ou 35)
( 27   0.520   ou 44    u 41    uo 37    k 33)
( 28   0.548   ou 47    u 37    uo 34    k 34)
( 29   0.572   u 39    ou 39    ong 36    uen 36)
( 30   0.591   ong 43    uen 38    uo 37    u 35)
( 31   0.607   ong 43    uo 40    uen 40    e 37)
( 32   0.620   ong 44    uen 43    e 41    eng 40)
( 33   0.630   ong 48    uan 45    eng 43    e 40)
( 34   0.634   ong 51    eng 48    n 46    uen 44)
( 35 120.629   ong 54    eng 52    n 52    uen 41)
( 36   0.613   eng 56    ong 56    n 45    uen 43)
( 37   0.584   eng 61    ong 60    uen 46    m 45)
( 38   0.542   eng 60    ong 60    m 52    uen 46)
( 39   0.489   eng 55    ong 54    n 48    uen 48)
( 40   0.431   n 56    m 54    eng 51    ong 48)
( 41   0.374   n 62    m 59    eng 46    ong 42)
( 42   0.325   n 69    m 64    u 48    eng 42)
( 43   0.292   n 62    l 56    m 56    b 55)
( 44   0.281   l 59    d 58    n 53    t 49)
( 45   0.296   p 57    l 54    r 53    x 48)
( 46   0.334   l 54    j 52    zh 52    t 49)
( 47   0.393   =i 56    i 55    l 54    zh 48)
( 48   0.464   =i 59    l 57    i 47    iou 44)
( 49   0.539   =i 62    l 54    iou 47    ian 40)
( 50   0.604   =i 82    l 54    iou 50    i 43)
( 51   0.654   =i 78    n 49    iou 47    i 47)
( 52   0.683   =i 57    n 48    iou 46    e 46)
( 53   0.687   e 54    iou 49    n 47    ia 44)
( 54 130.670   e 60    iou 53    n 48    ian 46)
( 55   0.632   e 63    iou 50    l 48    n 47)
( 56   0.580   e 54    l 53    iou 49    ong 43)
( 57   0.521   ou 52    ong 45    e 45    l 45)
( 58   0.460   h 53    u 48    ong 48    ou 45)
( 59   0.404   d 51    h 51    u 48    ong 45)
( 60   0.356   h 54    k 49    z 48    d 46)
( 61   0.326   h 59    k 57    z 52    j 48)
( 62   0.309   k 52    h 47    ong 44    z 44)
( 63   0.311   k 50    z 46    h 46    ong 41)
( 64   0.333   h 51    k 50    d 49    z 49)
( 65   0.378   h 55    u 45    ong 40    k 39)
( 66   0.420   h 59    k 56    u 47    d 43)
( 67   0.477   ou 54    u 42    h 42    k 39)
( 68   0.536   ou 47    uo 39    u 35    ong 35)
( 69   0.592   ang 37    ou 37    h 36    ong 34)
( 70   0.639   ang 42    ao 33    ong 32    h 31)
( 71   0.676   ang 46    ong 34    ao 33    uo 31)
( 72   0.698   ang 45    ao 37    ong 36    uo 34)
( 73   0.706   ang 49    ao 39    ong 38    uo 36)
( 74 140.702   ang 52    ao 41    ong 40    uo 39)
( 75   0.689   ang 50    ao 46    ong 42    iang 42)
( 76   0.668   ang 48    ao 46    iang 45    ong 44)
( 77   0.643   ang 49    iang 48    ong 47    e 47)
( 78   0.614   ang 54    e 54    iang 53    uen 50)
( 79   0.583   iang 58    ang 56    uen 54    e 54)
( 80   0.552   u 63    r 56    n 54    iang 52)
( 81   0.515   u 66    n 50    r 49    m 49)
( 82   0.474   m 58    d 54    u 52    h 50)
( 83   0.425   t 67    e 47    l 46    h 46)
( 84   0.370   l 65    r 54    j 52    z 50)
( 85   0.312   zh 65    l 62    p 53    r 52)
( 86   0.255   p 62    z 59    zh 58    sh 57)
( 87   0.205   zh 69    sh 67    z 66    p 61)
( 88   0.165   zh 72    sh 67    z 65    p 63)
( 89   0.151   zh 76    sh 63    z 61    p 54)
( 90   0.157   zh 73    z 67    sh 61    p 55)
( 91   0.187   zh 61    z 57    p 54    sh 52)
( 92   0.240   t 51    p 50    zh 43    sh 43)
( 93   0.308   h 48    u 42    d 42    n 40)
( 94   0.386   h 51    u 50    ou 48    r 47)
( 95   0.457   u 54    ou 53    h 46    d 37)
( 96   0.542   u 57    ou 49    uo 39    h 34)
( 97   0.605   u 54    ou 45    uo 40    ong 34)
( 98   0.650   ou 49    u 48    uo 44    ong 37)
( 99   0.674   ou 54    u 46    uo 42    ong 40)
(100   0.677   ou 47    u 47    uo 44    ong 43)
(101 150.659   uo 48    ong 46    u 44    ou 43)
(102   0.624   ou 51    u 50    uo 47    ong 46)
(103   0.576   ou 55    u 49    uo 48    ong 42)
(104   0.518   h 50    u 49    ou 47    -i 47)
(105   0.456   t 54    h 51    zh 48    u 48)
(106   0.393   p 54    j 52    zh 52    h 52)
(107   0.334   p 63    zh 60    r 53    l 52)
(108   0.281   p 62    z 53    zh 51    q 48)
(109   0.237   p 59    sh 56    zh 52    z 48)
(110   0.205   sh 59    p 55    ch 47    q 47)
(111   0.186   sh 60    p 48    ch 47    j 45)
(112   0.182   sh 60    p 59    ch 52    x 45)
(113   0.196   sh 71    x 53    ch 51    p 50)
(114   0.226   sh 70    ch 54    x 50    p 45)
(115   0.273   sh 67    x 47    ch 44    p 40)
(116   0.335   sh 61    zh 42    p 42    l 36)
(117   0.409   ia 45    sh 43    e 39    iou 38)
(118   0.490   ia 51    l 49    n 40    ing 38)
(119   0.573   ia 57    ao 45    er 44    l 44)
(120   0.651   ia 64    ao 51    er 45    ang 42)
(121   0.716   ia 61    ao 57    er 50    ang 45)
(122   0.762   ao 64    ia 57    ang 50    iang 49)
(123   0.785   ao 58    ang 55    ia 52    uan 49)
(124   0.786   ang 50    ao 53    uan 51    ia 47)
(125 160.764   ang 55    uan 48    ao 48    ia 43)
(126   0.725   ang 64    e 45    uan 45    ao 43)
(127   0.672   ang 61    e 54    iang 45    uan 42)
(128   0.613   ang 57    l 51    iang 51    e 47)
(129   0.554   ang 57    h 53    l 49    ong 48)
(130   0.498   n 56    h 54    ang 53    m 51)
(131   0.450   n 62    ang 52    m 51    h 49)
(132   0.413   n 58    m 56    eng 52    uen 49)
(133   0.388   m 55    n 54    uen 53    eng 51)
(134   0.378   eng 55    m 52    b 49    uen 49)
(135   0.385   m 57    eng 51    u 48    b 48)
(136   0.407   u 56    b 56    m 53    n 48)
(137   0.445   b 59    l 56    d 55    m 52)
(138   0.494   u% 48    d 48    m 47    zh 45)
(139   0.551   i 61    ei 48    u% 47    ian 44)
(140   0.609   i 55    ian 47    u% 43    ei 43)
(141   0.663   ian 51    ei 48    i 46    =i 44)
(142   0.709   ian 55    ei 51    iou 44    uen 41)
(143   0.740   ian 59    iou 49    ei 46    uen 43)
(144   0.753   ian 62    iou 49    e 47    uen 45)
(145 170.746   ian 59    e 53    uen 47    iou 46)
(146   0.718   e 60    ian 56    iou 47    =i 47)
(147   0.672   =i 61    ian 52    eng 52    e 51)
(148   0.612   eng 58    n 51    =i 49    ian 49)
(149   0.546   eng 62    m 51    en 47    ian 45)
(150   0.482   eng 55    m 53    uen 46    n 46)
(151   0.427   n 53    eng 50    m 50    uen 49)
(152   0.388   n 53    uen 46    d 45    eng 44)
(153   0.369   l 48    n 47    d 47    uen 42)
(154   0.369   l 55    b 54    d 54    n 50)
(155   0.387   d 76    t 53    b 45    m 44)
(156   0.420   =i 57    e 54    d 54    n 51)
(157   0.461   l 55    e 51    =i bw    n 50)
(158   0.504   e 57    l 49    ai 47    uen 46)
(159   0.541   e 67    ai 51    l 51    en 50)
(160   0.568   e 72    l 63    en 54    ai 46)
(161   0.580   l 81    e 73    en 55    n 50)
(162 180.575   l 81    e 63    en 54    n 50)
(163   0.553   l 70    n 56    e 56    eng 52)
(164   0.516   n 57    en 55    l 51    eng 51)
(165   0.466   m 52    uen 49    en 49    n 47)
(166   0.410   d 59    en 49    uen 46    n 43)
(167   0.352   d 47    f 45    l 45    b 44)
```

```
(158   0.296    f 47    b 44    k 44    d 39)
(159   0.248    d 53    h 48    b 46    f 45)
(170   0.214    z 56    d 55    t 54    h 52)
(171   0.194    j 60    z 58    s 56    f 54)
(172   0.190    j 71    t 66   sh 61    s 55)
(173   0.204   sh 73    j 65   ch 61    t 53)
(174   0.234   sh 73    j 61   ch 56    t 46)
(175   0.281   sh 65    j 59   ch 44   ei 44)
(176   0.339    j 51   sh 50   ei 44   ch 43)
(177   0.404   ch 59    j 53    q 46   sh 45)
(178   0.471    j 54    t 53   sh 49    p 45)
(179   0.535    l 55   u% 49   =i 45    i 40)
(180   0.592   u% 59    i 54   =i 50    n 43)
(181   0.638    i 53  ian 42   u% 41    n 40)
(182   0.674  ian 41   ai 39   en 39  uen 38)
(183   0.696   ai 46  ian 44   en 41  uen 42)
(184   0.703  ian 48   ai 43   en 43  uen 43)
(185 190.693  ian 48   ai 47  uen 45   en 45)
(186   0.665  ian 48  uen 48   en 47  uan 44)
(187   0.621   en 49   an 46  uen 45  eng 44)
(188   0.563    l 49   en 47    m 45    n 45)
(189   0.497    m 49  eng 48    l 48   en 47)
(190   0.429  eng 53    m 52    n 51  uen 50)
(191   0.367    n 61    m 57  uen 53  eng 52)
(192   0.321    n 64    d 55    m 53    r 52)
(193   0.296    l 66    n 63    m 56    d 55)
(194   0.296    l 63    b 59    d 58    m 55)
(195   0.318    l 54    d 52    t 45    b 42)
(196   0.361    p 52    l 49    r 48    i 42)
(197   0.417    l 57    i 53   u% 49    p 36)
(198   0.480    i 58    l 55  ing 38   u% 34)
(199   0.537    i 65    l 56  ing 42   en 32)
(200   0.582    i 66    l 57  ing 45  ian 34)
(201   0.609    i 64    l 55  ing 48  ian 37)
(202   0.615    i 61    l 56  ing 44   u% 35)
(203 200.603    i 49    l 47   u% 41  ing 41)
(204   0.576  ing 43    i 41    x 40    b 37)
(205   0.540    b 54  ing 44    i 40   ie 36)
(206   0.506    b 59    i 47    d 46  ing 45)
(207   0.483    i 56    b 49  ing 41    d 40)
(208   0.476    i 47  ing 38    b 37    d 37)
(209   0.467    l 46    d 42    b 41    i 39)
(210   0.513    d 52    l 50   =i 44    p 42)
(211   0.549    e 48   u% 46   =i 45   ai 42)
(212   0.587    e 56   =i 48   ai 40   u% 39)
(213   0.620    e 61   =i 51   ai 41  eng 40)
(214   0.638    e 60   =i 51    n 44    l 44)
(215 210.635    e 62   =i 50    n 46    l 44)
(216   0.609    e 57   =i 53    n 50    l 44)
(217   0.560    e 59   an 44    n 42   =i 42)
(218   0.494    e 50    l 44    u 42    m 41)
(219   0.418    l 63    m 59    b 56    d 55)
(220   0.344    b 52    l 51    f 48    d 46)
(221   0.282    g 54    f 50    d 45    j 41)
(222   0.241    d 56    f 55    j 53   =i 49)
(223   0.228    z 56    s 53    j 48   zh 47)
(224   0.245    z 68    q 51   sh 46   ch 46)
(225   0.294    j 55   ch 54    z 54    s 50)
(226   0.364    l 48   -i 48    s 46   =i 43)
(227   0.450   -i 56    l 49    t 47    h 46)
(228   0.541    e 56   -i 53   uo 48    l 42)
(229   0.627    e 64   uo 55  iou 40   ao 39)
(230   0.697   uo 62    e 61   ao 43  iou 42)
(231   0.743   uo 69    e 58   ao 42    u 42)
(232   0.758   uo 69    e 66   ao 43  ang 41)
(233 220.743    e 77   uo 62  ang 43  iou 40)
(234   0.702    e 77   uo 55  ang 41   ou 36)
(235   0.645    e 60   uo 49   ou 45    u 42)
(236   0.581    h 47    e 42  uen 41   en 41)
(237   0.521    u 48    m 47    b 45  eng 43)
(238   0.475    m 54    b 52    u 52  eng 40)
(239   0.452    b 51    u 50    m 48    d 43)
(240   0.452    b 63    d 61    l 55    m 51)
(241   0.473   u% 54    b 44  uei 44    t 43)
(242   0.509    i 51   =i 44   u% 43  uei 40)
(243   0.554    i 49   ai 44   u% 43  ian 40)
(244   0.599    i 52   ai 49  ian 44   ao 43)
(245   0.639    i 52  ian 48   ao 47  iou 45)
(246   0.671  ian 52   ao 51  iou 50   ai 44)
(247   0.691   ao 55  iou 54  ian 48   ia 46)
(248   0.704   ao 56   ia 50    l 49  iou 49)
(249   0.711   ao 58   ia 49    l 49 iang 46)
(250   0.714   ao 58   ia 47    a 46    e 46)
(251 230.714   ao 54    a 47   ia 43 iang 39)
(252   0.711   ao 60    a 44    e 40   ia 40)
(253   0.702   ao 54    e 48    a 40   ia 36)
(254   0.686    e 48   ao 46  ang 39    a 37)
(255   0.660   ou 43    e 43   ao 42    h 41)
(256   0.634   ou 47    h 45   ao 41  ang 37)
(257   0.576   ou 42   ao 41    h 40  ang 34)
(258   0.522   ao 38    h 37    k 36  ang 31)
(259   0.466    k 52    h 35   ao 34    d 34)
(260   0.414    k 50    h 47    g 39    b 38)
(261   0.366    k 51    h 42    d 41    g 40)
(262   0.331    k 51    h 47    g 44    d 41)
(263   0.302    k 47    g 47    h 43    u 37)
(264   0.281    k 49    h 47    g 42    f 38)
(265   0.263    k 51    h 50    g 46   ss 37)
(266   0.247    h 54    g 51    k 49   ss 42)
(267   0.229    g 53    h 52    k 50   ss 41)
(268   0.209    h 56    k 53   ss 47    g 45)
(269   0.185    h 52   ss 52    g 52    f 51)
(270   0.156    f 57    h 53    k 53   ss 53)
(271   0.131   ss 58    h 49    f 44    k 40)
(272   0.103   ss 64    f 45    h 44    b 40)
(273   0.077   ss 72    h 46    f 45    b 42)
(274   0.055   ss 80    c 49   sh 49    h 46)
(275   0.037   ss 80    c 50   sh 60    q 49)
(276   0.023   ss 80   sh 70    q 70    c 70)
(277   0.013   ss 80    l 70    q 70    x 70)
(278   0.006   ss 80    l 70   zh 70    h 70)
(279 240.000   ss 80    l 70   zh 70    h 70)
)) ; end of table.
; 12 syllables, average uttering speed = 160.0 (ms/syllable)
```




