Détection de mots clés dans un flux de parole :

Application à l'indexation de documents multimédia

Thèse présentée à la Section
Systèmes de Communication / Institut Eurécom

Ecole Polytechnique Fédérale de Lausanne

pour l'obtention du grade de docteur ès sciences techniques

par

Philippe Gelin

Ingénieur civil électricien
de l'Université de Liège
de nationalité belge

Composition du jury :

Président:	Prof M. Hasler (SSC, EPFL)
Directeur de thèse:	Prof. Ch. Wellekens (Institut Eurécom)
Corapporteurs:	Dr. A. Drygajlo (DE, EPFL) Prof. D. Slock (Institut Eurécom) Dr. H. Bourlard (IDIAP) Dr. F. Bimbot (ENST)

Résumé

La quantité d'information multimédia accessible croît de façon vertigineuse. L'avènement de la micro informatique permet à chacun d'apporter sa participation à la création de cette nouvelle source d'information planétaire qu'est "Internet", tandis que les média proposent maintenant la connexion à des bouquets de chaînes de télévision numériques transmis par satellite. L'accès à cette quantité croissante d'information ne s'effectue pas sans problème, et les besoins en outils d'indexation se font cruellement ressentir.

Cette thèse propose diverses solutions pour exploiter les signaux sonores d'un document multimédia afin de repérer les endroits où les mots clés sont prononcés, pour permettre l'indexation plus aisée de ce document.

Cette thèse fixe tout d'abord le cadre de l'étude de l'indexation multimédia et définit les outils nécessaires à son élaboration. Alors que l'indexation de textes écrits existe depuis des décennies, l'indexation du contenu des autres média (images fixes, séquences vidéo, musique, parole) est toujours au stade de développement. Les travaux existants en indexation sur l'image ainsi que ceux sur la reconnaissance de locuteur sont brièvement exposés afin de mieux situer le contexte exact de la thèse qui se focalise sur la détection de mots clés.

Ensuite cette thèse expose les éléments théoriques nécessaires à la mise en oeuvre d'un tel système d'indexation par mots clés. Elle explicite d'une part les méthodes d'analyse du signal acoustique nécessaires à l'extraction des informations caractéristiques de la parole (LPC, PLP, Cepstre, Pitch-Energie), et d'autre part les méthodes de modélisation du langage. On montre comment, en partant d'une modélisation markovienne, deux critères de maximisation peuvent être mis en oeuvre. L'un, classique, est un critère de maximisation de la vraisemblance, et le second, issu d'une théorie émergente (REMAP), est un critère de maximisation de la probabilité a posteriori.

Dans la suite, l'ouvrage se réfère à la littérature spécifique au sujet traité. Il énonce tout d'abord les méthodes existantes pour l'évaluation des systèmes de reconnaissance de parole et montre les contraintes qui y afférent. Ensuite, les recherches successives en détection de mots clés sont présentées en y relevant les idées novatrices. Les récentes avancées dans le domaine voisin qu'est le tri automatique de messages acoustiques sont également exposées.

Après une brève énumération des contraintes spécifiques à l'indexation de la parole par la recherche de mots clés (indépendance du vocabulaire sur lequel porte la recherche, rapidité d'exécution de la recherche, indépendance du locuteur), le manuscrit décrit trois outils de détection de mots clés respectant ces contraintes spécifiques. Le premier de ces outils extrait des segments acoustiques les probabilités qu'ils aient été produits lors de la prononciation de phonèmes. A partir de ces informations, l'outil détecte les régions du signal où la probabilité de présence d'un phonème est élevée et place ces "hypothèses phonétiques" dans un treillis qui sera sauvegardé et utilisé lors des requêtes. Quand une recherche sur un mot donné est nécessaire, il suffit que le système parcoure le treillis à la recherche de la séquence phonétique correspondant au mot recherché pour en effectuer la détection. La tâche est ainsi séparée en une partie préalable à toute détection et qui, de ce fait peut être effectuée par une méthode sophistiquée et précise, et en une autre partie nécessitant un temps de réponse rapide.
Le deuxième outil d'indexation part d'un schéma identique de séparation de la tâche, mais utilise, pour sa part, une modélisation du langage par chaîne de Markov. Il est montré dans la thèse que cette modélisation offre, outre une augmentation des performances vis-à-vis du premier outil, une accélération du processus de recherche sur le treillis.
Le dernier outil mis en oeuvre se base sur les développements récents d'une méthode d'entraînement discriminant des modèles markoviens pour améliorer l'exactitude du treillis phonétique et ainsi produire des résultats de recherche de meilleure qualité.

Finalement les résultats comparatifs entre les différents outils d'indexation sont utilisés pour tirer les conclusions, et envisager les perspectives de futurs développements.

Summary

The amount of accessible multimedia information has been growing drastically. The increasing number of personal computers enables the layman to contribute to the development of Internet, this new worldwide source of information. At the same time, there is an increasing number of cables and satellites operators providing new video services. Access to this growing amount of information is not easy and indexing tools are needed not only by data base managers and professionals involved in archiving but also by private users.

This thesis proposes different solutions for identifying the location of keywords in the audiotrack of multimedia documents in view of a subsequent indexing of the document.

We first give the framework for multimedia indexing and describe its elements. Although text indexing has been widely used for several decades, content based indexing of other media (still images, video, music, speech) is still in its infancy. Earlier work on image indexing and on speaker identification are briefly outlined to better identify the role of keyword spotting which is the focus of this thesis.

We then review the theoretical aspects behind the implementation of our keyword spotting indexing system. On one hand, we explain analysis methods of the signal required to extract the characteristic speech features (LPC, PLP, Cepstrum, Pitch, Energy) and on the other hand, the methods used to model the language. Using Markov models, it is shown how two maximization criteria can be implemented. The first one is standard and relies on the Maximum Likelihood criterion and the second one, an emergent strategy (REMAP), targets to maximize the a posteriori probability.

An overview of the related literature is presented. We describe existing methods for the evaluation of recognition systems and show the associated constraints. The different contributions to the domain are analyzed and their innovative ideas are pointed out and discussed. Automatic sorting of acoustic messages is a parent domain and its recent advances are reviewed.

After a brief enumeration of the constraints specific to speech indexing by keyword spotting (openness of the vocabulary, speed of queries, speaker independency), we describe three indexing tools we developed which satisfy these constraints. Given a series of acoustic segments, the first tool computes the probabilities to associate their utterance with given phonemes. From this information, the tool identifies signal locationss where the probability of presence is high and uses the “phonetic hypotheses” to build a lattice that will be saved and used for query processing. When searching for a word, the lattice is scanned to find the corresponding phonetic transcription. In this manner, indexing task is separated from the query and is achieved off-line in a preliminary sophisticated and accurate processing while the query can be quickened.
The second indexing tool uses the same strategy of task separation but uses Markov models of the phonemes and the language. It is observed that this method accelerates the query and in addition increases the scores.
The third tool relies on a recently described theory of discriminative training where Hidden Markov Models and neural networks are blended to efficiently train the a posteriori probabilities of phonemes given an utterance. The aim is to increase the reliability of the phonetic lattice and increase the scores of the word-spotter.

Finally, the comparative results between our indexing tools are given. Conclusions are drawn and perspectives for future work are proposed.

Remerciements

Mes premiers remerciements vont à Claude Guegen, le directeur de l'Institut Eurécom, qui m'a accueilli dans cet institut et permis d'accomplir ma thèse dans le cadre le plus parfait qu'il soit.

Je tiens également à remercier les membres de mon jury :

Hervé Bourlard qui a guidé mes tous premiers pas dans la reconnaissance de parole,
Frédéric Bimbot qui m'a un jour réservé un accueil chaleureux dans son laboratoire,
Dirk Slock qui m'a si gentiment prêté ses "actes de conférence" et dont la femme fait de si bons gâteaux,
Andrzej Drygajlo qui a aimablement accepté de juger ce travail,
Martin Hasler qui me fait l'honneur de présider ce jury.

Je les remercie pour le temps qu'ils consacreront à la lecture de cet ouvrage et je souhaite qu'ils y trouvent entière satisfaction.

Je tiens également à remercier tous les membres d'Eurécom qui de près ou de loin m'ont permis de croire qu'il était possible de réaliser ce travail et qui m'ont aidé de tout leur savoir, patience et bonne humeur.
Je tiens de plus à remercier tous les membres du département Multimédia, pour avoir, de bonne grâce, subi les surcharges des machines du département tout au long des entraînements de réseaux de neurones.

Plus particulièrement, je souhaite vivement remercier :

Alain que, quoiqu'il m'en défende, je considère un peu comme un grand frère. J'espère qu'il trouvera un jour ce "beat" qu'il cherche désespérément.
Manu, Laurent et Pauline chez qui nous avons pu trouver refuge et trouver une de ces amitiés que l'on oublie jamais.
Stéphane et Soraya qui ont partagé avec nous un grand moment et beaucoup d'autres.
Perrine qui, avec beaucoup de patience, a lu et relu cette thèse et émis nombre de remarques constructives. J'espère qu'elle trouvera dans cette aventure qui commence pour elle autant de satisfaction qu'elle en espère.
Eric et Stéphane qui quoique débordés ont pourtant trouvé le temps pour relire et corriger ce document.
Raymond qui a su trouver le temps de corriger mon anglais.
Et de nouveau Manu et Perrine qui ont réussi à m'inculquer les bases du traitement d'image nécessaire à mon premier chapitre.

Je tiens à remercier mon directeur de thèse, Christian Wellekens qui m'a guidé tout au long de ce chemin et mainte fois prouvé que la recherche peut être gratifiante pour ceux qui savent s'abaisser et s'acharner à la comprendre, mais aussi pour son enthousiasme débordant dès que les mots clés "recherche" et "parole" sont prononcés.

Je voudrais aussi remercier mes parents et ceux de Cécile qui tout au long de cette séparation ont fait preuve de courage et ont su cacher leur tristesse quand il le fallait.
Enfin, il y a mon épouse, qui tout au long de cette épreuve, m'a soutenu, mais aussi et surtout encouragé de tout son coeur pour que ce travail aboutisse. Elle n'a compté ni sa peine ni le temps passé pour m'aider.
Quand le temps était maussade, elle m'encourageait à reprendre le flambeau, quand il était au beau fixe, elle m'encourageait à poursuivre plus loin encore.
C'est aussi pour cela que je lui dédie ma thèse.

Table des Matières

Notations -v

Préface -ix

CHAPITRE 1 **Position du problème** -1

Intérêt des outils d'indexation - 2
Outils vidéo - 4
Détection visuelle de séquences - *4*
Analyse de scènes - *5*
Reconnaissance de visages - *6*
Outils audio - 7
Reconnaissance de mots clés - *7*
Reconnaissance du locuteur - *7*
Autres outils - 8

CHAPITRE 2 **Fondements théoriques**- -9

Les vecteurs acoustiques - 10
L'analyse cepstrale - *10*
Les réseaux de Neurones - 13
Historique - *13*
Principes - *14*
Entraînement - *15*
Surentraînement - *16*
Utilisation en classification - *17*
Vérification expérimentale - *17*
Les chaînes de Markov - 21
Historique - *21*
Modélisation - *22*
Définition des paramètres - *24*
Critère de maximisation pour l'entraînement des modèles - - - - - - - - - *25*
Algorithme de Baum-Welch - 29
Estimation des paramètres - *33*
Réseaux de neurones et algorithme de Baum-Welch - - - - - - - - - - - - - *35*
Algorithme de Viterbi - 39
Estimation des paramètres - *40*

Réseaux de neurones et algorithme de Viterbi - - - - - - - - - - - - - - - - - - **41**
Parcours rapide - **44**
Parcours rapide pour les N meilleurs chemins - - - - - - - - - - - - - - - - - - **46**
Parcours rapide pour l'approximation des N meilleurs chemins. - - - - - **48**
Propagation arrière automatique - **50**
REMAP - **57**
Modélisation - **57**
Critère discriminant - **61**
Formules de récurrence - **64**
Estimation des paramètres - **68**

CHAPITRE 3 **Etat de l'art** - **73**

Les méthodes d'évaluation - **74**
Problématique - **74**
La perplexité - **74**
Estimation du taux d'erreur - **75**
Mots clés - **76**
Tri par le contenu - **77**
Indexation par le contenu - **79**
Lien entre "courbe caractéristique" et "position" - - - - - - - - - - - - - - - **80**
La recherche de mots clés - **81**
Raison d'être - **81**
Applications - **81**
Méthodes existantes - **82**
Identification de sujets - **88**
Lincoln Laboratory - **88**
Dragon System - **90**
Ensigma - **92**
Cambridge University - **93**

CHAPITRE 4 **Solutions proposées** - **97**

Les contraintes de l'indexation - **98**
Indépendance vis-à-vis du locuteur - **98**
Indépendance du contenu lexical du signal à indexer - - - - - - - - - - - - - **98**
Connaissance du mot clé à rechercher au moment même de la requête. - **99**
Solution - **100**
Indexation par étiquetage de trames - **101**
Principe de la méthode - **101**
Probabilités locales - **101**
Génération du treillis - **106**
Analyse du contenu du treillis - **111**
Algorithme de recherche dans le treillis d'hypothèses - - - - - - - - - - - - **115**
Résultats - **117**
Indexation par maximum de vraisemblance - - - - - - - - - - - - - - - - - **129**
Principe de la méthode - **129**
Modèle de langage - **130**
Modèle acoustique - **131**
Génération du treillis - **131**
Recherche dans le treillis - **138**
Résultats - **141**

Indexation par probabilités a posteriori - **145**
Principe de la méthode - ***145***
Génération du treillis - ***145***
Recherche dans le treillis - ***150***
Résultats - ***152***
Comparaison entre les trois méthodes - **155**
Nombre de paramètres - ***155***
Mesures en terme de "position" - ***155***
Mesures en termes de "précision" - ***156***
Mesures en termes de "gain de temps" - ***156***
Mesures par les "courbes caractéristiques" - - - - - - - - - - - - - - - - - - ***157***
Commentaires - ***158***

CHAPITRE 5 **Conclusions & Perspectives** - **159**

Conclusion - **160**
Perspectives - **162**
Estimation du prédicteur acoustique de REMAP - - - - - - - - - - - - - - - - ***162***
Mise en parallèle de Baum-Welch et REMAP - - - - - - - - - - - - - - - - - - ***162***
Représentation phonétique multiple des mots clés - - - - - - - - - - - - - - - ***163***
Robustesse aux bruits - ***163***
Détection du signal de parole - ***164***
Association à d'autres outils d'indexation - - - - - - - - - - - - - - - - - - - ***164***

ANNEXE A **Algorithme de Baum-Welch : Preuve de convergence** - **167**

Plan - **168**
Première Partie - **169**
Concavité de "x log(ax)" - ***169***
Inégalité de Jensen - ***169***
Utilité de la fonction auxiliaire - ***170***
Deuxième partie - **172**
Définition des paramètres - ***172***
Optimisation - ***172***
Estimation des paramètres de transition - ***173***
Estimation des moyennes - ***175***
Estimation des variances - ***177***

ANNEXE B **Algorithme "REMAP" : Preuve de convergence** - - - - - - **179**

Plan - **180**
Première Partie - **181**
Utilité de la fonction auxiliaire - ***181***
Deuxième partie - **183**
Définition des paramètres - ***183***
Optimisation - ***183***
Troisième Partie - **187**
Principe général - ***187***
Critère d'erreur - ***188***
Convergence - ***189***

ANNEXE C **Classification probabiliste par réseaux de Neurones - 193**

Dans le cas discret, avec quantification - - - - - - - - - - - - - - - - - - - 194

Notations - 194

Démonstration - 195

Vecteur d'apprentissage indépendant de la catégorie - - - - - - - - - - - - - 197

Vecteur d'apprentissage indépendant de la classe - - - - - - - - - - - - - - - 198

Autres métriques pour le calcul de l'erreur - - - - - - - - - - - - - - - - - - - 199

Dans le cas discret, sans quantification - - - - - - - - - - - - - - - - - - - 200

Notations - 200

Démonstration - 200

Soit chaque associé à une classe unique - 202

Soit chaque associé statistiquement à chaque classe - - - - - - - - - - - - - 202

Autres métriques pour le calcul de l'erreur - - - - - - - - - - - - - - - - - - - 203

Dans le cas continu - 204

Notations - 204

Démonstration - 204

Soit , une fonction connue reliant tout élément à une et une seul classe 206

Soit chaque associé statistiquement à chaque classe - - - - - - - - - - - - - 207

Bibliographie - 209

Index - 219

Curriculum Vitae - 1

Notations

$x = \{v_1, \ldots, v_{N_a}\}$ Un *vecteur acoustique*, décrivant, à l'aide de N_a coefficients, le signal acoustique émis durant une période d'analyse de l'ordre de la centiseconde.

$X = \{x_1, \ldots, x_{N_s}\}$ Une *séquence acoustique*, définie comme une séquence de N_s vecteurs acoustiques.

X_b^e Une sous-séquence de X, tel que $X_b^e = \{x_b, \ldots, x_e\}$.

$\mathcal{E} = \{X_{\mathcal{E}, 1}, \ldots, X_{\mathcal{E}, N_{\mathcal{E}}}\}$

L'ensemble des $N_{\mathcal{E}}$ séquences acoustiques associées à la base de données d'entraînement.

$\mathcal{T} = \{X_{\mathcal{T}, 1}, \ldots, X_{\mathcal{T}, N_{\mathcal{T}}}\}$

L'ensemble des $N_{\mathcal{T}}$ séquences acoustiques associées à la base de données de test.

$\Phi = \{\varphi_1, \ldots, \varphi_P\}$ L'ensemble des P phonèmes existant dans la langue modélisée.

$\phi = \{\varphi_1, \ldots, \varphi_N\}$ Une *séquence phonétique* de N phonèmes pouvant représenter un mot clé.

$\mathcal{Q} = \{q_1, \ldots, q_K\}$ L'ensemble des K états markoviens utilisés pour la modélisation du langage étudié.

q_k^t L'évènement que constitue la visite de l'état q_k au temps t.