```
{\scriptsize
\begin{verbatim}
? (chf ``TG/12'')
? (society)
#:date:#[1987 8 27 17 49 39 () 4]
Message from "acoustic-phon" to "lexicon" Intention: proposition
-------- Active association: lexicon
All possible starting phonemes (time = 54): (u ou ong ia -i e ian iou i ei)
#:phon:#[u 57 58 59 .4413323 -1.]
#:phon:#[ong 56 59 63 .3824987 -1.]
#:phon:#[-i 47 49 52 .5253315 -1.]
#:phon:#[e 50 53 57 .5324974 -1.]
#:phon:#[iou 50 53 57 .472249 -1.]
#:phon:#[i 47 49 51 .4239988 -1.]
#:phon:#[ei 48 49 51 .7024994 -1.]
Proposed phonemes: (#:phon:g116 #:phon:g113 #:phon:g112 #:phon:g114 #:phon:
g110 #:phon:g115 #:phon:g111)
#:phon:#[zh 45 46 47 .4569993 -1.]
Verified = #:lexc:#[zh-i.4 45 48 51 3 .5972843 -1.]
Verified = #:lexc:#[zh-i.3 45 48 51 3 .5972843 -1.]
Verified = #:lexc:#[zh-i.2 45 48 51 3 .5972843 -1.]
#:phon:#[sh 45 46 47 .378665 -1.]
Verified = #:lexc:#[sh-i.4 45 48 51 3 .5637131 -1.]
Verified = #:lexc:#[sh-i.2 45 48 51 3 .5637131 -1.]
#:phon:#[z 45 46 47 .3775992 -1.]
Verified = #:lexc:#[z-i.0 45 51 57 3 .4179311 -1.]
#:phon:#[x 44 45 46 .4060655 -1.]
#:phon:#[t 43 44 46 .4639993 -1.]
Verified = #:lexc:#[te.4 43 50 57 3 .4077315 -1.]
Verified = #:lexc:#[shen.2 45 51 57 3 .4150748 -1.]
Verified = #:lexc:#[she.4 45 51 57 3 .4150748 -1.]
#:phon:#[m 39 41 44 .5061646 -1.]
Verified = #:lexc:#[me.0 39 48 57 3 .3840508 -1.]
#:phon:#[d 41 43 46 .4034157 -1.]
Verified = #:lexc:#[de.0 41 49 57 3 .3929687 -1.]
#:phon:#[ch 46 46 47 .4249988 -1.]
Verified = #:lexc:#[che.1 46 51 57 3 .4256308 -1.]
#:phon:#[z 45 46 47 .3775992 -1.]
Verified = #:lexc:#[z-i.0 46 49 52 3 .5042257 -1.]
Verified = #:lexc:#[you.4 50 53 57 3 .472249 -1.]
#:phon:#[i 47 49 51 .4239988 -1.]
Verified = #:lexc:#[you.4 47 52 57 3 .4590888 -1.]
Verified = #:lexc:#[xiu.1 44 50 57 3 .3568702 -1.]
#:phon:#[x 44 45 46 .4060655 -1.]
Verified = #:lexc:#[xiu.1 44 50 57 3 .4477252 -1.]
#:phon:#[l 41 43 46 .5049152 -1.]
Verified = #:lexc:#[liu.4 41 49 57 3 .4004397 -1.]
#:phon:#[l 41 43 46 .5049152 -1.]
Verified = #:lexc:#[liu.4 41 49 57 3 .4752626 -1.]
#:phon:#[j 44 45 47 .4197493 -1.]
Verified = #:lexc:#[jiu.3 44 50 57 3 .3897843 -1.]
#:phon:#[j 44 45 47 .4197493 -1.]
Verified = #:lexc:#[jiu.3 44 50 57 3 .4506593 -1.]
Verified = #:lexc:#[wu.4 47 58 59 3 .4413319 -1.]
Verified = #:lexc:#[wu.3 47 58 59 3 .4413319 -1.]
#:phon:#[sh 45 46 47 .378665 -1.]
#:phon:#[p 44 45 46 .4639993 -1.]
#:phon:#[l 47 48 50 .5474987 -1.]
Verified = #:lexc:#[lu.4 47 53 59 3 .2703066 -1.]
#:phon:#[d 42 44 46 .4034986 -1.]
#:phon:#[iou 52 55 59 .434999 -1.]
Verified = #:lexc:#[you.4 47 53 59 3 .4307675 -1.]
Verified = #:lexc:#[yi.4 47 49 51 3 .4239979 -1.]
Verified = #:lexc:#[yi.3 47 49 51 3 .4239979 -1.]
Verified = #:lexc:#[yi.2 47 49 51 3 .4239979 -1.]
Verified = #:lexc:#[yi.1 47 49 51 3 .4239979 -1.]
#:phon:#[x 44 45 46 .4060655 -1.]
Verified = #:lexc:#[xiu.1 44 51 59 3 .426136 -1.]
Verified = #:lexc:#[xi.1 44 47 51 3 .417273 -1.]
#:phon:#[t 43 44 46 .4639993 -1.]
Verified = #:lexc:#[ti.3 43 47 51 3 .4417763 -1.]
#:phon:#[r 45 46 47 .4433317 -1.]
Verified = #:lexc:#[ri.4 45 48 51 3 .4295216 -1.]
#:phon:#[n 39 41 44 .5833321 -1.]
Verified = #:lexc:#[ni.3 39 45 51 3 .4323058 -1.]
#:phon:#[m 39 41 44 .5061646 -1.]
Verified = #:lexc:#[mi.3 39 45 51 3 .3956904 -1.]
#:phon:#[l 41 43 46 .5049152 -1.]
Verified = #:lexc:#[liu.4 41 50 59 3 .4541826 -1.]
Verified = #:lexc:#[li.3 47 49 51 3 .4239979 -1.]
Verified = #:lexc:#[li.3 41 46 51 3 .4681344 -1.]
Verified = #:lexc:#[li.2 47 49 51 3 .4239979 -1.]
Verified = #:lexc:#[li.2 41 46 51 3 .4681344 -1.]
Verified = #:lexc:#[jiu.3 44 51 59 3 .4287014 -1.]
Verified = #:lexc:#[ji.4 44 47 51 3 .4224048 -1.]
Verified = #:lexc:#[ji.3 44 47 51 3 .4224048 -1.]
```

```
Verified = #:lexc:#[ji.1 44 47 51 3 .4224048 -1.]
#:phon:#[b 41 42 44 .4678741 -1.]
Verified = #:lexc:#[bi.3 41 46 51 3 .3628621 -1.]
#:phon:#[zh 45 46 47 .4569993 -1.]
#:phon:#[k 44 45 46 .3616657 -1.]
#:phon:#[g 43 43 44 .4382491 -1.]
#:phon:#[d 43 44 46 .3922491 -1.]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Next main pic (L) from (3) [45] = 35
#:ilot:g102 (L) #:pred:#[44 unsticked 45 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (R) from (3) [51] = 74
diff = .08992195
--- Sticked (51,62): .03136361
#:ilot:g100 (R) #:pred:#[51 56 62 55 74 82 (ia ian q x c p j zh z g b t d k f
s h n l =i uen iang eng ao ang uo u ou ong iou -i e)]
Next main pic (L) from (3) [46] = 35
#:ilot:g105 (L) #:pred:#[44 unsticked 46 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (R) from (3) [57] = 74
#:ilot:g105 (R) #:pred:#[57 unsticked 62 57 74 82 (q x c p j zh z g b t d k f
s h l =i uen -i iang eng ao ang uo u e ong ou)]
Next main pic (L) from (3) [43] = 35
#:ilot:g106 (L) #:pred:#[43 unsticked 43 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (L) from (3) [39] = 35
#:ilot:g109 (L) #:pred:#[13 unsticked 39 16 35 39 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (L) from (3) [41] = 35
#:ilot:g110 (L) #:pred:#[41 unsticked 41 16 35 41 (m n f s p k t h d b g ss
eng e Uen ong ou uo u)]
Next main pic (R) from (3) [52] = 74
diff = .148922
--- Sticked (52,62): .03740001
#:ilot:g112 (R) #:pred:#[52 57 62 55 74 82 (ia ian q x c p j zh z g b t d k f
s h n l =i uen iang eng ao ang uo u ou ong iou -i e)]
Next main pic (L) from (3) [50] = 35
#:ilot:g113 (L) #:pred:#[44 unsticked 50 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (L) from (3) [47] = 35
#:ilot:g114 (L) #:pred:#[44 unsticked 47 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (L) from (3) [44] = 35
#:ilot:g115 (L) #:pred:#[44 unsticked 44 16 35 43 (m n f s p k t h d b g ss
eng e uen ong ou uo u)]
Next main pic (L) from (3) [57] = 35
diff = .06592274
--- Sticked (44,57): .01846156
#:ilot:g121 (L) #:pred:#[44 50 57 16 35 43 (=i i iou ian -i ia l r zh x j q z
sh ch m n f s p k t h d b g ss eng e uen ong ou uo u)]
Next main pic (R) from (3) [59] = 74
#:ilot:g121 (R) #:pred:#[59 unsticked 62 59 74 82 (q x c p j zh z k f s h g b
t d =i uen -i e iang eng ao ang uo ou ong u)]
<1, 32, 0.270>
Executing the specialist: "forward-chain-main"
New list of ilots = [79]
Executing the specialist: "backward-chain-left"
New list of ilots = [57]
Executing the specialist: "hypoth-prepare"
New list of ilots = [57]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ou 23 27 31 .3649963 -1.]
#:phon:#[ong 31 36 41 .5090885 -1.]
#:phon:#[g 17 19 22 .5143318 -1.]
Verified = #:lexc:#(gong.1 17 29 41 2 .4642367 -1.]
#:phon:#[ong 29 34 40 .4974976 -1.]
#:phon:#[g 19 21 24 .5756314 -1.]
Verified = #:lexc:#[gong.1 19 29 40 2 .4284067 -1.]
(17,29,41,2) #:node:g302 gong1 recognized
(17,29,41,2) #:node:g300 gong1 recognized
#:phon:#[u 21 23 26 .4279145 -1.]
Verified = #:lexc:#[wu.4 21 23 26 2 .4279141 -1.]
(21,23,26,2) #:node:g298 wu4 recognized
(-1,10000,57,()) #:node:g296 xia4 not found
(-1,10000,57,()) #:node:g294 zhong1 not found
(-1,10000,46,()) #:node:g292 he2 not found
#:phon:#[uo 26 28 31 .3849988 -1.]
#:phon:#[h 21 23 25 .5133314 -1.]
Verified = #:lexc:#[huo.3 21 26 31 2 .4433308 -1.]
(21,25,31,2) #:node:g289 huo3 recognized
(-1,10000,46,()) #:node:g287 he2 not found
#:phon:#[k 19 21 24 .519165 -1.]
Verified = #:lexc:#[kong.4 19 30 41 2 .4507215 -1.]
#:phon:#[k 19 21 24 .519165 -1.]
```