M Un *modèle de Markov* construit à partir d'états $q_k \in \mathcal{Q}$.

$M_{\mathcal{E},\,i} = \{M_{\varphi(\mathcal{E},\,i,\,1)}, \ldots, M_{\varphi(\mathcal{E},\,i,\,N_{\mathcal{E},\,i})}\}$

Le $i^{\text{ème}}$ modèle markovien composé de l'enchaînement des $N_{\mathcal{E},\,i}$ modèles de la transcription phonétique de la séquence acoustique $X_{\mathcal{E},\,i}$.

$\mathcal{M} = \{M_{\mathcal{E},\,1}, \ldots, M_{\mathcal{E},\,N_{\mathcal{E}}}\}$

L'ensemble des $N_{\mathcal{E}}$ modèles markoviens décrivant les $N_{\mathcal{E}}$ séquences phonétiques de $\mathcal{E}$.

$\gamma = \left\{q^{1}_{\gamma_1}, q^{2}_{\gamma_2}, \ldots, q^{N}_{\gamma_N}\right\}$

Un *chemin*, défini par une séquence d'état, $q_{\gamma_i} \in Q$.

Si ce chemin est implicitement lié à une séquence acoustique X de longueur N, ce chemin sera considéré de longueur N pour garder toute consistance.

Si ce chemin est implicitement lié à un modèle M, on supposera ce *chemin valide* vis-à-vis de M, c'est-à-dire pouvant être généré par le processus markovien décrit par M.

$\gamma_m^n = \left\{q^{m}_{\gamma_m}, q^{m+1}_{\gamma_{m+1}}, \ldots, q^{n}_{\gamma_n}\right\}$

Une *section du chemin* γ.

$\Gamma = \{\gamma_1, \ldots, \gamma_{N_\Gamma}\}$

L'ensemble de tous les chemins.

Si cet ensemble est implicitement lié à une séquence acoustique X de longueur N, cet ensemble contiendra tous les chemins de longueur N.

Si cet ensemble est implicitement lié à un modèle M, cet ensemble contiendra tous les chemins valides vis-à-vis de ce modèle. En cas de confusion cet ensemble sera noté Γ_M.

$\Lambda = \{\lambda_1, \ldots, \lambda_{N_\Lambda}\}$ L'ensemble des N_Λ paramètres décrivant un modèle.

$P(A)$ La probabilité d'un évènement A.

Préface

Grâce au développement des technologies numériques, le foisonnement d'informations n'a jamais été aussi important qu'en cette fin de XXème siècle. La création de cette nouvelle source de données planétaire qu'est Internet et son expansion rapide constituent les prémices de cette nouvelle ère de l'information.

Cette évolution n'est pas sans poser de problèmes, qu'ils soient d'ordre technique (bande passante pour la parole et la vidéo, synchronisation et qualité de service), d'ordre moral (protection des libertés d'expressions et violation des droits d'auteurs), ou d'accessibilité (moteurs de recherche, analyse et recherche par le contenu, mise à jour afin d'éviter la fossilisation de l'information).

Cette thèse porte sur l'intérêt et la faisabiliter de la détection de mots clés dans un flux de parole et sur son application à l'indexation de documents multimédia. Elle se situe clairement dans l'amélioration des techniques favorisant l'accès aux sources de données multimédia. En développant des outils d'analyse par le contenu, on apporte non seulement une aide substantielle à l'extraction d'informations utiles, mais aussi un outil de supervision, permettant la vérification du contenu du flux d'informations.

Cette thèse s'articule en cinq chapitres :

Le premier chapitre intitulé “Position du problème”, montre l'intérêt d'un outil d'indexation multimédia basé sur la détection de mots clés et le situe parmi les autres outils d'indexation existants. Nous y verrons que la recherche d'informations prend de plus en plus d'importance dans un monde où l'information directement accessible croît de façon exponentielle.

Le second chapitre, “Fondements théoriques”, reprend les bases théoriques de la reconnaissance de parole et montre les modifications nécessaires à l'élaboration d'un tel outil. Nous analyserons la méthode employée pour modéliser le signal de parole et en extraire les caractéristiques pertinentes pour la reconnaissance de parole.
Par la suite, nous insisterons sur l'intérêt des réseaux de neurones en tant que classificateurs et nous montrerons comment ils peuvent être couplés aux méthodes classiques de reconnaissance de parole.
Après un bref rappel historique des chaînes cachées de Markov, nous mettrons en évidence les hypothèses pour adapter cette théorie à la reconnaissance de parole. Nous présenterons égale-

ment une approche originale pour faciliter l'utilisation des chaînes de Markov sur de très longs signaux de parole.
Nous présenterons par la suite une théorie émergente, "REMAP", permettant de soulager l'approche probabiliste d'hypothèses contraignantes. Nous y soulignerons son caractère discriminant, la comparerons avec la théorie classique et en déduirons de nouvelles relations.

Dans le troisième chapitre, "Etat de l'art", nous présenterons les mesures existantes pour évaluer les différents systèmes de reconnaissance de parole. Nous verrons leurs spécificités vis-à-vis de la tâche que doit remplir le système de reconnaissance, et sélectionnerons la mesure la plus appropriée. De plus, nous nous appliquerons à retracer l'historique de la détection de mots clés et y soulignerons les faits marquants qui ont jalonné l'évolution de ces systèmes de reconnaissance. Nous nous pencherons spécialement sur les travaux menant au tri automatique de messages vocaux, et attacherons un soin particulier à y relever les progrès réalisés ces dernières années.

Dans le quatrième chapitre, "Solutions proposées", nous préciserons tout d'abord les contraintes imposées par l'indexation et expliquerons comment cette tâche peut être scindée en deux parties. La première extrait des hypothèses sur la position des phonèmes dans le signal acoustique et conserve celles-ci dans une structure en treillis. La deuxième partie utilise l'information contenue dans ce treillis pour détecter les occurrences des mots clés.
Par la suite, nous développerons en détail trois méthodes originales respectant ces contraintes spécifiques:
- la première d'entre elles est fondée sur une détection phonétique par le biais d'un étiquetage des trames acoustiques ;
- la deuxième s'appuie sur une modélisation probabiliste des phonèmes par les chaînes de Markov ;
- la troisième utilise le processus de reconnaissance issu de la théorie "REMAP" afin d'exploiter son caractère discriminant.
Pour chacune de ces méthodes, nous expliquerons successivement la création du treillis d'hypothèses phonétiques, l'utilisation d'une matrice de confusion augmentant la flexibilité du système et les algorithmes de recherche mis en oeuvre pour la détection des mots clés.
Nous achèverons ce chapitre par la confrontation des trois méthodes. Nous exposerons les résultats obtenus lors de leurs évaluations respectives.

Finalement nous présenterons au dernier chapitre, "Conclusions & Perspectives", les conclusions que nous pouvons tirer des développements précédents et nous nous permettrons d'énoncer quelques perspectives possibles de ce travail.

CHAPITRE 1

Position du problème

Dans ce chapitre, nous situons le contexte de notre étude, à savoir l'indexation de documents multimédia.

Après avoir justifié la nécessité d'un tel outil, nous survolons les différentes méthodes mises en oeuvre pour l'étiquetage de ces documents tant au niveau visuel qu'acoustique.

Ceci permettra de situer et de caractériser notre travail dans un système global d'indexation.

L'utilisateur s'est trouvé fort dépourvu
Lorsque le multimédia fut venu.
Plus de repères, ni la moindre référence
Pour extraire le renseignement souhaité,
Tant le débit d'informations était élevé.

1.1 Intérêt des outils d'indexation

Depuis quelques années déjà, on perçoit une mutation des outils de stockage d'information passant de l'analogique au numérique. Les systèmes numériques ont surclassé les systèmes analogiques car, outre un prix sans cesse à la baisse, ils offrent des outils de traitement jusqu'alors inimaginables.

En effet, comme le montre la figure 1, le nombre de transistors composant les ordinateurs croît de manière exponentielle, ce qui s'accompagne d'une augmentation proportionnelle de la puissance de calcul directement accessible et conduit à la généralisation du numérique.

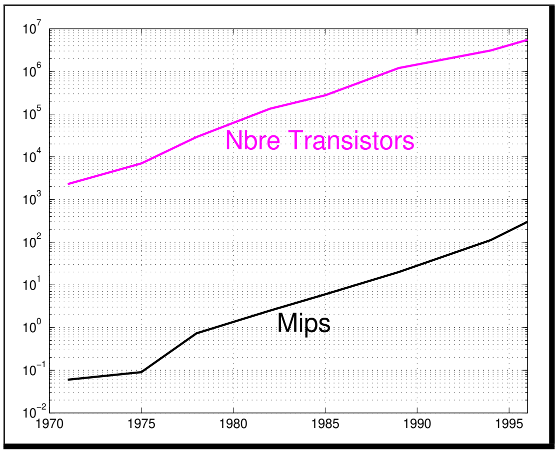

FIGURE 1. **Evolution de la puissance de calcul en micro informatique.**

La capacité de stockage numérique de l'information a suivi une même courbe d'évolution. Citons par exemple, les premières disquettes qui contenaient 360 Kbytes, puis 720 Kbytes et 1,2 Mbytes. Le standard actuel tend vers 100 Mbytes. L'arrivée du CD-ROM participa également à cette évolution. Les méthodes de codage actuelles permettent d'inscrire sur les 740 Mbytes d'un CD-ROM, plus de 5 heures de musique ou près de 40 heures de parole, ou encore 1 heure de séquences vidéo de haute qualité ou enfin 5000 images haute résolution.

La transmission de l'information a également subi de forts changements ces dernières décennies. Alors que la télévision couleur constituait il y a à peine 20 ans une révolution, nous parlons maintenant de bouquets de chaînes numériques transmis par satellite. Le progrès le plus impressionnant dans la transmission des informations reste sans aucun doute l'avènement du réseau Internet qui, bien que ne procurant encore qu'un faible débit d'information, génère un engouement hors pair parmi la population (voir figure 2).

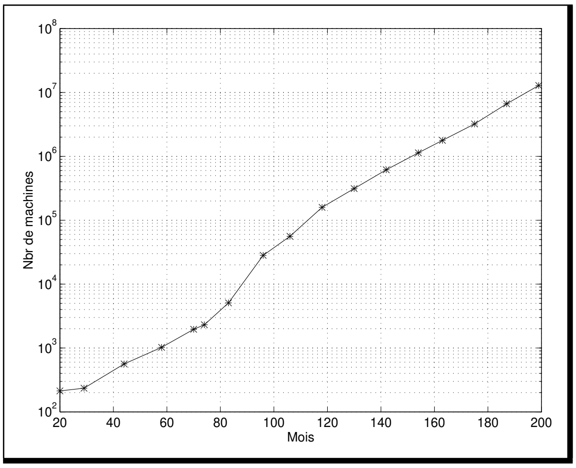

FIGURE 2. **Progression du parc de machines connectées à internet, depuis août 1981.**

Le mariage entre ces moyens de traitement, transmission et stockage de l'information a permis à l'utilisateur de tels services d'accéder à une quantité d'informations jusqu'alors inimaginable.

De plus, les bases de données accessibles, qui contenaient initialement des textes, se voient de plus en plus enrichies d'éléments hétérogènes. D'abord l'image fixe, puis le son et finalement l'image animée s'ajoutent aux données textuelles, accroissant ainsi la diversité des informations disponibles.

Ce bouleversement rapide n'est pas sans inconvénient. En effet, l'utilisateur qui préalablement gérait au cas par cas les informations qu'il recevait, s'est trouvé fort dépourvu lorsque le multimédia fut venu. Plus de repères, ni la moindre référence pour extraire le renseignement souhaité, tant le débit d'informations était élevé.

La nécessité d'outils pour trier ces informations se fait cruellement sentir, et les systèmes d'indexation commencent à voir le jour. Cependant leur développement est difficile car les informations conservées dans les bases de données multimédia sont par définition très diverses. Outre le texte pour lequel des traitements sont connus et la musique pour laquelle on a montré peu d'intérêt jusqu'à présent, l'image et la parole représentent les média sur lesquels la recherche en indexation est actuellement focalisée.

Outre les problèmes rencontrés pour satisfaire la requête, s'ajoute la difficulté de formuler cette requête. Exprimer la demande de recherche des apparitions du visage d'une personne dans un film ou d'un évènement particulier dans une séquence vidéo pose encore un réel problème. Par contre, décrire le mot recherché parait une approche plus simple quoique non triviale étant données les diverses transcriptions phonétiques d'un même mot plongé dans le contexte différent.

Passons rapidement en revue les différents traitements que l'on peut effectuer sur l'image et la parole en vue de l'indexation des données multimédia.

1.2 Outils vidéo

Les séquences vidéo sont riches en informations. Il peut s'agir d'informations de bas niveau comme la visualisation d'un objet ou d'informations ayant un contenu sémantique plus profond comme la dispute entre deux acteurs, ou une déclaration d'amour le soir sur une terrasse de Manhattan. Pour effectuer une indexation la plus complète possible, les méthodes d'indexation automatique devront se baser sur des outils d'extraction exploitant toutes les facettes de l'analyse.

Dans les paragraphes suivants, nous relevons rapidement quelques-uns de ces outils. Nous survolerons tout d'abord les méthodes utilisées pour la détection de césures de plans et pour l'extraction d'éléments composant l'image. Nous décrirons ensuite succinctement la reconnaissance de visages.

1.2.1 Détection visuelle de séquences

Le but de cet outil est de segmenter la bande vidéo en fonction des différents plans utilisés lors de l'enregistrement. Une séquence dans un film, tout comme un paragraphe dans un texte, représente un message que le cinéaste veut faire passer. Tout comme on segmente un paragraphe en phrases, le cinéaste utilise un ensemble de plans pour construire sa séquence. Ces plans peuvent être séparés de différentes manières. On peut utiliser des coupures nettes, mais aussi des fondus enchaînés ou des passages par fond grisé.

1.2.1.1 Rupture rapide

Dans le cas de ruptures brutales, des méthodes simples basées sur le traitement entre deux images successives peuvent être mises en oeuvre avec un taux de détection élevé. La méthode la plus simple consiste à comparer deux images successives en fonction de l'intensité de leurs pixels (en niveaux de gris pour accélérer le processus, ou en couleurs pour obtenir des valeurs plus précises, [NAG91]). Cependant, cette méthode est sensible à différents facteurs tels que la variation rapide de luminosité, le mouvement rapide de la caméra ou d'objets dans la scène, et peut dès lors générer de fausses détections ou ne pas remarquer les transitions douces. Différentes évolutions sont apparues pour palier ces défauts. Citons l'utilisation d'histogrammes d'intensité pour réduire l'effet de mouvements légers d'objets de la scène ; le filtrage passe bas de l'image dans le but d'atténuer le bruit d'enregistrement et dans une moindre mesure, les effets dus aux mouvements, [CHER95] ; l'utilisation du champ de vecteurs déplacements pour détecter les transitions brutales, tout en rejetant les zooms ou les travellings rapides, [AKU92]. Ces méthodes impliquent une charge de calcul élevée, et dans le but de réduire cette charge, différentes méthodes ont été proposées : l'analyse par blocs qui étudie les variations statistiques de sections de l'image, mais qui impose un choix dans la sélection de la taille et la position des blocs dans l'image, [KAS91] ; le traitement direct sur les images comprimées comme par exemple dans l'utilisation de format “Jpeg”, où les coefficients de la DCT sont comparés entre deux images successives, [ARM93].

1.2.1.2 Rupture lente

Dans la détection de transitions progressives entre plans, la transition porte sur plusieurs images successives. La détection doit donc elle aussi être basée sur plusieurs images regroupées. Pour ce faire, on compare souvent la différence entre deux images contiguës et celle obtenue entre l'image courante et une image plus distante, supposée être au début de la transition,[ZHA93]. L'étude des champs de vecteurs déplacements s'impose si l'on désire réduire au minimum les effets de travelling ou de zoom. Une autre méthode, [AIG94], consiste à analyser la variation de niveaux de gris qui, quoique faible entre deux images, progresse toujours dans le même sens si l'on tient compte de plusieurs images consécutives.

1.2.2 Analyse de scènes

Le but est ici de décrire le contenu réel des images. Partant d'observations de bas niveau (les pixels, changement de séquences,...) on recherche des éléments de niveau de plus en plus élevé (surfaces planes, frontière entre objets, détection d'objets,...). Si pour les bas niveaux, une analyse du signal est généralement suffisante, l'analyse des niveaux hauts (une personne entre dans une pièce, prend un objet,...) fait généralement appel à des techniques telles que l'intelligence artificielle. Le travail d'analyse de scènes peut généralement être séparé de la manière suivante : d'une part l'analyse d'images fixes, et d'autre part l'analyse d'images animées.

1.2.2.1 Images fixes

Nombreux sont les traitements de bas niveau que l'on peut appliquer aux images. Les plus connus sont :

- l'analyse par histogramme, qui permet d'obtenir des informations sur la distribution des niveaux de gris dans l'image et ainsi de séparer le fond de l'image et les objets ;
- la détection de contours et de régions par Laplacien ou gradient ;
- la détection de formes géométriques par transformée de Hough (voir par exemple [CAN83]).

Une fois ces éléments détectés, nous pouvons construire les primitives de moyen niveau. Les frontières peuvent être approximées par des formes géométriques simples (droites, courbes polynomiales,...) afin d'être manipulées plus facilement dans les niveaux supérieurs. Les éléments ainsi obtenus peuvent être regroupés pour former des objets plus structurés et plus proches du monde réel, (voir par exemple [BER87]).

Il faut ensuite reconnaître ces objets. Toutes les méthodes de classification peuvent être utilisées, telle la quantification vectorielle, les graphes relationnels ou les réseaux de neurones.