```
Verified = #:lexc:#[kong.4 19 29 40 2 .4129524 -1.]
(19,30,41,2) #:node:g285 kong4 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g148 (TIME) length = 1    0.422 accepted
#:ilot:g149 (MEASURE) length = 1    0.422 accepted
#:ilot:g150 (DIGT) length = 1    0.429 accepted
#:ilot:g154 (P-ADV) length = 1    0.458 accepted
#:ilot:g157 (P-ADV) length = 1    0.424 accepted
#:ilot:g159 (WEEK-DAYS) length = 1    0.454 accepted
#:ilot:g160 (DIGT) length = 1    0.454 accepted
#:ilot:g161 (METER) length = 1    0.397 accepted
#:ilot:g163 (WEEK-DAYS) length = 1    0.430 accepted
#:ilot:g164 (PARAM) length = 1    0.442 accepted
#:ilot:g166 (CITY) length = 1    0.417 accepted
#:ilot:g167 (VERBMV) length = 1    0.426 accepted
#:ilot:g168 (SOME) length = 1    0.424 accepted
#:ilot:g169 (WEEK-DAYS) length = 1    0.424 accepted
#:ilot:g171 (DIGT) length = 1    0.424 accepted
#:ilot:g172 (GO) length = 1    0.424 accepted
#:ilot:g173 (PUT) length = 1    0.424 accepted
#:ilot:g174 (VAR-NAME) length = 1    0.424 accepted
#:ilot:g176 (DIR-LR) length = 1    0.431 accepted
#:ilot:g177 (DIST) length = 1    0.270 rejected
#:ilot:g178 (WEEK-DAYS) length = 1    0.441 accepted
#:ilot:g182 (DIGT) length = 1    0.441 accepted
#:ilot:g183 (UNDEF) length = 1    0.441 accepted
#:ilot:g184 (DIGT) length = 1    0.451 accepted
#:ilot:g185 (DIGT) length = 1    0.390 accepted
#:ilot:g186 (WEEK-DAYS) length = 1    0.475 accepted
#:ilot:g187 (DIGT) length = 1    0.475 accepted
#:ilot:g188 (WEEK-DAYS) length = 1    0.400 accepted
#:ilot:g189 (DIGT) length = 1    0.400 accepted
#:ilot:g190 (VERBMV) length = 1    0.448 accepted
#:ilot:g191 (VERBMV) length = 1    0.357 accepted
#:ilot:g192 (DIR-LR) length = 1    0.459 accepted
#:ilot:g193 (DIR-LR) length = 1    0.472 accepted
#:ilot:g197 (TICKET) length = 1    0.426 accepted
#:ilot:g199 (INF-PARAM) length = 1    0.426 accepted
#:ilot:g200 (DET) length = 1    0.393 accepted
#:ilot:g202 (VERB-DEF) length = 1    0.415 accepted
#:ilot:g204 (TICKET) length = 1    0.408 accepted
#:ilot:g210 (ONE-THREE) length = 1    0.564 accepted
#:ilot:g211 (NUM-MONTH) length = 1    0.564 accepted
#:ilot:g212 (NUM-MONTH) length = 1    0.564 accepted
#:ilot:g213 (NUM-MONTH) length = 1    0.564 accepted
#:ilot:g214 (WHAT) length = 1    0.564 accepted
#:ilot:g215 (HOWMANY) length = 1    0.564 accepted
#:ilot:g218 (TICKET) length = 1    0.597 accepted
#:ilot:g219 (EXECUT) length = 1    0.597 accepted
#:ilot:g220 (INST) length = 1    0.597 accepted
#:ilot:g221 (VERB-DEF) length = 1    0.597 accepted
#:ilot:g224 (ADJUST) length = 2    0.448 accepted
#:ilot:g225 terminal rejected
#:ilot:g226 (TICKET) length = 2    0.270 accepted
#:ilot:g227 terminal rejected
#:ilot:g228 terminal rejected
#:ilot:g229 terminal rejected
#:ilot:g230 (UNDEF) length = 2    0.211 rejected
#:ilot:g231 (METER) length = 2    0.392 accepted
#:ilot:g232 (METER) length = 2    0.479 accepted
New list of ilots = [51]
Executing the specialist: "beam-search"
<1, 26, 0.357> <2, 4, 0.270>
New list of ilots = [51]
Executing the specialist: "predict-ph"
New list of ilots = [51]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (L) from (2) [17] = 0
#:ilot:g232 (L) #:pred:#[13 unsticked 17 -19 0 8 (g d t b h s f ss ss)]
Next main pic (L) from (2) [21] = 0
#:ilot:g226 (L) #:pred:#[13 unsticked 21 -19 0 8 (g d t b h s f ss ss)]
Next main pic (L) from (2) [19] = 0
#:ilot:g224 (L) #:pred:#[13 unsticked 19 -19 0 8 (g d t b h s f ss ss)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-left"
New list of ilots = [51]
Executing the specialist: "position-constrain"
#:ilot:g232(METER)  - positionally incorect
#:ilot:g231(METER)  - positionally incorect
#:ilot:g226(TICKET)  - positionally incorect
#:ilot:g220(INST)  - positionally incorect
#:ilot:g215(HOWMANY)  - positionally incorect
```

```
#:ilot:g199(INF-PARAM)  - positionally incorect
#:ilot:g188(WEEK-DAYS)  - positionally incorect
#:ilot:g186(WEEK-DAYS)  - positionally incorect
#:ilot:g178(WEEK-DAYS)  - positionally incorect
#:ilot:g169(WEEK-DAYS)  - positionally incorect
#:ilot:g163(WEEK-DAYS)  - positionally incorect
#:ilot:g161(METER)  - positionally incorect
#:ilot:g159(WEEK-DAYS)  - positionally incorect
New list of ilots = [36]
Executing the specialist: "backward-chain-right"
New list of ilots = [23]
Executing the specialist: "hypoth-prepare"
New list of ilots = [23]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ao 71 74 78 .3858242 -1.]
#:phon:#[p 63 65 57 .327599 -1.]
Verified = #:lexc:#[piao.4 63 70 78 4 .2952867 -1.]
(63,70,78,4) #:node:g309 piao4 recognized
#:phon:#[ong 56 59 62 .3785696 -1.]
#:phon:#[d 52 53 55 .3087492 -1.]
Verified = #:lexc:#[dong.4 52 57 62 3 .3531799 -1.]
#:phon:#[ou 65 68 71 .3667126 -1.]
#:phon:#[ong 71 75 79 .4027095 -1.]
#:phon:#[d 61 63 66 .3759985 -1.]
Verified = #:lexc:#[dong.4 61 70 79 4 .3857179 -1.]
#:phon:#[ong 68 73 79 .386198 -1.]
#:phon:#[d 61 63 66 .3759985 -1.]
Verified = #:lexc:#[dong.4 61 70 79 4 .3626504 -1.]
Sticking words ... (52,57,62,3) (61,70,79,4) #:node:g307 dong4 recognized
#:phon:#[ian 69 72 75 .3219986 -1.]
Verified = #:lexc:#[dian.3 61 68 75 4 .3006649 -1.]
(61,68,75,4) #:node:g304 dian3 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g224 (ADJUST) length = 2    0.448 accepted
#:ilot:g213 (NUM-MONTH) length = 1    0.564 accepted
#:ilot:g210 (ONE-THREE) length = 1    0.564 accepted
#:ilot:g202 (VERB-DEP) length = 1    0.415 accepted
#:ilot:g200 (DET) length = 1    0.393 accepted
#:ilot:g193 (DIR-LR) length = 1    0.472 accepted
#:ilot:g192 (DIR-LR) length = 1    0.459 accepted
#:ilot:g189 (DIGT) length = 1    0.400 accepted
#:ilot:g187 (DIGT) length = 1    0.475 accepted
#:ilot:g185 (DIGT) length = 1    0.390 accepted
#:ilot:g184 (DIGT) length = 1    0.451 accepted
#:ilot:g182 (DIGT) length = 1    0.441 accepted
#:ilot:g176 (DIR-LR) length = 1    0.431 accepted
#:ilot:g174 (VAR-NAME) length = 1    0.424 accepted
#:ilot:g173 (PUT) length = 1    0.424 accepted
#:ilot:g171 (DIGT) length = 1    0.424 accepted
#:ilot:g160 (DIGT) length = 1    0.454 accepted
#:ilot:g157 (P-ADV) length = 1    0.424 accepted
#:ilot:g154 (P-ADV) length = 1    0.458 accepted
#:ilot:g150 (DIGT) length = 1    0.429 accepted
#:ilot:g233 (TIME) length = 2    0.247 accepted
#:ilot:g234 (GO) length = 2    0.286 accepted
#:ilot:g236 (GO) length = 2    0.375 accepted
#:ilot:g235 (TICKET) length = 2    0.298 accepted
New list of ilots = [24]
Executing the specialist: "beam-search"
<1, 15, 0.390> <2, 5, 0.247>
New list of ilots = [24]
Executing the specialist: "predict-ph"
New list of ilots = [24]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (4) [78] = 101
#:ilot:g235 (R) #:pred:#[78 unsticked 89 82 101 109 (c ch sh q z zh r k p x j
f s g h t d b n l m -i ong uo ou =i e u)]
Next main pic (R) from (3) [62] = 74
#:ilot:g236 (R) #:pred:#[62 unsticked 62 62 74 82 (d j zh z f s g h b p t k =i
uen -i e iang eng ao ang uo ou u ong)]
Next main pic (R) from (4) [79] = 101
#:ilot:g234 (R) #:pred:#[79 unsticked 89 82 101 109 (c ch sh q z zh r k p x j
f s g h t d b n l m -i ong uo ou =i e u)]
Next main pic (R) from (4) [75] = 101
diff = .2279943
--- Sticked (75,89): .03842849
#:ilot:g233 (R) #:pred:#[75 82 89 82 101 109 (ang ao iang eng uen c ch sh q z
zh r k p x j f s g h t d b n l m -i ong uo ou =i e u)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [24]
```



```
Executing the specialist: "hypoth-prepare"
New list of ilots = [24]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ong 75 77 79 .4287967 -1.]
#:phon:#[ou 94 98 103 .4941978 -1.]
#:phon:#[ong 98 100 103 .4233322 -1.]
#:phon:#[zh 86 86 91 .6816645 -1.]
Verified = #:lexc:#[zhong.1 86 94 103 5 .4781528 -1.]
#:phon:#[ong 97 100 103 .4114275 -1.]
#:phon:#[zh 87 89 92 .6566639 -1.]
Verified = #:lexc:#[zhong.1 87 95 103 5 .4011745 -1.]
(86,94,103,5) #:node:g313 zhong1 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g235 (TICKET) length = 2    0.298 accepted
#:ilot:g236 (GO) length = 2    0.375 accepted
#:ilot:g234 (GO) length = 2    0.286 accepted
#:ilot:g150 (DIGT) length = 1    0.429 accepted
#:ilot:g154 (P-ADV) length = 1    0.458 accepted
#:ilot:g157 (P-ADV) length = 1    0.424 accepted
#:ilot:g160 (DIGT) length = 1    0.454 accepted
#:ilot:g171 (DIGT) length = 1    0.424 accepted
#:ilot:g173 (PUT) length = 1    0.424 accepted
#:ilot:g174 (VAR-NAME) length = 1    0.424 accepted
#:ilot:g176 (DIR-LR) length = 1    0.431 accepted
#:ilot:g182 (DIGT) length = 1    0.441 accepted
#:ilot:g184 (DIGT) length = 1    0.451 accepted
#:ilot:g185 (DIGT) length = 1    0.390 accepted
#:ilot:g187 (DIGT) length = 1    0.475 accepted
#:ilot:g189 (DIGT) length = 1    0.400 accepted
#:ilot:g192 (DIR-LR) length = 1    0.459 accepted
#:ilot:g193 (DIR-LR) length = 1    0.472 accepted
#:ilot:g200 (DET) length = 1    0.393 accepted
#:ilot:g202 (VERB-DEP) length = 1    0.415 accepted
#:ilot:g210 (ONE-THREE) length = 1    0.564 accepted
#:ilot:g213 (NUM-MONTH) length = 1    0.564 accepted
#:ilot:g224 (ADJUST) length = 2    0.448 accepted
#:ilot:g237 (TIME) length = 3    0.275 accepted
New list of ilots = [24]
Executing the specialist: "beam-search"
<1, 15, 0.390> <2, 4, 0.286> <3, 1, 0.275>
New list of ilots = [24]
Executing the specialist: "predict-ph"
New list of ilots = [24]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (5) [103] = 125
#:ilot:g237 (R) #:pred:#[103 unsticked 112 106 125 133 (n g d c l f s h r ch
sh q z x zh j k t p uen ong -i uan eng iang ang er ao ing iou e ia =i)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [24]
Executing the specialist: "position-constrain"
#:ilot:g237(TIME)  - positionally incorect
#:ilot:g210(ONE-THREE)  - positionally incorect
#:ilot:g234(GO)  - positionally incorect
#:ilot:g236(GO)  - positionally incorect
#:ilot:g235(TICKET)  - positionally incorect
New list of ilots = [19]
Executing the specialist: "forward-chain-main"
New list of ilots = [77]
Executing the specialist: "backward-chain-left"
New list of ilots = [64]
Executing the specialist: "hypoth-prepare"
New list of ilots = [64]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[=i 48 49 51 .7024994 -1.]
#:phon:#[sh 45 46 47 .378665 -1.]
Verified = #:lexc:#[sh=i.2 45 48 51 3 .5637131 -1.]
Verified = #:lexc:#[sh=i.2 45 46 47 3 .3786645 -1.]
(45,48,51,3) #:node:g504 sh=i2 recognized
(45,48,51,3) #:node:g508 sh=i2 recognized
(45,48,51,3) #:node:g511 sh=i2 recognized
Verified = #:lexc:#[wu.3 21 23 26 2 .4279141 -1.]
(21,23,26,2) #:node:g476 wu3 recognized
(21,23,26,2) #:node:g484 wu3 recognized
(21,23,26,2) #:node:g492 wu3 recognized
(21,23,26,2) #:node:g500 wu3 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g239 (DAY) length = 1    0.429 accepted
```

```
#:ilot:g240 (NUM-MONTH) length = 1    0.429 accepted
#:ilot:g241 (DIGT1) length = 1    0.429 accepted
#:ilot:g242 (DIGTS) length = 1    0.429 accepted
#:ilot:g243 (DIGTS) length = 1    0.429 accepted
#:ilot:g244 (COORD) length = 1    0.429 accepted
#:ilot:g246 (DIR) length = 1    0.468 accepted
#:ilot:g248 (DIR) length = 1    0.424 accepted
#:ilot:g250 (DAY) length = 1    0.454 accepted
#:ilot:g251 (NUM-MONTH) length = 1    0.454 accepted
#:ilot:g252 (DIGT1) length = 1    0.454 accepted
#:ilot:g253 (DIGTS) length = 1    0.454 accepted
#:ilot:g254 (DIGTS) length = 1    0.454 accepted
#:ilot:g255 (COORD) length = 1    0.454 accepted
#:ilot:g257 (DAY) length = 1    0.424 accepted
#:ilot:g258 (NUM-MONTH) length = 1    0.424 accepted
#:ilot:g259 (DIGT1) length = 1    0.424 accepted
#:ilot:g260 (DIGTS) length = 1    0.424 accepted
#:ilot:g261 (DIGTS) length = 1    0.424 accepted
#:ilot:g262 (COORD) length = 1    0.424 accepted
#:ilot:g264 (R-OBJ) length = 1    0.424 accepted
#:ilot:g267 (P-ADV) length = 1    0.431 accepted
#:ilot:g269 (DAY) length = 1    0.441 accepted
#:ilot:g270 (NUM-MONTH) length = 1    0.441 accepted
#:ilot:g271 (DIGT1) length = 1    0.441 accepted
#:ilot:g272 (DIGTS) length = 1    0.441 accepted
#:ilot:g273 (DIGTS) length = 1    0.441 accepted
#:ilot:g274 (COORD) length = 1    0.441 accepted
#:ilot:g276 (DAY) length = 1    0.451 accepted
#:ilot:g277 (NUM-MONTH) length = 1    0.451 accepted
#:ilot:g278 (DIGT1) length = 1    0.451 accepted
#:ilot:g279 (DIGTS) length = 1    0.451 accepted
#:ilot:g280 (DIGTS) length = 1    0.451 accepted
#:ilot:g281 (COORD) length = 1    0.451 accepted
#:ilot:g283 (DAY) length = 1    0.390 accepted
#:ilot:g284 (NUM-MONTH) length = 1    0.390 accepted
#:ilot:g285 (DIGT1) length = 1    0.390 accepted
#:ilot:g286 (DIGTS) length = 1    0.390 accepted
#:ilot:g287 (DIGTS) length = 1    0.390 accepted
#:ilot:g288 (COORD) length = 1    0.390 accepted
#:ilot:g290 (DAY) length = 1    0.475 accepted
#:ilot:g291 (NUM-MONTH) length = 1    0.475 accepted
#:ilot:g292 (DIGT1) length = 1    0.475 accepted
#:ilot:g293 (DIGTS) length = 1    0.475 accepted
#:ilot:g294 (DIGTS) length = 1    0.475 accepted
#:ilot:g295 (COORD) length = 1    0.475 accepted
#:ilot:g297 (DAY) length = 1    0.400 accepted
#:ilot:g298 (NUM-MONTH) length = 1    0.400 accepted
#:ilot:g299 (DIGT1) length = 1    0.400 accepted
#:ilot:g300 (DIGTS) length = 1    0.400 accepted
#:ilot:g301 (DIGTS) length = 1    0.400 accepted
#:ilot:g302 (COORD) length = 1    0.400 accepted
#:ilot:g304 (P-ADV) length = 1    0.459 accepted
#:ilot:g306 (P-ADV) length = 1    0.472 accepted
#:ilot:g312 (DEF) length = 1    0.415 accepted
#:ilot:g313 (MONTH) length = 1    0.564 accepted
#:ilot:g314 (FIND) length = 2    0.448 accepted
#:ilot:g316 (OBJ-TIK) length = 2    0.250 accepted
#:ilot:g317 (OBJ-TIK) length = 2    0.250 accepted
#:ilot:g316 (OBJ-TIK) length = 2    0.250 accepted
#:ilot:g315 (OBJ-TIK) length = 2    0.250 accepted
#:ilot:g321 (DAY) length = 2    0.351 accepted
#:ilot:g320 (DAY) length = 2    0.351 accepted
#:ilot:g319 (DAY) length = 2    0.351 accepted
New list of ilots = [64]
Executing the specialist: "beam-search"
<1, 14, 0.390> <2, 3, 0.250>
New list of ilots = [64]
Executing the specialist: "predict-ph"
New list of ilots = [64]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-left"
New list of ilots = [58]
Executing the specialist: "position-constrain"
New list of ilots = [58]
Executing the specialist: "backward-chain-right"
New list of ilots = [59]
Executing the specialist: "hypoth-prepare"
New list of ilots = [59]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[u 65 67 69 .3897991 -1.]
Verified = #:lexc:#[wu.3 65 67 69 4 .3897991 -1.]
(65,67,69,4) #:node:g589 wu3 recognized
```