1.2.2.2 Images dynamiques

Pour l'analyse d'images dynamiques, des traitements analogues à ceux du paragraphe précédent restent de mise. Cependant, il est possible d'extraire davantage d'informations des séquences dynamiques, comme par exemple le suivi d'objets ou la déformation de ceux-ci le long de la séquence. Pour la détection de mouvements, on peut citer les méthodes basées sur la corrélation (recherche de la correspondance entre deux blocs) ou les méthodes différentielles s'appuyant sur l'hypothèse Lambertienne, [HIL82]. Pour le suivi et la déformation de mouvement, citons l'utilisation des contours actifs ("snakes") qui une fois placés sur les contours d'un objet, peuvent se déformer suivant un modèle complexe dans le but de les suivre le long de la séquence. Quant à la modélisation et à la prédiction de mouvements, citons les modèles physiques de trajectoires et les filtres de Kalman.

1.2.3 Reconnaissance de visages

Reconnaître le visage des personnages intervenant dans les séquences vidéo apporte indéniablement un plus à l'indexation. Cependant, elle est confrontée à une multitude de problèmes : l'orientation du visage varie, l'éclairage n'est pas constant, le facteur d'échelle utilisé peut être différent, l'expression de la personne peut changer et des accessoires (lunettes, barbe) peuvent troubler le système de reconnaissance.

Plusieurs méthodes ont été envisagées. Citons les plus connues :

- L'utilisation de traits caractéristiques tel que la taille des yeux et leur position relative. Cependant, ces caractéristiques sont sensibles aux conditions d'éclairement, d'orientation et d'échelle, [KAY72].
- La comparaison de profils extraits d'images 2D ou d'un modèle 3D. Cette méthode est sensible aux bruits tel que le déplacement des lèvres et elle nécessite une base de données haute résolution, [KAUF76].
- La comparaison de valeurs propres extraites directement des images ("Analyse en Composantes Principales"). Ceci nécessite autant d'images de référence que d'orientations possibles, [TUR91].
- Les réseaux de neurones ou les champs de Markov dont l'apprentissage nécessite une grande base de données, [STO84].
- Le calcul d'invariants projectifs qui implique une extraction robuste des points caractéristiques, [KAM93].
- La modélisation 3D, soit à partir d'une succession d'images 2D qui nécessite dès lors un grand nombre d'images, soit à partir d'un modèle 3D déformable suffisamment flexible et insensible aux bruits, [TER88].

1.3 Outils audio

1.3.1 Reconnaissance de mots clés

La reconnaissance de mots clés est un outil susceptible de prendre une place importante parmi l'ensemble des méthodes d'indexation tant son rôle est complémentaire aux outils préalablement exposés. En effet, elle repose sur des ressources indépendantes du signal visuel ou des éventuels signaux de synchronisation. Elle est donc insensible à une quelconque dégradation des signaux visuels. De plus, elle apporte une information complémentaire à celles obtenues par traitement de l'image.

Cependant, pour conserver toute son efficacité, la reconnaissance de mots clés doit pouvoir s'adapter aux situations les plus variées possibles. Elle doit être notamment faire preuve de robustesse face au bruit de fond et rester indépendante des locuteurs. De plus, la recherche de mots clés doit pouvoir être appliquée sur tous les mots imaginables, aussi bien issus d'un vocabulaire entraîné que de noms propres prononcés pour la première fois. Comme nous allons le voir par la suite, elle présente de plus l'avantage d'offrir une réponse rapide aux requêtes d'indexation, quelque soit leur contenu.

1.3.2 Reconnaissance du locuteur

Ce problème est le dual de la reconnaissance de parole. Il nécessite en effet d'extraire de la parole les caractéristiques propres aux différents locuteurs tout en restant, si possible, indépendant des mots prononcés, [BIM93]. L'utilisation d'un tel outil dans l'indexation impose également certaines contraintes qui impliquent une réduction du taux de reconnaissance. Les modèles les plus robustes, tels ceux utilisés pour les problèmes de vérification, se basent généralement sur la prononciation d'une phrase pré-déterminée pour identifier le locuteur. Cette méthode n'est cependant pas envisageable dans le cadre de l'indexation. D'autres procédés, comme l'extraction et la reconnaissance de phonèmes voisés ont été utilisés pour comparer différents locuteurs sur des phrases différentes, [HUE95].

Un autre problème introduit par l'indexation, est la détermination du nombre de locuteurs dans la bande sonore. Une difficulté sous-jacente est le repérage des zones de transition entre les différents locuteurs. Pour ce faire, il faut au préalable segmenter le signal de parole de manière à obtenir des segments où un seul locuteur intervient. La segmentation fournit en général des régions trop courtes pour une détection fiable du locuteur. Il faut donc regrouper les régions de parole offrant les mêmes caractéristiques.

1.4 Autres outils

Les outils de reconnaissance que l'on peut utiliser pour l'indexation sont encore nombreux. Le plus trivial reste la reconnaissance visuelle de caractères que l'on peut extraire des séquences vidéo. Dans le cas d'indexation de journaux télévisés, cet outil peut s'avérer très utile voire même autosuffisant. Citons également les efforts effectués actuellement pour utiliser le mouvement des lèvres lors la reconnaissance de parole, [ADJ96].

Ce monde est en constante mutation et il ne serait pas surprenant que d'autres outils apparaissent bientôt. En attendant, il nous reste à présenter celui-ci...

CHAPITRE 2

Fondements théoriques

Après avoir précisé le contexte de notre étude, considérons à présent le problème de détection de mots clés dans un enregistrement sonore.

Nous étudierons tout d'abord l'analyse du signal sonore fournissant les vecteurs acoustiques contenant l'information pertinente pour la reconnaissance de parole.

Nous mettrons ensuite en évidence l'intérêt des réseaux de neurones et leur rôle dans la classification des vecteurs acoustiques.

Ensuite, nous aborderons l'étude de la modélisation du langage. Pour ce faire, nous introduirons le concept de chaînes de Markov et soulignerons les hypothèses sous-jacentes à leur utilisation.

Nous décrirons ensuite une technique novatrice (REMAP) développée à l'ICSI permettant une maximisation directe des probabilités a posteriori associées aux modèles de parole, ainsi qu'un apprentissage discriminant de ces modèles.

2.1 Les vecteurs acoustiques

Le pré-traitement du signal de parole, est d'une importance capitale si l'on désire obtenir de bons résultats dans les systèmes de reconnaissance.

Nombre d'études ont été effectuées pour déterminer non seulement les coefficients les plus représentatifs de la parole, mais aussi les plus représentatifs du locuteur, les plus robustes au bruit, les moins coûteux en temps de calcul ou minimisant le débit de transmission. Toutefois, cette thèse ne se focalise pas sur les optmisations éventuelles auquelles on peut s'attendre en analysant plus en détails les différentes méthodes de prétraitement du signal. Nous avons préféré conserver une même méthode d'analyse acoustique tout au long du travail, de manière à permettre des comparaisons plus aisées entre les différentes solutions proposées.

Nous avons choisi l'analyse cepstrale comme pré-traitement du signal car cette méthode est connue pour sa robustesse vis à vis du bruit, mais aussi vis à vis des différents locuteurs.

Avant d'étudier plus en détail cette analyse, notons que le signal de parole est préalablement numérisé en conformité avec le théorème d'échantillonnage avec une fréquence d'échantillonnage de 16 KHz et avec une précision de 16 bits par échantillon.

2.1.1 L'analyse cepstrale

En reconnaissance de parole, on recherche les caractéristiques du signal sonore spécifique au message prononcé tout en évitant les caractéristiques propres à la prononciation du locuteur. Or, il a été montré que l'enveloppe du spectre du signal était crucial pour la compréhension tandis que la structure fine du spectre était surtout spécifique au locuteur. (voir figure 3)

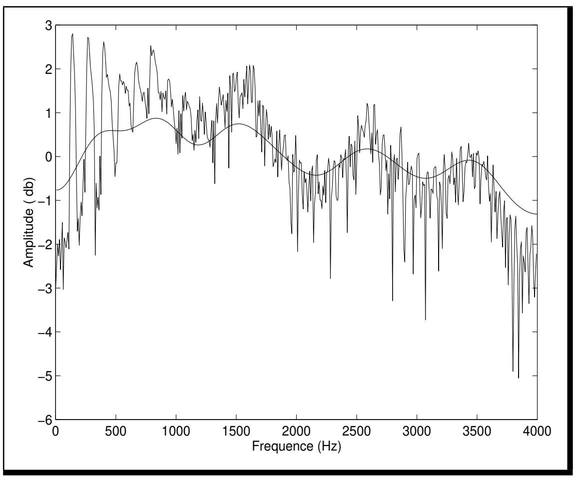

FIGURE 3. **Spectre, d'un son voisé, lissé par analyse cepstrale.**

L'idée sous-jacente à l'analyse homomorphique est de séparer cette structure fine, générée par l'excitation, de l'enveloppe, résultant du filtrage de cette excitation par le conduit vocal.

Pour ce faire, on définit le cepstre par :

$$c(\tau) = \mathcal{F}^{-1}\{\log|\mathcal{F}\{s(t)\}|\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|S(\omega)|e^{j\omega\tau}d\omega, \qquad \textbf{(EQ 1)}$$

où $\mathcal{F}\{\ \cdot\ \}$ représente la transformée de Fourier discrète et τ est appelée quéfrence.

De cette façon, en notant $g(t)$ le signal d'excitation, et $h(t)$ la réponse impulsionnelle du conduit vocal, on a :

$$s(t) = \int_{-\infty}^{\infty} g(\tau)h(t-\tau)d\tau \quad \Leftrightarrow \quad S(\omega) = G(\omega)H(\omega),$$

qui introduit dans **(EQ 1)** conduit à :

$$c(\tau) = \mathcal{F}^{-1}\{\log|S(\omega)|\} = \mathcal{F}^{-1}\{\log|G(\omega)|\} + \mathcal{F}^{-1}\{\log|H(\omega)|\}.$$

Pour un signal voisé, le premier terme dû à l'excitation se traduit par la présence d'un pic et de ses harmoniques éventuelles dans les hautes quéfrences. Le second terme induit pour sa part les composantes basses quéfrences. (voir figure 4)

Il suffit donc, de séparer les basses quéfrences et les hautes quéfrences ("liftrer passe-bas") pour extraire, dans le domaine spectral, l'enveloppe du signal comme le montre la figure 3.

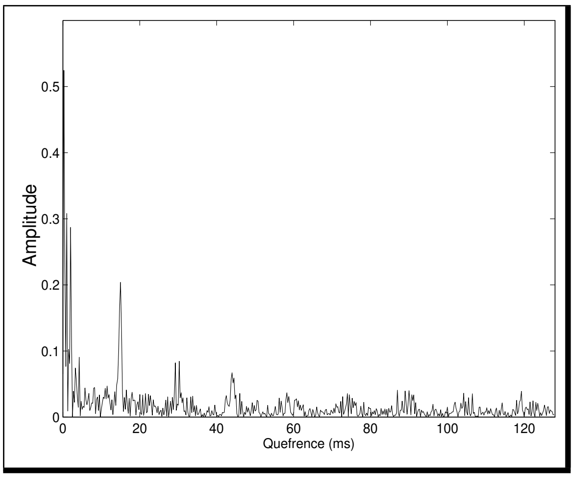

FIGURE 4. Analyse cepstrale d'un son voisé.

2.2 Les réseaux de Neurones

Les réseaux de neurones constituent un outil de classification flexible. Ils ont été appliqués à de nombreux problèmes où la modélisation explicite des relations cause-effet a résisté à toute analyse conventionnelle, en particulier dans le cadre de la reconnaissance de parole. Il nous paraît donc utile de souligner leurs caractéristiques principales ayant une implication directe dans la reconnaissance de parole. Il est clair que le rôle de classificateur est prépondérant dans cette application, et nous nous y cantonnerons.

En 1988, Bourlard et Wellekens [BOU88] montrent qu'un réseau de neurones peut estimer les probabilités a posteriori qu'un élément placé à l'entrée du réseau (un vecteur acoustique par exemple) appartienne à une classe représentée par une des sorties du réseau (une classe phonétique par exemple). Cette propriété est développée en annexe C.

Nous reprenons ici rapidement les bases des réseaux de neurones ainsi que quelques résultats intéressants.

2.2.1 Historique

Les réseaux de neurones sont apparus dans les années 40 lorsque McCulloch et Pitts proposèrent un modèle de calcul basé sur de simples éléments logiques mimant les relations neuronales. En 1949, D. Hebb énonça sa règle d'apprentissage, aussi connue sous le nom de loi delta, et permit ainsi le développement d'intérêt que la communauté scientifique lui porta jusque dans les années 60 (Rosenblatt, Widrow& Hoff) où elle tomba en léthargie. Ce n'est que dans les années 80 avec les travaux de Hopfield, que l'approche neuronale connut un regain d'intérêt. Rumelhart, Hinton et Williams, en 1986, généralisèrent la loi delta et développèrent une méthode efficace d'entraînement des réseaux multicouches.

Durant ces dernières années, les réseaux de neurones ont été appliqués à des problèmes aussi variés que complexes : reconnaissance de visages, contrôle de robots, identification de locuteurs, reconnaissance de langages, reconnaissance de phonèmes, détection de mots clés, ...

2.2.2 Principes

L'élément de base peut être décrit par la figure ci-dessous :

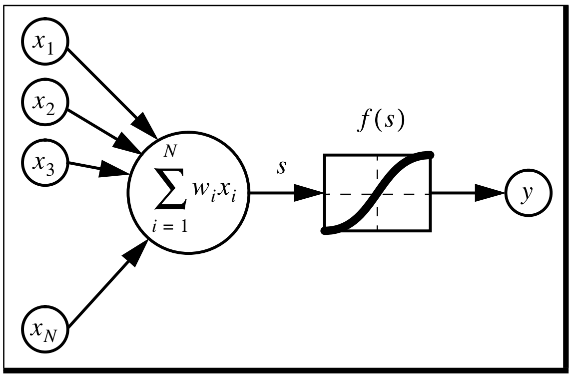

FIGURE 5. **Neurone.**

où x_i correspond à l'entrée du neurone et y à sa sortie. La fonction $f(s)$, appelée *fonction d'activation* peut avoir de multiples formes. Les plus courantes sont les fonctions de forme sigmoïdale telles que :

$$f(s) = \frac{1}{1 + e^{-\frac{s}{T}}},$$

où T représente un terme définissant la pente de la fonction,

et les fonctions de type hyperbolique :

$$f(s) = tanh(s)$$

Les réseaux les plus courants en reconnaissance de parole restent les réseaux de type perceptron multicouches ("Multi Layer Perceptron", "MLP"). R. P. Lippman, montre en 1987 [LIPP87], que 3 couches sont suffisantes pour générer n'importe quelle séparation de l'espace d'entrée. Nous nous cantonnerons donc à l'utilisation de réseaux ayant une seule couche cachée, tel celui décrit dans la figure ci-dessous :

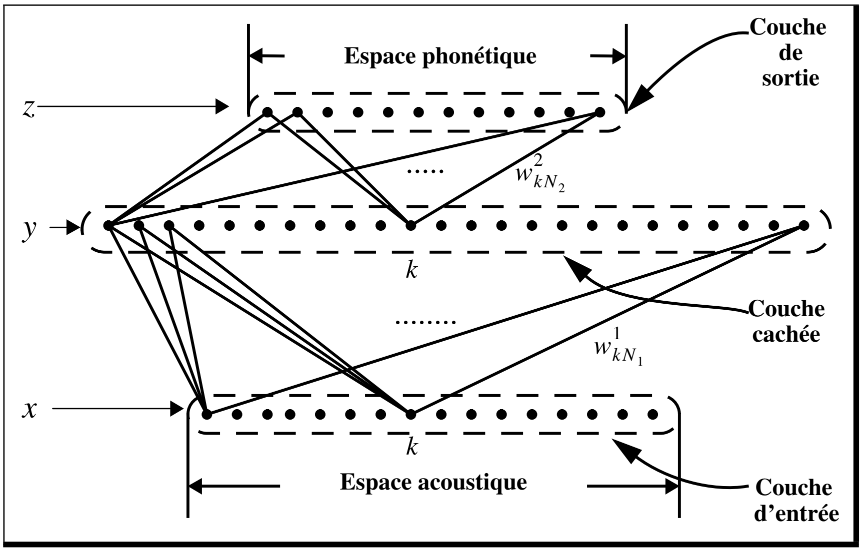

FIGURE 6. Réseau de neurones à une seule couche cachée.

2.2.3 Entraînement

L'entraînement du réseau de neurones consiste à minimiser un critère d'erreur. Ce critère peut être quadratique :

$$E = \frac{1}{2} \sum_{n=1}^{N_X} \sum_{l=1}^{N_Q} (z_l(n, \Theta) - d_l(n))^2, \qquad \text{(EQ 2)}$$

où Θ représente l'ensemble des paramètres du réseau (principalement les poids $w_{ij}^{1,2}$), $z_l(n, \Theta)$ représente la valeur obtenue à la $l^{\text{ième}}$ sortie lorsque l'on applique le $n^{\text{ième}}$ vecteur acoustique à l'entrée, et $d_l(n)$ correspond à la sortie désirée.

D'autres types de minimisation existent, citons simplement celle basée sur l'entropie relative :

$$E = \frac{1}{2} \sum_{n=1}^{N_X} \sum_{l=1}^{N_Q} \left((1 + d_l(n)) \ln\left(\frac{1 + d_l(n)}{1 + z_l(n, \Theta)}\right) + (1 - d_l(n)) \ln\left(\frac{1 - d_l(n)}{1 - z_l(n, \Theta)}\right) \right). \qquad \text{(EQ 3)}$$

La minimisation du critère d'erreur choisi est effectuée en dérivant les équations précédentes par rapport aux différents poids. On en déduit ainsi la loi delta généralisée qui modifie les poids du réseau.

Pour la couche de sortie, nous avons :

$$\Delta w_{ij}^2 = \eta(d_i - f(s_i^2))f'(s_i^2)y_j,$$

et pour la couche cachée :

$$\Delta w_{ij}^1 = \left\{\sum_j [w_{ij}^2 \Delta w_{ij}^2]\right\} f'(s_i^1)x_j,$$

où η représente le taux d'apprentissage.