```
#:phon:#[-1 77 78 79 .3358326 -1.]
#:phon:#[s 63 65 67 .3944988 -1.]
(57,10000,10000,()) #:node:g593 s-i4 not found
(65,67,69,4) #:node:g581 wu3 recognized
(57,10000,10000,()) #:node:g585 s-i4 not found
(65,67,69,4) #:node:g573 wu3 recognized
(57,10000,10000,()) #:node:g577 s-i4 not found
(65,67,69,4) #:node:g565 wu3 recognized
(57,10000,10000,()) #:node:g569 s-i4 not found
(65,67,69,4) #:node:g557 wu3 recognized
(59,10000,10000,()) #:node:g561 s-i4 not found
(61,68,75,4) #:node:g553 dian3 recognized
#:phon:#[iou 52 55 59 .434999 -1.]
(51,10000,10000,()) #:node:g537 jiu3 not found
(51,10000,10000,()) #:node:g541 liu4 not found
#:phon:#[u 57 58 59 .4413323 -1.]
Verified = #:lexc:#[wu.3 57 58 59 3 .4413319 -1.]
#:phon:#[u 65 67 69 .3897991 -1.]
Verified = #:lexc:#[wu.3 65 67 69 4 .3897991 -1.]
Sticking words ... (57,58,59,3) (65,67,69,4) #:node:g545 wu3 recognized
#:phon:#[-i 51 53 56 .4302483 -1.]
#:phon:#[-i 77 78 79 .3358326 -1.]
(51,10000,10000,()) #:node:g549 s-i4 not found
(65,67,69,4) #:node:g529 wu3 recognized
(59,10000,10000,()) #:node:g533 s-i4 not found
#:phon:#[b 62 63 65 .3569989 -1.]
Verified = #:lexc:#[bian.1 62 68 75 4 .2629986 -1.]
(62,68,75,4) #:node:g525 bian1 recognized
(62,68,75,4) #:node:g522 bian1 recognized
(65,67,69,4) #:node:g515 wu3 recognized
(59,10000,10000,()) #:node:g519 s-i4 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g321 (DAY) length = 2    0.351 accepted
#:ilot:g314 (FIND) length = 2    0.448 accepted
#:ilot:g306 (P-ADV) length = 1    0.472 accepted
#:ilot:g304 (P-ADV) length = 1    0.459 accepted
#:ilot:g301 (DIGTS) length = 1    0.400 accepted
#:ilot:g299 (DIGT1) length = 1    0.400 accepted
#:ilot:g298 (NUM-MONTH) length = 1    0.400 accepted
#:ilot:g297 (DAY) length = 1    0.400 accepted
#:ilot:g294 (DIGTS) length = 1    0.475 accepted
#:ilot:g292 (DIGT1) length = 1    0.475 accepted
#:ilot:g291 (NUM-MONTH) length = 1    0.475 accepted
#:ilot:g290 (DAY) length = 1    0.475 accepted
#:ilot:g287 (DIGTS) length = 1    0.390 accepted
#:ilot:g285 (DIGT1) length = 1    0.390 accepted
#:ilot:g284 (NUM-MONTH) length = 1    0.390 accepted
#:ilot:g283 (DAY) length = 1    0.390 accepted
#:ilot:g280 (DIGTS) length = 1    0.451 accepted
#:ilot:g278 (DIGT1) length = 1    0.451 accepted
#:ilot:g277 (NUM-MONTH) length = 1    0.451 accepted
#:ilot:g276 (DAY) length = 1    0.451 accepted
#:ilot:g273 (DIGTS) length = 1    0.441 accepted
#:ilot:g271 (DIGT1) length = 1    0.441 accepted
#:ilot:g270 (NUM-MONTH) length = 1    0.441 accepted
#:ilot:g269 (DAY) length = 1    0.441 accepted
#:ilot:g267 (P-ADV) length = 1    0.431 accepted
#:ilot:g264 (R-OBJ) length = 1    0.424 accepted
#:ilot:g261 (DIGTS) length = 1    0.424 accepted
#:ilot:g259 (DIGT1) length = 1    0.424 accepted
#:ilot:g258 (NUM-MONTH) length = 1    0.424 accepted
#:ilot:g257 (DAY) length = 1    0.424 accepted
#:ilot:g254 (DIGTS) length = 1    0.454 accepted
#:ilot:g252 (DIGT1) length = 1    0.454 accepted
#:ilot:g251 (NUM-MONTH) length = 1    0.454 accepted
#:ilot:g250 (DAY) length = 1    0.454 accepted
#:ilot:g243 (DIGTS) length = 1    0.429 accepted
#:ilot:g241 (DIGT1) length = 1    0.429 accepted
#:ilot:g240 (NUM-MONTH) length = 1    0.429 accepted
#:ilot:g239 (DAY) length = 1    0.429 accepted
#:ilot:g323 terminal rejected
#:ilot:g322 (DIGTS) length = 2    0.339 accepted
#:ilot:g324 (DIR) length = 2    0.252 accepted
#:ilot:g325 (DIR) length = 2    0.200 rejected
#:ilot:g327 terminal rejected
#:ilot:g326 (DIGTS) length = 2    0.365 accepted
#:ilot:g331 terminal rejected
#:ilot:g330 (DIGTS) length = 2    0.177 rejected
#:ilot:g343 (DIGTS) length = 2    0.265 accepted
#:ilot:g329 terminal rejected
#:ilot:g328 terminal rejected
#:ilot:g332 (COORD) length = 2    0.229 accepted
#:ilot:g334 terminal rejected
#:ilot:g333 (DIGTS) length = 2    0.252 accepted
```

```
#:ilot:g336 terminal rejected
#:ilot:g335 (DIGTS) length = 2    0.318 accepted
#:ilot:g338 terminal rejected
#:ilot:g337 (DIGTS) length = 2    0.285 accepted
#:ilot:g340 terminal rejected
#:ilot:g339 (DIGTS) length = 2    0.346 accepted
#:ilot:g342 terminal rejected
#:ilot:g341 (DIGTS) length = 2    0.302 accepted
New list of ilots = [48]
Executing the specialist: "beam-search"
<1, 11, 0.390> <2, 12, 0.229>
New list of ilots = [48]
Executing the specialist: "predict-ph"
New list of ilots = [48]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (4) [69] = 101
diff = .1189051
--- Sticked (69,89): .02204993
#:ilot:g341 (R) #:pred:#[69 79 89 82 101 109 (ang ao eng iang uen c ch sh q z
zh r k p x j f s g h t d b n l m -1 ong uo ou =i e u)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [49]
Executing the specialist: "hypoth-prepare"
New list of ilots = [49]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[u 80 81 82 .6033325 -1.]
Verified = #:lexc:#[wu.3 80 81 82 4 .6033325 -1.]
#:phon:#[u 93 96 99 .5014277 -1.]
Verified = #:lexc:#[wu.3 93 96 99 5 .5014267 -1.]
Sticking words ... (80,81,82,4) (93,96,99,5) #:node:g601 wu3 recognized
#:phon:#[-1 77 78 79 .3358326 -1.]
(75,10000,10000,()) #:node:g605 s-i4 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g341 (DIGTS) length = 2    0.302 accepted
#:ilot:g339 (DIGTS) length = 2    0.346 accepted
#:ilot:g337 (DIGTS) length = 2    0.285 accepted
#:ilot:g335 (DIGTS) length = 2    0.318 accepted
#:ilot:g333 (DIGTS) length = 2    0.252 accepted
#:ilot:g343 (DIGTS) length = 2    0.255 accepted
#:ilot:g326 (DIGTS) length = 2    0.365 accepted
#:ilot:g324 (DIR) length = 2    0.252 accepted
#:ilot:g322 (DIGTS) length = 2    0.339 accepted
#:ilot:g239 (DAY) length = 1    0.429 accepted
#:ilot:g240 (NUM-MONTH) length = 1    0.429 accepted
#:ilot:g241 (DIGT1) length = 1    0.429 accepted
#:ilot:g243 (DIGTS) length = 1    0.429 accepted
#:ilot:g250 (DAY) length = 1    0.454 accepted
#:ilot:g251 (NUM-MONTH) length = 1    0.454 accepted
#:ilot:g252 (DIGT1) length = 1    0.454 accepted
#:ilot:g254 (DIGTS) length = 1    0.454 accepted
#:ilot:g257 (DAY) length = 1    0.424 accepted
#:ilot:g258 (NUM-MONTH) length = 1    0.424 accepted
#:ilot:g259 (DIGT1) length = 1    0.424 accepted
#:ilot:g261 (DIGTS) length = 1    0.424 accepted
#:ilot:g264 (R-OBJ) length = 1    0.424 accepted
#:ilot:g267 (P-ADV) length = 1    0.431 accepted
#:ilot:g269 (DAY) length = 1    0.441 accepted
#:ilot:g270 (NUM-MONTH) length = 1    0.441 accepted
#:ilot:g271 (DIGT1) length = 1    0.441 accepted
#:ilot:g273 (DIGTS) length = 1    0.441 accepted
#:ilot:g276 (DAY) length = 1    0.451 accepted
#:ilot:g277 (NUM-MONTH) length = 1    0.451 accepted
#:ilot:g278 (DIGT1) length = 1    0.451 accepted
#:ilot:g280 (DIGTS) length = 1    0.451 accepted
#:ilot:g283 (DAY) length = 1    0.390 accepted
#:ilot:g284 (NUM-MONTH) length = 1    0.390 accepted
#:ilot:g285 (DIGT1) length = 1    0.390 accepted
#:ilot:g287 (DIGTS) length = 1    0.390 accepted
#:ilot:g290 (DAY) length = 1    0.475 accepted
#:ilot:g291 (NUM-MONTH) length = 1    0.475 accepted
#:ilot:g292 (DIGT1) length = 1    0.475 accepted
#:ilot:g294 (DIGTS) length = 1    0.475 accepted
#:ilot:g297 (DAY) length = 1    0.400 accepted
#:ilot:g298 (NUM-MONTH) length = 1    0.400 accepted
#:ilot:g299 (DIGT1) length = 1    0.400 accepted
#:ilot:g301 (DIGTS) length = 1    0.400 accepted
#:ilot:g304 (P-ADV) length = 1    0.459 accepted
#:ilot:g306 (P-ADV) length = 1    0.472 accepted
#:ilot:g314 (FIND) length = 2    0.448 accepted
#:ilot:g321 (DAY) length = 2    0.351 accepted
```



```
#:ilot:g345 terminal rejected
#:ilot:g344 (COORD) length = 3    0.191 rejected
#:ilot:g346 (COORD) length = 3    0.234 accepted
New list of ilots = [48]
Executing the specialist: "beam-search"
<1, 11, 0.390> <2, 11, 0.252> <3, 1, 0.234>
New list of ilots = [48]
Executing the specialist: "predict-ph"
New list of ilots = [48]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (4) [82] = 101
#:ilot:g346 (R) #:pred:#[82 unsticked 89 82 101 109 (c ch sh q z zh r k p x j
f s g h t d b n l m -1 ong uo ou =i e u)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [48]
Executing the specialist: "position-constrain"
#:ilot:g346(COORD)  - positionally incorect
#:ilot:g321(DAY)  - positionally incorect
#:ilot:g301(DIGTS)  - positionally incorect
#:ilot:g299(DIGT1)  - positionally incorect
#:ilot:g297(DAY)  - positionally incorect
#:ilot:g294(DIGTS)  - positionally incorect
#:ilot:g292(DIGT1)  - positionally incorect
#:ilot:g290(DAY)  - positionally incorect
#:ilot:g287(DIGTS)  - positionally incorect
#:ilot:g285(DIGT1)  - positionally incorect
#:ilot:g283(DAY)  - positionally incorect
#:ilot:g280(DIGTS)  - positionally incorect
#:ilot:g278(DIGT1)  - positionally incorect
#:ilot:g276(DAY)  - positionally incorect
#:ilot:g273(DIGTS)  - positionally incorect
#:ilot:g271(DIGT1)  - positionally incorect
#:ilot:g269(DAY)  - positionally incorect
#:ilot:g261(DIGTS)  - positionally incorect
#:ilot:g259(DIGT1)  - positionally incorect
#:ilot:g257(DAY)  - positionally incorect
#:ilot:g254(DIGTS)  - positionally incorect
#:ilot:g252(DIGT1)  - positionally incorect
#:ilot:g250(DAY)  - positionally incorect
#:ilot:g243(DIGTS)  - positionally incorect
#:ilot:g241(DIGT1)  - positionally incorect
#:ilot:g239(DAY)  - positionally incorect
#:ilot:g322(DIGTS)  - positionally incorect
#:ilot:g326(DIGTS)  - positionally incorect
#:ilot:g343(DIGTS)  - positionally incorect
#:ilot:g333(DIGTS)  - positionally incorect
#:ilot:g335(DIGTS)  - positionally incorect
#:ilot:g337(DIGTS)  - positionally incorect
#:ilot:g339(DIGTS)  - positionally incorect
#:ilot:g341(DIGTS)  - positionally incorect
New list of ilots = [14]
Executing the specialist: "forward-chain-main"
New list of ilots = [22]
Executing the specialist: "backward-chain-left"
New list of ilots = [18]
Executing the specialist: "hypoth-prepare"
New list of ilots = [18]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(-1,10000,41,()) #:node:g678 de0 not found
(-1,10000,47,()) #:node:g674 de0 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g349 (MONTH) length = 1    0.429 accepted
#:ilot:g350 (MONTH) length = 1    0.454 accepted
#:ilot:g351 (MONTH) length = 1    0.424 accepted
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g355 (OBJ-REL) length = 1    0.424 accepted
#:ilot:g356 (DESCR) length = 1    0.424 accepted
#:ilot:g358 (DIR) length = 1    0.431 accepted
#:ilot:g359 (MONTH) length = 1    0.441 accepted
#:ilot:g360 (MONTH) length = 1    0.451 accepted
#:ilot:g361 (MONTH) length = 1    0.390 accepted
#:ilot:g362 (MONTH) length = 1    0.475 accepted
#:ilot:g363 (MONTH) length = 1    0.400 accepted
#:ilot:g365 (DIR) length = 1    0.459 accepted
#:ilot:g367 (DIR) length = 1    0.472 accepted
#:ilot:g368 (TASK) length = 2    0.448 accepted
#:ilot:g369 terminal rejected
#:ilot:g370 terminal rejected
New list of ilots = [16]
```