2.2.4 Surentraînement

Le surentraînement est l'un des phénomènes que l'on peut rencontrer lors de l'apprentissage de réseau de neurones. Si le réseau possède trop de degrés de liberté par rapport à la complexité du problème, il aura tendance à apprendre les exemples du problème qu'on lui soumet à l'entraînement, et cela au détriment du caractère de généralisation que l'on attend du système. La méthode la plus fréquemment utilisée pour palier ce défaut consiste à constamment tester l'efficacité du réseau sur une partie de données non utilisées pour l'entraînement. Cette méthode porte le nom de validation croisée ("cross validation"). Nous montrons dans la figure ci-dessous un exemple typique de comportement d'entraînement.

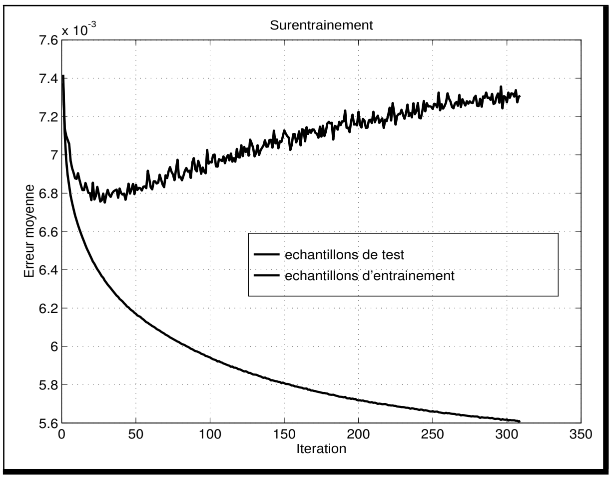

FIGURE 7. **Courbes d'erreurs sur les échantillons de test et d'entraînement.**

2.2.5 Utilisation en classification

L'utilisation de réseaux de neurones en vue de la classification de données se présente comme suit :

Chaque sortie du réseau correspond à une des classes. Ceci implique directement que le nombre de classes soit préalablement fixé. En reconnaissance de parole, nous choisirons évidemment les classes phonétiques.

L'entrée du réseau correspond généralement aux vecteurs acoustiques que l'on désire classer.

Cependant, l'analyse statistique montre que les probabilités dépendent des vecteurs voisins. Une approche analytique à été proposée par Ch. Wellekens, [WELL87], pour prendre en compte les vecteurs voisins dans la définition des probabilités. Une autre façon de tenir compte de cet effet est d'adjoindre les vecteurs acoustiques temporellement proches ("Entrée contextuelle") ou une représentation de l'état précédent (Voir "REMAP", section 2.6), à l'entrée du réseau de neurones.

Comme il est montré dans l'annexe B, la minimisation du critère d'erreur lors de la classification de phonèmes induit l'apprentissage des probabilités a posteriori.

2.2.6 Vérification expérimentale

Il est aisé de vérifier expérimentalement la validité des sorties d'un réseau de neurones vis-à-vis des probabilités a posteriori en se basant sur une estimation par dénombrement.

Pour chaque classe phonétique, nous pouvons séparer à l'aide de la segmentation de référence, les vecteurs acoustiques de la base de données d'entraînement en deux ensembles : l'un correspondant à tous les vecteurs ayant étés émis lors de la prononciation du phonème et l'autre, plus grand, contenant tous les autres vecteurs. Par exemple, pour la classe phonétique "ay", nous obtenons l'ensemble des vecteurs $x \in X_{ay}$, et l'ensemble des vecteurs $x \in X_{\overline{ay}}$.

Quand on applique les vecteurs acoustiques x à l'entrée du réseau de neurones, les valeurs $S_{ay}(x)$ obtenues à la sortie du réseau de neurones associée à la classe phonétique "ay" peuvent aisément être quantifiées par un nombre fini de valeurs régulièrement espacés :

$$S_{ay}(x) = \frac{i}{C} \qquad ,i = 1, \ldots, C,$$

où C vaut 30 dans nos exemples.

En appliquant à l'entrée du réseau successivement tous les vecteurs acoustiques émis lors de la prononciation de phonèmes "ay", $x \in X_{ay}$, on peut dénombrer ceux produisant à la sortie du réseau un résultat quantifié identique :

$$N(X_{ay}, i)$$

Le résultat de ce dénombrement est reporté à la figure 8, ainsi que le dénombrement effectué sur l'ensemble des vecteurs acoustiques $x \in X_{\overline{ay}}$:

$$N(X_{\overline{ay}}, i)\,.$$

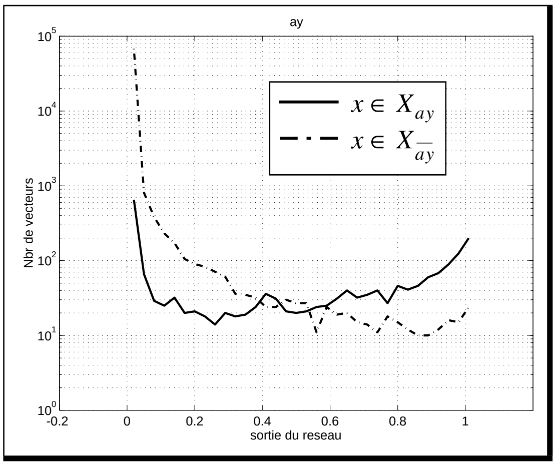

FIGURE 8. Histogramme d'utilisation de la sortie associée au phonème "ay" en fonction des vecteurs acoustiques générés lors de la prononciation ou non du phonème "ay".

En normalisant ces deux histogrammes, nous pouvons estimer la probabilité associée à la sortie selon le type de classification :

$$P(S_{ay}(x) = i \mid x \in X_{ay}) = \frac{N(X_{ay}, i)}{\sum_{i=1}^{C} N(X_{ay}, i)}$$

$$\text{et } P(S_{\overline{ay}}(x) = i \mid x \in X_{\overline{ay}}) = \frac{N(X_{\overline{ay}}, i)}{\sum_{i=1}^{C} N(X_{\overline{ay}}, i)}\,.$$

Les valeurs expérimentales sont reportées à la figure ci-dessous :

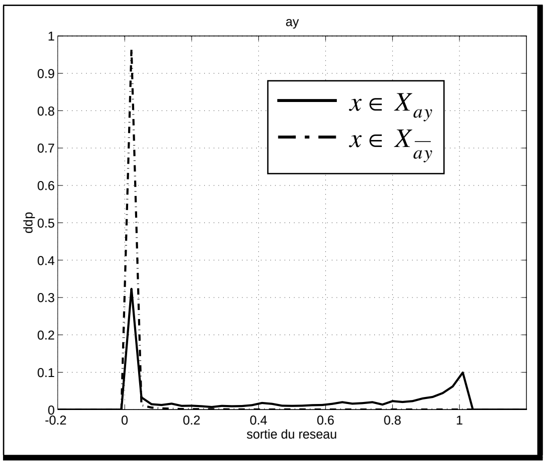

FIGURE 9. Densité de probabilité associée à la sortie du réseau de neurones, selon le type de vecteurs utilisés en entrée.

De plus, on peut également estimer les probabilités suivantes, présentée à la figure 10 :

$$P(x \in X_{ay} | S_{ay}(x) = i) = \frac{N(X_{ay}, i)}{N(X_{ay}, i) + N(X_{\overline{ay}}, i)}$$

$$\text{et } P(x \in X_{\overline{ay}} | S_{ay}(x) = i) = \frac{N(X_{\overline{ay}}, i)}{N(X_{ay}, i) + N(X_{\overline{ay}}, i)}.$$

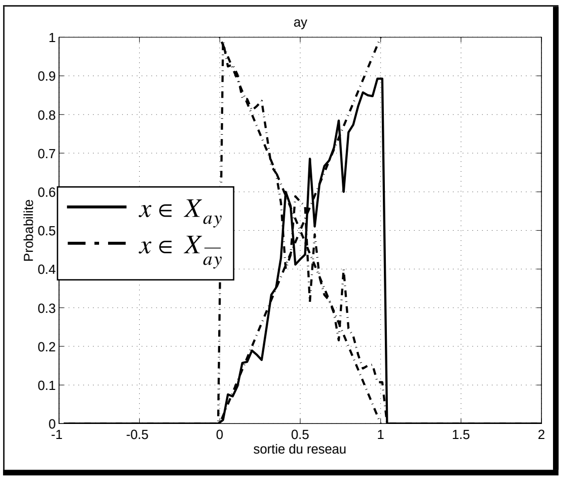

**FIGURE 10. Estimation de la probabilité de bonne classification en fonction de la sortie du réseau
(à l'aide des vecteurs de la base d'entraînement).**

Cette figure représente la probabilité qu'un vecteur acoustique x soit associé ou non à la classe "ay", connaissant la sortie du réseau de neurones $S_{ay}(x)$. On remarque que ces courbes expérimentales s'approchent fortement des courbes théoriques (en pointillés) correspondant à une transformation linéaires. Dans ce cas, nous pouvons aisément effectuer les approximations :

$$P(x \in X_{ay}) = S_{ay}(x) \text{ et } P(x \in X_{\overline{ay}}) = 1 - S_{ay}(x).$$

Notons que ces résultats sont obtenu à l'aide des vecteurs issus de la base d'entraînement.