```
Executing the specialist: "beam-search"
<1, 11, 0.390> <2, 1, 0.448>
New list of ilots = [16]
Executing the specialist: "predict-ph"
New list of ilots = [16]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-left"
New list of ilots = [16]
Executing the specialist: "position-constrain"
#:ilot:g355(OBJ-RKL)  - positionally incorect
New list of ilots = [15]
Executing the specialist: "backward-chain-right"
New list of ilots = [24]
Executing the specialist: "hypoth-prepare"
New list of ilots = [24]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(51,10000,10000,()) #:node:g689 jia3 not found
#:phon:#[ao 53 55 58 .3769655 -1.]
#:phon:#[ao 71 74 78 .3858242 -1.]
(51,10000,10000,()) #:node:g701 xiao3 not found
#:phon:#[ang 65 71 77 .399229 -1.]
#:phon:#[h 58 60 63 .5166645 -1.]
Verified = #:lexc:#[huang.2 58 67 77 4 .4144974 -1.]
(58,67,77,4) #:node:g713 huang2 recognized
Verified = #:lexc:#[hong.2 58 68 79 4 .4056649 -1.]
#:phon:#[h 61 63 66 .5449982 -1.]
Verified = #:lexc:#[hong.2 61 70 79 4 .416019 -1.]
(61,70,79,4) #:node:g725 hong2 recognized
#:phon:#[s 51 54 57 .5324259 -1.]
#:phon:#[e 74 76 79 .4356651 -1.]
#:phon:#[h 63 65 67 .5259981 -1.]
Verified = #:lexc:#[he.2 63 71 79 4 .3084693 -1.]
(63,71,79,4) #:node:g736 he2 recognized
(51,10000,10000,()) #:node:g748 qian1 not found
Verified = #:lexc:#[dian.4 61 68 75 4 .3006649 -1.]
(61,68,75,4) #:node:g750 dian4 recognized
#:phon:#[f 58 60 63 .4908314 -1.]
Verified = #:lexc:#[fang.1 58 67 77 4 .4067473 -1.]
(58,67,77,4) #:node:g772 fang1 recognized
(51,10000,10000,()) #:node:g784 qiu2 not found
#:phon:#[uo 65 68 71 .3373413 -1.]
#:phon:#[zh 60 61 63 .3799987 -1.]
Verified = #:lexc:#[zhuo.1 60 65 71 4 .3234482 -1.]
(60,65,71,4) #:node:g795 zhuo1 recognized
Verified = #:lexc:#[wu.4 57 58 59 3 .4413319 -1.]
Verified = #:lexc:#[wu.4 65 67 69 4 .3897991 -1.]
Sticking words ... (57,58,59,3) (65,67,69,4) #:node:g808 wu4 recognized
(51,10000,10000,()) #:node:g816 jia3 not found
(51,10000,10000,()) #:node:g824 xiao3 not found
(58,67,77,4) #:node:g832 huang2 recognized
(61,70,79,4) #:node:g840 hong2 recognized
(63,71,79,4) #:node:g847 he2 recognized
(51,10000,10000,()) #:node:g855 qian1 not found
(61,68,75,4) #:node:g863 dian4 recognized
(58,67,77,4) #:node:g871 fang1 recognized
(51,10000,10000,()) #:node:g879 qiu2 not found
(60,65,71,4) #:node:g886 zhuo1 recognized
Sticking words ... (57,58,59,3) (65,67,69,4) #:node:g895 wu4 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g392 (TASK) length = 3    0.328 accepted
#:ilot:g393 (TASK) length = 3    0.393 accepted
#:ilot:g391 (TASK) length = 3    0.352 accepted
#:ilot:g390 terminal rejected
#:ilot:g389 (TASK) length = 3    0.368 accepted
#:ilot:g388 (TASK) length = 3    0.338 accepted
#:ilot:g387 terminal rejected
#:ilot:g386 (TASK) length = 3    0.328 accepted
#:ilot:g385 (TASK) length = 3    0.372 accepted
#:ilot:g384 (TASK) length = 3    0.391 accepted
#:ilot:g383 terminal rejected
#:ilot:g382 terminal rejected
#:ilot:g381 (TASK) length = 3    0.328 accepted
#:ilot:g394 (TASK) length = 3    0.393 accepted
#:ilot:g380 (TASK) length = 3    0.352 accepted
#:ilot:g379 terminal rejected
#:ilot:g378 (TASK) length = 3    0.368 accepted
#:ilot:g377 (TASK) length = 3    0.338 accepted
#:ilot:g376 terminal rejected
```



```
#:ilot:g375 (TASK) length = 3    0.328 accepted
#:ilot:g374 (TASK) length = 3    0.372 accepted
#:ilot:g373 (TASK) length = 3    0.391 accepted
#:ilot:g372 terminal rejected
#:ilot:g371 terminal rejected
New list of ilots = [16]
Executing the specialist: "beam-search"
<1, 1, 0.424> <3, 8, 0.328>
New list of ilots = [18]
Executing the specialist: "predict-ph"
New list of ilots = [18]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (4) [77] = 101
#:ilot:g373 (R) #:pred:#[77 unsticked 89 82 101 109 (c ch sh q z zh r k p x j
f s g h t d b n l m -i ong uo ou ei e u)]
Next main pic (R) from (4) [71] = 101
diff = .1499052
--- Sticked (71,89): .02916664
#:ilot:g380 (R) #:pred:#[71 80 89 82 101 109 (ang ao eng iang uen c ch sh q z
zh r k p x j f s g h t d b n l m -i ong uo ou ei e u)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [22]
Executing the specialist: "hypoth-prepare"
New list of ilots = [22]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(77,10000,10000,()) #:node:g1077 he2 not found
#:phon:#[sh 85 86 91 .598558 -1.]
Verified = #:lexc:#[shu.1 85 92 99 5 .5133305 -1.]
(85,92,99,5) #:node:g1089 shu1 recognized
#:phon:#[uo 94 98 103 .4566975 -1.]
Verified = #:lexc:#[zhuo.1 86 94 103 5 .4809418 -1.]
(86,94,103,5) #:node:g1100 zhuo1 recognized
Verified = #:lexc:#[wu.4 93 96 99 5 .5014267 -1.]
(93,96,99,5) #:node:g1113 wu4 recognized
(79,10000,10000,()) #:node:g1029 he2 not found
(85,92,99,5) #:node:g1041 shu1 recognized
(86,94,103,5) #:node:g1052 zhuo1 recognized
(93,96,99,5) #:node:g1065 wu4 recognized
(79,10000,10000,()) #:node:g1013 z-10 not found
#:phon:#[-i 77 78 79 .3358336 -1.]
#:phon:#[e 75 77 79 .4571991 -1.]
(71,10000,10000,()) #:node:g1002 z-10 not found
(77,10000,10000,()) #:node:g971 he2 not found
(85,92,99,5) #:node:g979 shu1 recognized
(86,94,103,5) #:node:g986 zhuo1 recognized
(93,96,99,5) #:node:g995 wu4 recognized
(79,10000,10000,()) #:node:g939 he2 not found
(85,92,99,5) #:node:g947 shu1 recognized
(86,94,103,5) #:node:g954 zhuo1 recognized
(93,96,99,5) #:node:g963 wu4 recognized
(79,10000,10000,()) #:node:g930 z-10 not found
(71,10000,10000,()) #:node:g923 z-10 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g395 terminal rejected
#:ilot:g396 terminal rejected
#:ilot:g400 (TASK) length = 4    0.323 accepted
#:ilot:g399 (TASK) length = 4    0.369 accepted
#:ilot:g398 (TASK) length = 4    0.375 accepted
#:ilot:g397 terminal rejected
#:ilot:g404 (TASK) length = 4    0.328 accepted
#:ilot:g403 (TASK) length = 4    0.373 accepted
#:ilot:g402 (TASK) length = 4    0.380 accepted
#:ilot:g401 terminal rejected
#:ilot:g405 terminal rejected
#:ilot:g406 terminal rejected
#:ilot:g410 (TASK) length = 4    0.323 accepted
#:ilot:g409 (TASK) length = 4    0.369 accepted
#:ilot:g408 (TASK) length = 4    0.375 accepted
#:ilot:g407 terminal rejected
#:ilot:g414 (TASK) length = 4    0.328 accepted
#:ilot:g413 (TASK) length = 4    0.373 accepted
#:ilot:g412 (TASK) length = 4    0.380 accepted
#:ilot:g411 terminal rejected
New list of ilots = [14]
Executing the specialist: "beam-search"
<1, 1, 0.424> <4, 5, 0.323>
New list of ilots = [14]
Executing the specialist: "predict-ph"
```

```
New list of ilots = [14]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (5) [99] = 125
diff = .2029357
--- Sticked (99,112): .03784609
#:ilot:g412 (R) #:pred:#[99 105 112 106 125 133 (ou u uo b n g d c l f s h r
ch sh q z x zh j k t p uen ong -i uan eng iang ang er ao ing iou e ia =i)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [22]
Executing the specialist: "hypoth-prepare"
New list of ilots = [22]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ia 117 120 124 .5424986 -1.]
#:phon:#[x 109 112 116 .5291977 -1.]
Verified = #:lexc:#[xia.4 109 116 124 6 .5356477 -1.]
(109,116,124,6) #:node:g1239 xia4 recognized
#:phon:#[ang 99 101 103 .3471985 -1.]
#:phon:#[ang 120 125 131 .5586318 -1.]
#:phon:#[sh 110 113 117 .6137476 -1.]
Verified = #:lexc:#[shang.4 110 120 131 6 .523634 -1.]
(110,120,131,6) #:node:g1248 shang4 recognized
(109,116,124,6) #:node:g1259 xia4 recognized
(110,120,131,6) #:node:g1270 shang4 recognized
#:phon:#[iou 101 103 105 .3388987 -1.]
Verified = #:lexc:#[you.4 101 103 105 5 .3388982 -1.]
(101,103,105,5) #:node:g1281 you4 recognized
#:phon:#[uo 99 102 105 .4402836 -1.]
#:phon:#[uo 119 122 126 .4085236 -1.]
#:phon:#[z 105 108 111 .3995323 -1.]
Verified = #:lexc:#[zuo.3 105 116 125 6 .2697797 -1.]
(106,116,125,6) #:node:g1292 zuo3 recognized
#:phon:#[ou 99 102 105 .4725699 -1.]
#:phon:#[ou 119 122 126 .4189976 -1.]
(99,10000,10000,()) #:node:g1302 hou4 not found
#:phon:#[e 119 122 125 .4297118 -1.]
Verified = #:lexc:#[z-i.0 106 115 125 6 .2702584 -1.]
(106,115,125,6) #:node:g1225 z-i0 recognized
(109,116,124,6) #:node:g1156 xia4 recognized
(110,120,131,6) #:node:g1165 shang4 recognized
(109,116,124,6) #:node:g1175 xia4 recognized
(110,120,131,6) #:node:g1187 shang4 recognized
(101,103,105,5) #:node:g1198 you4 recognized
(106,116,125,6) #:node:g1209 zuo3 recognized
(99,10000,10000,()) #:node:g1219 hou4 not found
(106,115,125,6) #:node:g1142 z-i0 recognized
#:phon:#[d 108 108 109 .4039989 -1.]
(99,10000,10000,()) #:node:g1136 de0 not found
(106,115,125,6) #:node:g1130 z-i0 recognized
(99,10000,10000,()) #:node:g1125 de0 not found
(106,115,125,5) #:node:g1119 z-i0 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g415 (TASK) length = 5    0.343 accepted
#:ilot:g416 terminal rejected
#:ilot:g417 (TASK) length = 5    0.347 accepted
#:ilot:g418 terminal rejected
#:ilot:g419 (TASK) length = 5    0.343 accepted
#:ilot:g426 terminal rejected
#:ilot:g425 (TASK) length = 5    0.334 accepted
#:ilot:g424 (TASK) length = 5    0.369 accepted
#:ilot:g423 (TASK) length = 5    0.371 accepted
#:ilot:g422 (TASK) length = 5    0.368 accepted
#:ilot:g421 (TASK) length = 5    0.371 accepted
#:ilot:g420 (TASK) length = 5    0.368 accepted
#:ilot:g427 (TASK) length = 5    0.347 accepted
#:ilot:g434 terminal rejected
#:ilot:g433 (TASK) length = 5    0.337 accepted
#:ilot:g432 (TASK) length = 5    0.373 accepted
#:ilot:g431 (TASK) length = 5    0.374 accepted
#:ilot:g430 (TASK) length = 5    0.371 accepted
#:ilot:g429 (TASK) length = 5    0.374 accepted
#:ilot:g428 (TASK) length = 5    0.371 accepted
New list of ilots = [18]
Executing the specialist: "beam-search"
<1, 1, 0.424> <5, 10, 0.334>
New list of ilots = [18]
Executing the specialist: "predict-ph"
New list of ilots = [18]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
```



```
-------- Active association: lexicon
Next main pic (R) from (6) [124] = 145
#:ilot:g428 (R) #:pred:#[124 unsticked 134 126 145 153 (j r ch sh zh t g b d m
n k f s h l en iou ie =i ian ei i u% u uen ong iang -i ao uan e eng ang)]
Next main pic (R) from (6) [131] = 145
#:ilot:g429 (R) #:pred:#[131 unsticked 134 131 145 153 (j r ch sh zh t g b l f
s h d m n en -i e iou ie =i ian ei i u% u uen eng ang)]
Next main pic (R) from (5) [105] = 125
diff = .3779359
#:ilot:g432 (R) #:pred:#[105 unsticked 112 106 125 133 (n g d c l f s h r ch
sh q z x zh j k t p uen ong -i uan eng iang ang er ao ing iou e ia =i)]
Next main pic (R) from (6) [126] = 145
#:ilot:g433 (R) #:pred:#[126 unsticked 134 126 145 153 (j r ch sh zh t g b d m
n k f s h l en iou ie =i ian ei i u% u uen ong iang -i ao uan e eng ang)]
Next main pic (R) from (6) [125] = 145
#:ilot:g427 (R) #:pred:#[125 unsticked 134 126 145 153 (j r ch sh zh t g b d m
n k f s h l en iou ie =i ian ei i u% u uen ong iang -i ao uan e eng ang)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [28]
Executing the specialist: "hypoth-prepare"
New list of ilots = [28]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[e 144 146 149 .4556646 -1.]
#:phon:#[d 134 136 138 .4813986 -1.]
Verified = #:lexc:#[de.0 134 141 149 7 .321311 -1.]
(134,141,149,7) #:node:g1528 de0 recognized
#:phon:#[e 144 146 149 .4556646 -1.]
#:phon:#[d 134 136 138 .4813986 -1.]
Verified = #:lexc:#[de.0 134 141 149 7 .321311 -1.]
(134,141,149,7) #:node:g1521 de0 recognized
#:phon:#[ian 139 143 148 .5339975 -1.]
#:phon:#[m 132 134 137 .5416651 -1.]
Verified = #:lexc:#[mian.4 132 140 148 7 .505291 -1.]
(132,140,148,7) #:node:g1505 mian4 recognized
#:phon:#[b 135 136 138 .5154991 -1.]
Verified = #:lexc:#[bian.1 135 141 148 7 .5287113 -1.]
(135,141,148,7) #:node:g1514 bian1 recognized
#:phon:#[ian 139 143 148 .5339975 -1.]
#:phon:#[m 132 134 137 .5416651 -1.]
Verified = #:lexc:#[mian.4 132 140 148 7 .505291 -1.]
(132,140,148,7) #:node:g1487 mian4 recognized
#:phon:#[b 135 136 138 .5154991 -1.]
Verified = #:lexc:#[bian.1 135 141 148 7 .5287113 -1.]
(135,141,148,7) #:node:g1496 bian1 recognized
(132,140,148,7) #:node:g1469 mian4 recognized
(135,141,148,7) #:node:g1478 bian1 recognized
#:phon:#[ang 147 149 152 .4024987 -1.]
(125,10000,10000,()) #:node:g1428 shang4 not found
#:phon:#[i 139 140 141 .539999 -1.]
#:phon:#[l 134 136 138 .3836989 -1.]
Verified = #:lexc:#[li.3 134 137 141 7 .4423113 -1.]
Verified = #:lexc:#[li.3 139 140 141 7 .539999 -1.]
(139,140,141,7) #:node:g1438 li3 recognized
(125,10000,10000,()) #:node:g1449 shang4 not found
#:phon:#[i 139 140 141 .539999 -1.]
#:phon:#[iou 142 144 146 .4699988 -1.]
Verified = #:lexc:#[you.4 139 142 146 7 .4962482 -1.]
#:phon:#[iou 139 142 146 .3918743 -1.]
Verified = #:lexc:#[you.4 139 142 146 7 .3918743 -1.]
(139,142,146,7) #:node:g1460 you4 recognized
(134,141,149,7) #:node:g1419 de0 recognized
(134,141,149,7) #:node:g1412 de0 recognized
(132,140,148,7) #:node:g1396 mian4 recognized
(135,141,148,7) #:node:g1405 bian1 recognized
(132,140,148,7) #:node:g1378 mian4 recognized
(135,141,148,7) #:node:g1387 bian1 recognized
(132,140,148,7) #:node:g1360 mian4 recognized
(135,141,148,7) #:node:g1369 bian1 recognized
(125,10000,10000,()) #:node:g1319 shang4 not found
(139,140,141,7) #:node:g1329 li3 recognized
(125,10000,10000,()) #:node:g1340 shang4 not found
(139,142,146,7) #:node:g1351 you4 recognized
(134,141,149,7) #:node:g1310 de0 recognized
(134,141,149,7) #:node:g1306 de0 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g435 (TASK) length = 6    0.320 accepted
#:ilot:g436 (TASK) length = 6    0.323 accepted
#:ilot:g440 (TASK) length = 6    0.318 accepted
#:ilot:g439 terminal rejected
```

```
#:ilot:g438 (TASK) length = 6    0.312 accepted
#:ilot:g437 terminal rejected
#:ilot:g442 (TASK) length = 6    0.334 accepted
#:ilot:g441 (TASK) length = 6    0.343 accepted
#:ilot:g444 (TASK) length = 6    0.379 accepted
#:ilot:g443 (TASK) length = 6    0.388 accepted
#:ilot:g446 (TASK) length = 6    0.357 accepted
#:ilot:g445 (TASK) length = 6    0.366 accepted
#:ilot:g447 (TASK) length = 6    0.359 accepted
#:ilot:g448 (TASK) length = 6    0.337 accepted
#:ilot:g452 (TASK) length = 6    0.321 accepted
#:ilot:g451 terminal rejected
#:ilot:g450 (TASK) length = 6    0.315 accepted
#:ilot:g449 terminal rejected
#:ilot:g454 (TASK) length = 6    0.337 accepted
#:ilot:g453 (TASK) length = 6    0.346 accepted
#:ilot:g456 (TASK) length = 6    0.382 accepted
#:ilot:g455 (TASK) length = 6    0.391 accepted
#:ilot:g458 (TASK) length = 6    0.360 accepted
#:ilot:g457 (TASK) length = 6    0.369 accepted
#:ilot:g459 (TASK) length = 6    0.362 accepted
#:ilot:g460 (TASK) length = 6    0.340 accepted
New list of ilots = [24]
Executing the specialist: "beam-search"
#:ilot:g460 out of diameter
#:ilot:g453 out of diameter
#:ilot:g454 out of diameter
#:ilot:g450 out of diameter
#:ilot:g452 out of diameter
#:ilot:g448 out of diameter
#:ilot:g441 out of diameter
#:ilot:g442 out of diameter
#:ilot:g438 out of diameter
#:ilot:g440 out of diameter
#:ilot:g436 out of diameter
#:ilot:g435 out of diameter
<1, 1, 0.424> <6, 9, 0.359>
New list of ilots = [12]
Executing the specialist: "predict-ph"
New list of ilots = [12]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (7) [149] = 162
diff = .31885
#:ilot:g459 (R) #:pred:#[149 unsticked 154 149 162 170 (k p x j h t l g b d n
m ai -i e -i uan ian en eng)]
Next main pic (R) from (7) [148] = 162
#:ilot:g457 (R) #:pred:#[148 unsticked 154 148 162 170 (k p x j h t l g b d m
n ai -i e uen en ian -i eng)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [16]
Executing the specialist: "hypoth-prepare"
New list of ilots = [16]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ai 156 159 163 .365624 -1.]
#:phon:#[b 153 154 156 .5332489 -1.]
Verified = #:lexc:#[bai.2 153 156 163 8 .4113393 -1.]
(153,156,163,8) #:node:g1616 bai2 recognized
#:phon:#[e 156 159 163 .6162481 -1.]
#:phon:#[h 152 154 156 .3587985 -1.]
Verified = #:lexc:#[he.2 152 157 163 8 .5304308 -1.]
(152,157,163,8) #:node:g1623 he2 recognized
#:phon:#[ian 159 161 154 .3236651 -1.]
#:phon:#[d 152 154 157 .5141649 -1.]
Verified = #:lexc:#[dian.4 152 158 164 8 .3866901 -1.]
(152,158,164,8) #:node:g1631 dian4 recognized
#:phon:#[d 152 154 157 .5141649 -1.]
Verified = #:lexc:#[de.0 152 157 163 8 .5822191 -1.]
(152,157,163,8) #:node:g1608 de0 recognized
(152,157,163,8) #:node:g1601 de0 recognized
(152,157,163,8) #:node:g1594 de0 recognized
(152,157,163,8) #:node:g1587 de0 recognized
(153,158,163,8) #:node:g1564 bai2 recognized
(152,157,163,8) #:node:g1571 he2 recognized
(152,158,164,8) #:node:g1579 dian4 recognized
(152,157,163,8) #:node:g1556 de0 recognized
(152,157,163,8) #:node:g1549 de0 recognized
(152,157,163,8) #:node:g1542 de0 recognized
(152,157,163,8) #:node:g1535 de0 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
```



```
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g461 (TASK) length = 7    0.388 accepted
#:ilot:g462 (TASK) length = 7    0.396 accepted
#:ilot:g463 (TASK) length = 7    0.368 accepted
#:ilot:g464 (TASK) length = 7    0.376 accepted
#:ilot:g467 (TASK) length = 7    0.357 accepted
#:ilot:g466 (TASK) length = 7    0.368 accepted
#:ilot:g465 (TASK) length = 7    0.356 accepted
#:ilot:g468 (TASK) length = 7    0.391 accepted
#:ilot:g469 (TASK) length = 7    0.399 accepted
#:ilot:g470 (TASK) length = 7    0.371 accepted
#:ilot:g471 (TASK) length = 7    0.379 accepted
#:ilot:g474 (TASK) length = 7    0.359 accepted
#:ilot:g473 (TASK) length = 7    0.371 accepted
#:ilot:g472 (TASK) length = 7    0.358 accepted
New list of ilots = [16]
Executing the specialist: "beam-search"
#:ilot:g472 out of diameter
#:ilot:g474 out of diameter
#:ilot:g465 out of diameter
#:ilot:g467 out of diameter
<1, 1, 0.424> <7, 9, 0.358>
New list of ilots = [12]
Executing the specialist: "predict-ph"
New list of ilots = [12]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (8) [163] = 185
#:ilot:g473 (R) #:pred:#[163 unsticked 172 166 185 193 (ch sh q p x j c zh z k
l s f n g h b t d eng an uan ai ian i -i u% ei uen en)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [74]
Executing the specialist: "hypoth-prepare"
New list of ilots = [74]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[i 179 180 181 .4899998 -1.]
Verified = #:lexc:#[yi.3 179 180 181 9 .4899998 -1.]
(179,180,181,9) #:node:g2136 yi3 recognized
#:phon:#[ai 182 184 187 .3709989 -1.]
#:phon:#[b 172 173 174 .4399986 -1.]
(163,10000,10000,()) #:node:g2144 bai2 not found
#:phon:#[uan 182 184 187 .3283319 -1.]
#:phon:#[d 167 169 172 .4676652 -1.]
(163,10000,10000,()) #:node:g2152 duan3 not found
(163,10000,10000,()) #:node:g2160 hei1 not found
#:phon:#[an 183 185 189 .3374271 -1.]
(163,10000,10000,()) #:node:g2168 lan2 not found
#:phon:#[u% 179 180 181 .496666 -1.]
(163,10000,10000,()) #:node:g2176 lu%4 not found
#:phon:#[ian 181 184 188 .4217863 -1.]
#:phon:#[q 171 173 175 .5498361 -1.]
Verified = #:lexc:#[qian.1 171 179 188 9 .3421928 -1.]
(171,179,188,9) #:node:g2183 qian1 recognized
#:phon:#[d 167 169 172 .4676652 -1.]
Verified = #:lexc:#[dian.4 167 177 188 9 .2809215 -1.]
(167,177,188,9) #:node:g2191 dian4 recognized
#:phon:#[s 171 173 175 .5135973 -1.]
Verified = #:lexc:#[san.1 171 180 189 9 .2871346 -1.]
(171,180,189,9) #:node:g2199 san1 recognized
(179,180,181,9) #:node:g2065 yi3 recognized
(163,10000,10000,()) #:node:g2073 bai2 not found
(163,10000,10000,()) #:node:g2081 duan3 not found
(163,10000,10000,()) #:node:g2089 hei1 not found
(163,10000,10000,()) #:node:g2097 lan2 not found
(163,10000,10000,()) #:node:g2105 lu%4 not found
(171,179,188,9) #:node:g2112 qian1 recognized
(167,177,188,9) #:node:g2120 dian4 recognized
(171,180,189,9) #:node:g2128 san1 recognized
(179,180,181,9) #:node:g1994 yi3 recognized
(163,10000,10000,()) #:node:g2002 bai2 not found
(163,10000,10000,()) #:node:g2010 duan3 not found
(163,10000,10000,()) #:node:g2018 hei1 not found
(163,10000,10000,()) #:node:g2026 lan2 not found
(163,10000,10000,()) #:node:g2034 lu%4 not found
(171,179,188,9) #:node:g2041 qian1 recognized
(167,177,188,9) #:node:g2049 dian4 recognized
(171,180,189,9) #:node:g2057 san1 recognized
(179,180,181,9) #:node:g1923 yi3 recognized
(163,10000,10000,()) #:node:g1931 bai2 not found
(163,10000,10000,()) #:node:g1939 duan3 not found
(163,10000,10000,()) #:node:g1947 hei1 not found
(163,10000,10000,()) #:node:g1955 lan2 not found
(163,10000,10000,()) #:node:g1963 lu%4 not found
```

```
(171,179,188,9) #:node:g1970 qian1 recognized
(167,177,188,9) #:node:g1976 dian4 recognized
(171,180,189,9) #:node:g1986 san1 recognized
(179,180,181,9) #:node:g1852 yi3 recognized
(163,10000,10000,()) #:node:g1860 bai2 not found
(163,10000,10000,()) #:node:g1868 duan3 not found
(163,10000,10000,()) #:node:g1876 hei1 not found
(163,10000,10000,()) #:node:g1884 lan2 not found
(163,10000,10000,()) #:node:g1892 lu%4 not found
(171,179,188,9) #:node:g1899 qian1 recognized
(167,177,188,9) #:node:g1907 dian4 recognized
(171,180,189,9) #:node:g1915 san1 recognized
(179,180,181,9) #:node:g1781 yi3 recognized
(163,10000,10000,()) #:node:g1789 bai2 not found
(163,10000,10000,()) #:node:g1797 duan3 not found
(163,10000,10000,()) #:node:g1805 hei1 not found
(163,10000,10000,()) #:node:g1813 lan2 not found
(163,10000,10000,()) #:node:g1821 lu%4 not found
(171,179,188,9) #:node:g1828 qian1 recognized
(167,177,188,9) #:node:g1836 dian4 recognized
(171,180,189,9) #:node:g1844 san1 recognized
(179,180,181,9) #:node:g1710 yi3 recognized
(163,10000,10000,()) #:node:g1718 bai2 not found
(163,10000,10000,()) #:node:g1726 duan3 not found
(163,10000,10000,()) #:node:g1734 hei1 not found
(163,10000,10000,()) #:node:g1742 lan2 not found
(163,10000,10000,()) #:node:g1750 lu%4 not found
(171,179,188,9) #:node:g1757 qian1 recognized
(167,177,188,9) #:node:g1765 dian4 recognized
(171,180,189,9) #:node:g1773 san1 recognized
(179,180,181,9) #:node:g1639 yi3 recognized
(163,10000,10000,()) #:node:g1647 bai2 not found
(163,10000,10000,()) #:node:g1655 duan3 not found
(163,10000,10000,()) #:node:g1663 hei1 not found
(163,10000,10000,()) #:node:g1671 lan2 not found
(163,10000,10000,()) #:node:g1679 lu%4 not found
(171,179,188,9) #:node:g1686 qian1 recognized
(167,177,188,9) #:node:g1694 dian4 recognized
(171,180,189,9) #:node:g1702 san1 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g483 (TASK) length = 8    0.361 accepted
#:ilot:g482 (TASK) length = 8    0.367 accepted
#:ilot:g481 (TASK) length = 8    0.367 accepted
#:ilot:g480 terminal rejected
#:ilot:g479 terminal rejected
#:ilot:g478 terminal rejected
#:ilot:g477 terminal rejected
#:ilot:g476 terminal rejected
#:ilot:g475 (TASK) length = 8    0.354 accepted
#:ilot:g492 (TASK) length = 8    0.368 accepted
#:ilot:g491 (TASK) length = 8    0.374 accepted
#:ilot:g490 (TASK) length = 8    0.374 accepted
#:ilot:g489 terminal rejected
#:ilot:g488 terminal rejected
#:ilot:g487 terminal rejected
#:ilot:g486 terminal rejected
#:ilot:g485 terminal rejected
#:ilot:g484 (TASK) length = 8    0.362 accepted
#:ilot:g501 (TASK) length = 8    0.344 accepted
#:ilot:g500 (TASK) length = 8    0.350 accepted
#:ilot:g499 (TASK) length = 8    0.350 accepted
#:ilot:g498 terminal rejected
#:ilot:g497 terminal rejected
#:ilot:g496 terminal rejected
#:ilot:g495 terminal rejected
#:ilot:g494 terminal rejected
#:ilot:g493 (TASK) length = 8    0.336 accepted
#:ilot:g510 (TASK) length = 8    0.351 accepted
#:ilot:g509 (TASK) length = 8    0.357 accepted
#:ilot:g508 (TASK) length = 8    0.357 accepted
#:ilot:g507 terminal rejected
#:ilot:g506 terminal rejected
#:ilot:g505 terminal rejected
#:ilot:g504 terminal rejected
#:ilot:g503 terminal rejected
#:ilot:g502 (TASK) length = 8    0.344 accepted
#:ilot:g519 (TASK) length = 8    0.363 accepted
#:ilot:g518 (TASK) length = 8    0.370 accepted
#:ilot:g517 (TASK) length = 8    0.369 accepted
#:ilot:g516 terminal rejected
#:ilot:g515 terminal rejected
#:ilot:g514 terminal rejected
```



```
#:ilot:g513 terminal rejected
#:ilot:g512 terminal rejected
#:ilot:g511 (TASK) length = 8    0.357 accepted
#:ilot:g528 (TASK) length = 8    0.370 accepted
#:ilot:g527 (TASK) length = 8    0.377 accepted
#:ilot:g526 (TASK) length = 8    0.375 accepted
#:ilot:g525 terminal rejected
#:ilot:g524 terminal rejected
#:ilot:g523 terminal rejected
#:ilot:g522 terminal rejected
#:ilot:g521 terminal rejected
#:ilot:g520 (TASK) length = 8    0.364 accepted
#:ilot:g537 (TASK) length = 8    0.346 accepted
#:ilot:g536 (TASK) length = 8    0.352 accepted
#:ilot:g535 (TASK) length = 8    0.352 accepted
#:ilot:g534 terminal rejected
#:ilot:g533 terminal rejected
#:ilot:g532 terminal rejected
#:ilot:g531 terminal rejected
#:ilot:g530 terminal rejected
#:ilot:g529 (TASK) length = 8    0.339 accepted
#:ilot:g546 (TASK) length = 8    0.353 accepted
#:ilot:g545 (TASK) length = 8    0.359 accepted
#:ilot:g544 (TASK) length = 8    0.359 accepted
#:ilot:g543 terminal rejected
#:ilot:g542 terminal rejected
#:ilot:g541 terminal rejected
#:ilot:g540 terminal rejected
#:ilot:g539 terminal rejected
#:ilot:g538 (TASK) length = 8    0.346 accepted
New list of ilots = [34]
Executing the specialist: "beam-ssarch"
#:ilot:g538 out of diameter
#:ilot:g544 out of diameter
#:ilot:g545 out of diameter
#:ilot:g546 out of diameter
#:ilot:g529 out of diameter
#:ilot:g535 out of diameter
#:ilot:g536 out of diameter
#:ilot:g537 out of diameter
#:ilot:g511 out of diameter
#:ilot:g502 out of diameter
#:ilot:g508 out of diameter
#:ilot:g509 out of diameter
#:ilot:g510 out of diameter
#:ilot:g493 out of diameter
#:ilot:g499 out of diameter
#:ilot:g500 out of diameter
#:ilot:g501 out of diameter
#:ilot:g484 out of diameter
#:ilot:g475 out of diameter
#:ilot:g483 out of diameter
<1, 1, 0.424> <8, 8, 0.368>
New list of ilots = [14]
Executing the specialist: "predict-ph"
New list of ilots = [14]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (9) [181) = 203
diff = .07187176
--- Sticked (181,194): .82653767
#:ilot:g520 (R) #:pred:#[181 187 194 184 203 211 (k p zh r g h t b d n m l e
ie ing u% i =i eng an uan uen en ai ian)]
Next main pic (R) from (9) [188) = 203
#:ilot:g525 (R) #:pred:#[188 unsticked 194 188 203 211 (k p zh r g h t b d n m
l ai e ie ing u% i =i uen eng en)]
Next main pic (R) from (9) [189) = 203
#:ilot:g528 (R) #:pred:#[189 unsticked 194 189 203 211 (k p zh r g h t b d l n
m ai e ie ing u% i =i uen en eng)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [7]
Executing the specialist: "hypoth-prepare"
New list of ilots = [7]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(181,10000,10000,()) #:node:g2232 de0 not found
#:phon:#[i 196 200 204 .5344424 -1.]
#:phon:#[b 192 193 195 .5119582 -1.]
Verified = #:lexc:#[bi3 192 198 204 10 .5413618 -1.]
(192,198,204,10) #:node:g2226 bi3 recognized
(192,198,204,10) #:node:g2219 bi3 recognized
(192,198,204,10) #:node:g2212 bi3 recognized
(192,198,204,10) #:node:g2205 bi3 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
```

```
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g547 (TASK) length = 9    0.373 accepted
#:ilot:g548 (TASK) length = 9    0.360 accepted
#:ilot:g549 (TASK) length = 9    0.375 accepted
#:ilot:g550 (TASK) length = 9    0.382 accepted
#:ilot:g551 terminal rejected
New list of ilots = [6]
Executing the specialist: "beam-search"
<1, 1, 0.424> <9, 4, 0.373>
New list of ilots = [6]
Executing the specialist: "predict-ph"
New list of ilots = [6]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (10) [204] = 215
#:ilot:g550 (R) #:pred:#[204 unsticked 223 204 215 223 (n p l g d t b q x u an
eng -i ai u% e =i ie i ing)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [6]
Executing the specialist: "hypoth-prepare"
New list of ilots = [6]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[e 210 213 217 .5394953 -1.]
#:phon:#[d 207 208 210 .4377489 -1.]
Verified = #:lexc:#[de.0 207 212 217 11 .5117474 -1.]
(207,212,217,11) #:node:g2248 de0 recognized
(207,212,217,11) #:node:g2244 de0 recognized
(207,212,217,11) #:node:g2240 de0 recognized
(207,212,217,11) #:node:g2236 de0 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g552 (TASK) length = 10    0.377 accepted
#:ilot:g553 (TASK) length = 10    0.383 accepted
#:ilot:g554 (TASK) length = 10    0.379 accepted
#:ilot:g555 (TASK) length = 10    0.385 accepted
New list of ilots = [6]
Executing the specialist: "beam-search"
<1, 1, 0.424> <10, 4, 0.377>
New list of ilots = [6]
Executing the specialist: "predict-ph"
New list of ilots = [6]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (11) [217] = 233
#:ilot:g555 (R) #:pred:#[217 unsticked 223 217 233 241 (p r ch sh c z x zh q j
s f h g d t b m l n uei u% eng en uen ou ao iou uo u =i an -i e)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [14]
Executing the specialist: "hypoth-prepare"
New list of ilots = [14]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(217,10000,10000,()) #:node:g2287 wei4 not found
#:phon:#[ao 228 231 234 .5053835 -1.]
#:phon:#[g 221 223 225 .482399 -1.]
Verified = #:lexc:#[gao.1 221 227 234 12 .4249768 -1.]
(221,227,234,12) #:node:g2291 gao1 recognized
#:phon:#[uo 228 231 235 .5862484 -1.]
#:phon:#[z 221 223 226 .4824314 -1.]
Verified = #:lexc:#[zuo.4 221 228 235 12 .5056381 -1.]
(221,228,235,12) #:node:g2295 zuo4 recognized
(217,10000,10000,()) #:node:g2275 wei4 not found
(221,227,234,12) #:node:g2279 gao1 recognized
(221,228,235,12) #:node:g2283 zuo4 recognized
(217,10000,10000,()) #:node:g2263 wei4 not found
(221,227,234,12) #:node:g2267 gao1 recognized
(221,228,235,12) #:node:g2271 zuo4 recognized
(217,10000,10000,()) #:node:g2251 wei4 not found
(221,227,234,12) #:node:g2255 gao1 recognized
(221,228,235,12) #:node:g2259 zuo4 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
```



```
#:ilot:g558 (TASK) length = 11    0.381 accepted
#:ilot:g557 (TASK) length = 11    0.375 accepted
#:ilot:g556 terminal rejected
#:ilot:g561 (TASK) length = 11    0.366 accepted
#:ilot:g560 (TASK) length = 11    0.361 accepted
#:ilot:g559 terminal rejected
#:ilot:g564 (TASK) length = 11    0.383 accepted
#:ilot:g563 (TASK) length = 11    0.377 accepted
#:ilot:g562 terminal rejected
#:ilot:g567 (TASK) length = 11    0.388 accepted
#:ilot:g566 (TASK) length = 11    0.382 accepted
#:ilot:g565 terminal rejected
New list of ilots = [10]
Executing the specialist: "beam-search"
<1, 1, 0.424> <11, 7, 0.377>
New list of ilots = [10]
Executing the specialist: "predict-ph"
New list of ilots = [10]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (12) [234] = 251
#:ilot:g566 (R) #:pred:#[234 unsticked 240 234 251 259 (p t g l d b n m f s h
ang a iang ia iou ao ian ai =i i uei u% eng en uen u ou uo -i e)]
Next main pic (R) from (12) [235] = 251
#:ilot:g567 (R) #:pred:#[235 unsticked 267 235 251 259 (p t g l d b n m f s h
ang a iang ia iou ao ian ai =i i uei u% eng en uen u ou uo -i e)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [10]
Executing the specialist: "hypoth-prepare"
New list of ilots = [10]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[u 235 237 239 .4239988 -1.]
#:phon:#[d 237 237 237 .4049993 -1.]
Verified = #:lexc:#[du.4 237 238 239 13 .4366646 -1.]
(237,238,239,13) #:node:g2320 du4 recognized
#:phon:#[ao 244 247 251 .5274982 -1.]
#:phon:#[b 238 239 241 .5249987 -1.]
Verified = #:lexc:#[biao.1 238 244 251 13 .4514265 -1.]
#:phon:#[i 242 243 245 .5099993 -1.]
#:phon:#[ao 248 251 255 .5374985 -1.]
#:phon:#[b 238 239 241 .5249987 -1.]
Verified = #:lexc:#[biao.1 238 246 255 13 .4688873 -1.]
(238,246,255,13) #:node:g2317 biao1 recognized
(237,238,239,13) #:node:g2314 du4 recognized
(238,246,255,13) #:node:g2311 biao1 recognized
(237,238,239,13) #:node:g2308 du4 recognized
(238,246,255,13) #:node:g2305 biao1 recognized
(237,238,239,13) #:node:g2302 du4 recognized
(238,246,255,13) #:node:g2299 biao1 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g352 (R-OBJ1) length = 1    0.424 accepted
#:ilot:g354 (OBJECT) length = 1    0.424 accepted
#:ilot:g568 (TASK) length = 12    0.364 accepted
#:ilot:g569 (TASK) length = 12    0.373 accepted
#:ilot:g570 (TASK) length = 12    0.389 accepted
#:ilot:g571 (TASK) length = 12    0.378 accepted
#:ilot:g572 (TASK) length = 12    0.386 accepted
#:ilot:g573 (TASK) length = 12    0.374 accepted
#:ilot:g574 (TASK) length = 12    0.391 accepted
#:ilot:g575 (TASK) length = 12    0.380 accepted
New list of ilots = [10]
Executing the specialist: "beam-search"
<1, 1, 0.424> <12, 7, 0.374>
New list of ilots = [10]
Executing the specialist: "predict-ph"
New list of ilots = [10]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (13) [239] = 279
diff = .05386734
--- Sticked (239,271): .01003124
#:ilot:g575 (R) #:pred:#[239 255 271 268 279 287 (u% uei i =i ai ian ao iou ia
iang a e ang ou n r m l j q x ch sh z zh c ss d p b f s h g t k ss u)]
Next main pic (R) from (13) [255] = 279
#:ilot:g574 (R) #:pred:#[255 unsticked 278 268 279 287 (r m l j q x ch sh z zh
c ss d p b f s h g t k ss u)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [10]
Executing the specialist: "position-constrain"
```

```
New list of ilots = [10]
Executing the specialist: "forward-chain-main"
New list of ilots = [15]
Executing the specialist: "backward-chain-left"
New list of ilots = [11]
Executing the specialist: "position-constrain"
New list of ilots = [11]
Executing the specialist: "backward-chain-right"
New list of ilots = [15]
Executing the specialist: "hypoth-prepare"
New list of ilots = [15]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(51,10000,10000,()) #:node:g2367 xia4 not found
(51,10000,10000,()) #:node:g2375 xia4 not found
Verified = #:lexc:#(you.4 52 55 59 3 .434999 -1.)
(52,55,59,3) #:node:g2383 you4 recognized
#:phon:#[z 60 61 63 .4749994 -1.]
Verified = #:lexc:#[zuo.3 60 65 71 4 .3551149 -1.]
(60,65,71,4) #:node:g2391 zuo3 recognized
Verified = #:lexc:#[hou.4 58 64 71 4 .4047832 -1.]
(58,64,71,4) #:node:g2398 hou4 recognized
(51,10000,10000,()) #:node:g2405 qian2 not found
#:phon:#[d 51 52 53 .3119993 -1.]
Verified = #:lexc:#[de.0 51 54 57 3 .532424 -1.]
#:phon:#[d 61 63 66 .3759985 -1.]
(51,54,57,3) #:node:g2361 de0 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g583 (PHRASE) length = 12    0.384 accepted
#:ilot:g584 (PHRASE) length = 12    0.373 accepted
#:ilot:g585 (PHRASE) length = 12    0.389 accepted
#:ilot:g586 (PHRASE) length = 12    0.378 accepted
#:ilot:g587 (PHRASE) length = 12    0.386 accepted
#:ilot:g588 (PHRASE) length = 12    0.374 accepted
#:ilot:g589 (PHRASE) length = 12    0.391 accepted
#:ilot:g590 (PHRASE) length = 12    0.380 accepted
#:ilot:g591 (QUEST) length = 2    0.532 accepted
#:ilot:g597 terminal rejected
#:ilot:g596 (OBJECT) length = 2    0.311 accepted
#:ilot:g595 (OBJECT) length = 2    0.255 accepted
#:ilot:g594 (OBJECT) length = 2    0.431 accepted
#:ilot:g593 terminal rejected
#:ilot:g592 terminal rejected
New list of ilots = [12]
Executing the specialist: "beam-search"
<2, 4, 0.255> <12, 7, 0.374>
New list of ilots = [12]
Executing the specialist: "predict-ph"
New list of ilots = [12]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [10]
Executing the specialist: "hypoth-prepare"
New list of ilots = [10]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[g 63 65 67 .4211979 -1.]
Verified = #:lexc:#[gao.1 63 70 78 4 .3245363 -1.]
(63,70,78,4) #:node:g2409 gao1 recognized
#:phon:#[uo 65 68 71 .3373413 -1.]
Verified = #:lexc:#[zuo.4 60 65 71 4 .3551149 -1.]
(60,65,71,4) #:node:g2414 zuo4 recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g590 (PHRASE) length = 12    0.380 accepted
#:ilot:g589 (PHRASE) length = 12    0.391 accepted
#:ilot:g588 (PHRASE) length = 12    0.374 accepted
#:ilot:g587 (PHRASE) length = 12    0.386 accepted
#:ilot:g586 (PHRASE) length = 12    0.378 accepted
#:ilot:g585 (PHRASE) length = 12    0.389 accepted
#:ilot:g584 (PHRASE) length = 12    0.373 accepted
#:ilot:g583 (PHRASE) length = 12    0.384 accepted
#:ilot:g599 (QUEST) length = 3    0.424 accepted
#:ilot:g598 (QUEST) length = 3    0.345 accepted
New list of ilots = [10]
Executing the specialist: "beam-search"
<3, 2, 0.345> <12, 7, 0.374>
New list of ilots = [10]
Executing the specialist: "predict-ph"
New list of ilots = [10]
```



```
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [10]
Executing the specialist: "hypoth-prepare"
New list of ilots = [10]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[d 87 89 92 .4723306 -1.]
Verified = #:lexc:#[du.4 87 93 99 5 .4879975 -1.]
(87,93,99,5) #:node:g2422 du4 recognized
#:phon:#[ao 72 75 79 .3891244 -1.]
#:phon:#[ao 95 98 102 .351711 -1.]
#:phon:#[b 82 82 83 .5109987 -1.]
(71,10000,10000,()) #:node:g2418 biao1 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g583 (PHRASE) length = 12    0.384 accepted
#:ilot:g584 (PHRASE) length = 12    0.373 accepted
#:ilot:g585 (PHRASE) length = 12    0.389 accepted
#:ilot:g586 (PHRASE) length = 12    0.378 accepted
#:ilot:g587 (PHRASE) length = 12    0.386 accepted
#:ilot:g588 (PHRASE) length = 12    0.374 accepted
#:ilot:g589 (PHRASE) length = 12    0.391 accepted
#:ilot:g590 (PHRASE) length = 12    0.380 accepted
#:ilot:g600 terminal rejected
#:ilot:g601 (QUEST) length = 4    0.328 accepted
New list of ilots = [9]
Executing the specialist: "beam-search"
<4, 1, 0.328> <12, 7, 0.374>
New list of ilots = [9]
Executing the specialist: "predict-ph"
New list of ilots = [9]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [9]
Executing the specialist: "hypoth-prepare"
New list of ilots = [9]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
(99,10000,10000,()) #:node:g2425 she4 not found
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g590 (PHRASE) length = 12    0.380 accepted
#:ilot:g589 (PHRASE) length = 12    0.391 accepted
#:ilot:g588 (PHRASE) length = 12    0.374 accepted
#:ilot:g587 (PHRASE) length = 12    0.386 accepted
#:ilot:g586 (PHRASE) length = 12    0.378 accepted
#:ilot:g585 (PHRASE) length = 12    0.389 accepted
#:ilot:g584 (PHRASE) length = 12    0.373 accepted
#:ilot:g583 (PHRASE) length = 12    0.384 accepted
#:ilot:g602 terminal rejected
New list of ilots = [8]
Executing the specialist: "beam-search"
<12, 7, 0.374>
New list of ilots = [8]
Executing the specialist: "predict-ph"
New list of ilots = [8]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [8]
Executing the specialist: "position-constrain"
New list of ilots = [8]
Executing the specialist: "forward-chain-main"
New list of ilots = [8]
Executing the specialist: "backward-chain-left"
New list of ilots = [8]
Executing the specialist: "hypoth-prepare"
New list of ilots = [8]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#[ss -9 -4 0 .6909971 -1.]
Verified = #:lexc:#[ss -9 -4 0 1 .6909962 -1.]
(-9,-4,0,1) #:node:g2484 ss recognized
(-9,-4,0,1) #:node:g2480 ss recognized
(-9,-4,0,1) #:node:g2476 ss recognized
```

```
(-9,-4,0,1) #:node:g2472 ss recognized
(-9,-4,0,1) #:node:g2468 ss recognized
(-9,-4,0,1) #:node:g2464 ss recognized
(-9,-4,0,1) #:node:g2460 ss recognized
(-9,-4,0,1) #:node:g2456 ss recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g611 (SENTENCE) length = 13    0.370 accepted
#:ilot:g612 (SENTENCE) length = 13    0.358 accepted
#:ilot:g613 (SENTENCE) length = 13    0.374 accepted
#:ilot:g614 (SENTENCE) length = 13    0.363 accepted
#:ilot:g615 (SENTENCE) length = 13    0.371 accepted
#:ilot:g616 (SENTENCE) length = 13    0.360 accepted
#:ilot:g617 (SENTENCE) length = 13    0.376 accepted
#:ilot:g618 (SENTENCE) length = 13    0.365 accepted
New list of ilots = [8]
Executing the specialist: "beam-search"
<13, 7, 0.360>
New list of ilots = [8]
Executing the specialist: "predict-ph"
New list of ilots = [8]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (L) from (1) [-9] = 0
#:ilot:g618 (L) #:pred:#(0 unsticked -9 -19 0 -9 (g d t b h s f ss sh))
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-left"
New list of ilots = [8]
Executing the specialist: "position-constrain"
New list of ilots = [8]
Executing the specialist: "backward-chain-right"
New list of ilots = [8]
Executing the specialist: "hypoth-prepare"
New list of ilots = [8]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-word
-------- Active association: lexicon
#:phon:#(ss 270 274 279 .7269974 -1.)
Verified = #:lexe:#(ss 270 274 279 14 .7269974 -1.)
(270,274,279,14) #:node:g2506 ss recognized
(270,274,279,14) #:node:g2505 ss recognized
(270,274,279,14) #:node:g2502 ss recognized
(270,274,279,14) #:node:g2499 ss recognized
(270,274,279,14) #:node:g2496 ss recognized
(270,274,279,14) #:node:g2493 ss recognized
(270,274,279,14) #:node:g2490 ss recognized
(270,274,279,14) #:node:g2487 ss recognized
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "qualify"
#:ilot:g619 (SENTENCE) length = 14    0.364 accepted
#:ilot:g620 (SENTENCE) length = 14    0.334 accepted
#:ilot:g621 (SENTENCE) length = 14    0.368 accepted
#:ilot:g622 (SENTENCE) length = 14    0.338 accepted
#:ilot:g623 (SENTENCE) length = 14    0.366 accepted
#:ilot:g624 (SENTENCE) length = 14    0.335 accepted
#:ilot:g625 (SENTENCE) length = 14    0.370 accepted
#:ilot:g626 (SENTENCE) length = 14    0.339 accepted
New list of ilots = [8]
Executing the specialist: "beam-search"
---- Complete sentence:
>> ilot = #:ilot:g626 Len = 14 Quality =    0.339
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
---- Complete sentence:
>> ilot = #:ilot:g625 Len = 14 Quality =    0.370
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
---- Complete sentence:
>> ilot = #:ilot:g624 Len = 14 Quality =    0.335
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
---- Complete sentence:
>> ilot = #:ilot:g623 Len = 14 Quality =    0.366
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
---- Complete sentence:
>> ilot = #:ilot:g622 Len = 14 Quality =    0.338
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
---- Complete sentence:
>> ilot = #:ilot:g621 Len = 14 Quality =    0.368
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
---- Complete sentence:
>> ilot = #:ilot:g620 Len = 14 Quality =    0.334
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
---- Complete sentence:
>> ilot = #:ilot:g619 Len = 14 Quality =    0.364
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
<14, 7, 0.335>
```



```
New list of ilots = [8]
Executing the specialist: "predict-ph"
New list of ilots = [8]
Message from "syntactic-semantics" to "lexicon" Intention: hypothese-ph
-------- Active association: lexicon
Next main pic (R) from (14) [279] = 279
#:ilot:g626 (R) #:pred:#[279 unsticked 279 279 279 287 (g k f s h z j r ch sh
zh m l ss ss)]
Message from "lexicon" to "syntactic-semantics" Intention: proposition
-------- Active association: syntactic-semantics
Executing the specialist: "backward-chain-right"
New list of ilots = [8]
Executing the specialist: "position-constrain"
New list of ilots = [8]
Executing the specialist: "forward-chain-main"
New list of ilots = [8]
End of expert society execution
Time used = 222.7 seconds.
527 ilot, 2409 node, 394 lexc, 45 pred, 176 phon, 95 letter, .
>> ilot = #:ilot:g625 Len = 14 Quality =    0.370
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
>> ilot = #:ilot:g621 Len = 14 Quality =    0.368
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
>> ilot = #:ilot:g623 Len = 14 Quality =    0.366
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
>> ilot = #:ilot:g619 Len = 14 Quality =    0.364
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 zuo.4 biao.1 ss
>> ilot = #:ilot:g626 Len = 14 Quality =    0.339
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
>> ilot = #:ilot:g622 Len = 14 Quality =    0.338
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 mian.4 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
>> ilot = #:ilot:g624 Len = 14 Quality =    0.335
ss kong.4 zh=i.4 huang.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
>> ilot = #:ilot:g620 Len = 14 Quality =    0.334
ss kong.4 zh=i.4 hong.2 shu.1 shang.4 bian.1 de.0 qian.1 bi.3 de.0 gao.1 du.4 ss
= t

\end{verbatim}
} % scriptsize
```

NOM DE L'ETUDIANT : *GONG Yifan*

NATURE DE LA THESE : *Doctorat de l'Université de NANCY I en Informatique*

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le - 3 MAI 1988

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

R. MAINARD

RÉSUMÉ

L'interprétation du signal est un processus de transformations successives d'information vers une forme compréhensible par l'homme ou par la machine. A cause de l'incertitude du signal dûe à la production ou à la transmission du signal, le processus d'interprétation exige l'utilisation des traitements d'intelligence artificielle de différents niveaux d'abstraction.

Dans l'objectif de l'interprétation de la parole continue, nous avons proposé et mis en œuvre un ensemble de travaux sur les quatre aspects importants de l'interprétation: l'édition et modélisation du signal, la conversion du signal en symboles, l'analyse de la structure du signal et l'architecture du système d'interprétation.

Dans la partie I nous présentons d'abord un outil d'édition et de manipulation des signaux qui permet l'extraction et l'évaluation de paramètres et qui aide à l'acquisition des connaissances d'observation sur les signaux. Cet outil fournit également un environnement de simulation des nouveaux algorithmes de traitement à l'aide des opérateurs existants (chapitre 2). Ensuite nous étudions un modèle du signal non-stationnaire destiné à l'estimation de la fréquence fondamentale de la parole, où le signal est modélisé par une séquence de fonctions spécifiées de façon à autoriser la variabilité en fonction du temps de la période et de l'amplitude de l'excitation du signal. Les coefficients du modèle est obtenus par appariement spectral dans le domaine temporel (chapitre 3).

La partie II est consacrée aux problèmes de la conversion du signal vers les symboles en présence d'incertitude. Nous présentons une méthode de classification floue pour l'accès rapide à un grand dictionnaire signal-symbole qui autorise l'association d'un point à plusieurs classes simultanément (chapitre 4) et un système à base de connaissances, capable de traiter des règles imprécises et incertaines et d'effectuer le raisonnement inexact, pour la conversion du signal vers les symboles, lorsque le signal est déformé par le contexte (chapitre 5).

Nous exposons dans la partie III (chapitre 6) un analyseur de structure du signal qui fonctionne en mode contrôlé par les données et par les modèles et qui développe ses solutions en parallèle, avec un mécanisme de recherche en faisceau, afin de propager l'incertitude. Cet analyseur est capable de construire la structure syntaxique à partir de plusieurs îlots de confiance, avec des stratégies différentes.

Une architecture de système d'interprétation des signaux à niveaux et à connaissances multiples est présentée dans la partie IV (chapitre 7). Elle est fondée sur la décomposition du problème d'interprétation en sous-problèmes à des niveaux conceptuels successifs et sur la définition du contrôle explicitement lié à ces niveaux. Chaque niveau a en plus un contrôle local. L'échange d'information est assuré par une structure de données commune entre les sources de connaissances d'un même niveau et par un mécanisme de courrier entre deux niveaux différents.

Enfin, dans la partie V nous illustrons un système d'interprétation de parole continue où l'unité élémentaire de reconnaissance est le phonème et où une segmentation préalable n'est pas nécessaire. Nous proposons un modèle de phonème et l'algorithme de reconnaissance associé, et une méthode de reconnaissance de phrase par localisation des centres syllabiques (chapitre 8). Nous présentons un système opérationel de compréhension du chinois parlé (chapitre 9). Cette partie montre l'utilisation des méthodes développées dans les parties I-IV et valide l'ensemble de nos études dans l'interprétation de signaux incertains.