Nous pouvons bien évidement effectuer le même raisonnement pour des vecteurs issus de la base de données de test, dans le but d'analyser la généralisation effectuée par le réseau de neurones.

Le résultat est reporté à la figure 11 :

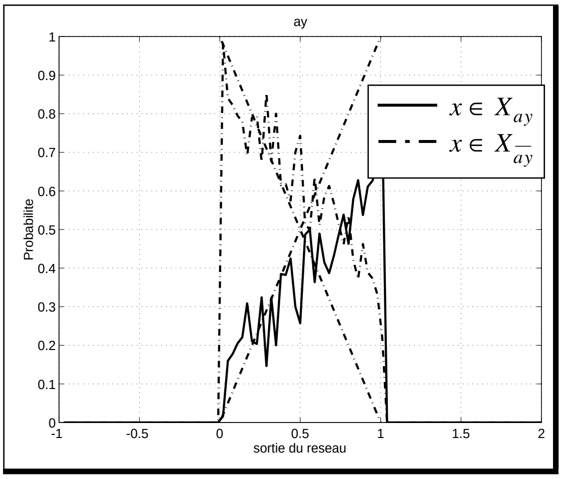

FIGURE 11. **Estimation de la probabilité de bonne classification en fonction de la sortie du réseau (à l'aide des vecteurs de la base de test).**

Nous nous apercevons, par cette figure, que l'estimation faite par le réseau de neurones reste acceptable pour les vecteurs issus de la base de données de test.

2.3 Les chaînes de Markov

2.3.1 Historique

La génération de parole est un phénomène extrêmement variable non seulement d'un locuteur à l'autre selon leurs caractéristiques physiologiques propres mais aussi pour un même locuteur dont la prononciation manque généralement de consistance. Cela se traduit non seulement par la distorsion non-linéaire de l'axe des temps mais aussi par la variabilité dans la prononciation d'un même son.

Pour résoudre le premier problème, on utilisa d'abord les méthodes des gabarits pour l'alignement temporel ("Template Matching") et on se tourna ensuite vers les *algorithmes de programmation dynamique*, ("Dynamic Time Warping", "DTW") [BELL52], permettant la mise en correspondance de deux segments de parole prononcés à des vitesses différentes en effectuant des dilatations ou compressions locales des signaux.

En prenant deux séquences acoustiques de signaux de parole différents, et en calculant les distances, dites locales, entre chaque vecteur acoustique, cet algorithme trouve l'alignement dont la somme des distances est minimale. Il permet ainsi de comparer une séquence test (un mot inconnu par exemple) avec plusieurs séquences de référence (les mots du vocabulaire) afin de déterminer quelle référence lui associer, indépendamment de la vitesse à laquelle est prononcée cette séquence.

Les programmes de reconnaissance basés sur cet algorithme ont beaucoup d'avantages. Ils sont faciles à implémenter, offrent une faible complexité de calcul et nécessitent une faible quantité de mémoire, pour autant que le nombre de signaux de référence reste raisonnable. Au delà de quelques centaines de mots de référence, la charge de calcul devient insurmontable, même en tenant compte d'un éventuel élagage ("pruning"). L'idée qui est apparue pour réduire la charge de calcul tout en permettant la reconnaissance d'un vocabulaire plus large, fut de découper ces mots en de phonèmes, de façon à diminuer le nombre de modèles de référence et d'enchaîner ces modèles pour former les mots du vocabulaire.

Le désavantage de cette approche phonétique est la mauvaise définition du phonème et sa fragilité: en effet, un phonème se présente de façon très différente selon les phonèmes voisins : c'est le *phénomène de coarticulation* qui obligerait en principe à définir chaque phonème dans son contexte gauche et droit, c'est-à-dire à définir des triphones dont le nombre élevé rend la modélisation plus difficile. Une autre difficulté est l'impossibilité de créer une base de données de phonèmes isolés.

Dans les années 80, les chercheurs se sont tournés vers une approche probabiliste de façon à modéliser aussi bien les problèmes de coarticulation entre phonèmes que les variabilités du type inter ou intra-locuteur, [BAHL75][JEL76]. Cette approche, basée sur les travaux de

Baum, [BAUM66], suppose que le phénomène modélisé (la parole) est associé à une séquence aléatoire d'états correspondant à un processus markovien inobservable, mais qui se manifeste par l'émission, elle-même aléatoire, de vecteurs acoustiques. L'introduction de ce deuxième niveau probabiliste, rendant le premier processus invisible, a conduit à appeler ces modèles des *Modèles de Markov Cachés* ("Hidden Markov Models", "HMM"). Ces deux niveaux donnent à l'approche markovienne une flexibilité nécessaire pour modéliser un phénomène aussi complexe que la production de la parole.

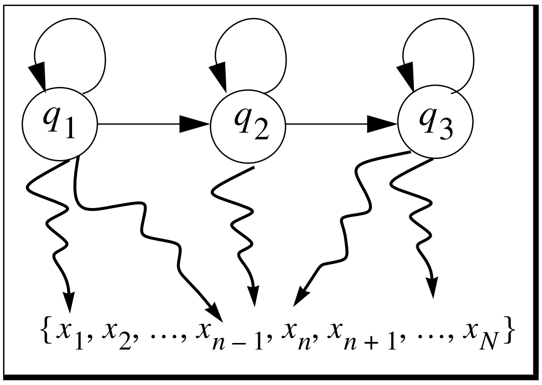

FIGURE 12. **Chaîne de Markov cachée.**

2.3.2 Modélisation

Les problèmes auxquels nous sommes confrontés sont les suivants :

Etant donnée une séquence acoustique, quelles unités de base choisir pour la modéliser : mots, syllabes, phonèmes,... En d'autres termes, que représentent les états de la chaîne, et comment ordonnance-t-on l'enchaînement de ces unités ?

Une fois cette épineuse question résolue, il reste à déterminer les paramètres de ces modèles de sorte que leur enchaînement représente au mieux la séquence acoustique correspondante, et diffère au maximum des autres.

Depuis leurs premières utilisations, [BAKER75][BAKIS76][JEL76] les chaînes de Markov ont montré qu'elles pouvaient modéliser à peu près n'importe quelle unité de parole. Le phénomène de la parole pouvant être considéré comme une évolution temporelle d'un système, les chaînes de Markov utilisées dans ce domaine ont toujours une relation unidirectionnelle vers le futur comme le montre la figure 13.

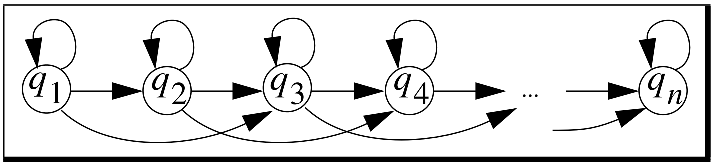

FIGURE 13. **Modèle de Bakis.**

2.3.2.1 Mots

L'unité la plus naturelle de la parole reste le mot. C'est cette unité linguistique qui fit l'objet des premières modélisations [RAB85][LIPP88]. A chaque mot, est associé un modèle dit de Bakis [BAKIS76], où les états modélisent l'évolution de la prononciation du mot au cours du temps. Le nombre d'états du modèle est fonction de la complexité de la prononciation du mot. Pourtant, il n'y a pas de relation directe entre le nombre de phonèmes composant le mot et le nombre d'états nécessaires à la modélisation. Ce nombre doit malheureusement être déterminé expérimentalement. Pour éviter ce délicat problème, les chercheurs choisissent généralement plus d'états que nécessaire et permettent au système de ne pas utiliser tous les états en insérant des sauts entre états non successifs.

2.3.2.2 Phonèmes

Comme vu précédemment, l'intérêt des chaînes de Markov réside dans leur capacité à modéliser des entités plus courtes que les mots, ce qui permet de réduire le nombre de modèles tout en augmentant la taille du vocabulaire accepté.

Le premier modèle de sous-unité linguistique qui vient à l'esprit est évidemment le phonème. La plupart des langues ont l'avantage de ne contenir que peu de phonèmes différents. L'anglais et le français n'en comportent qu'une cinquantaine, ce qui permet d'entraîner ces modèles avec peu de données.

L'approche fréquemment utilisée consiste à prendre un modèle à 3 états par phonème, en faisant l'hypothèse que l'état du milieu modélise la partie stationnaire du phonème et les états extérieurs modélisent la coarticulation avec les phonèmes voisins.

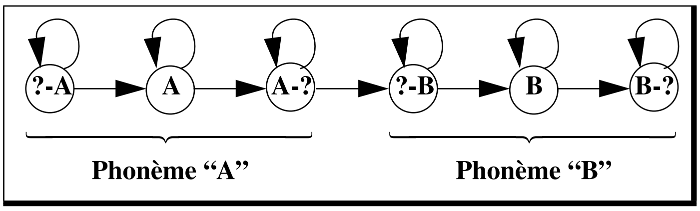

FIGURE 14. **Modèle de phonème à trois états.**

L'utilisation de diphones [ROS83][RUS81][MAR81] en reconnaissance de la parole, part du principe que les transitions entre phonèmes contiennent plus d'information que les endroits stables à l'intérieur des phonèmes. Le diphone devient alors une modélisation de la transition entre deux phonèmes avec une partie contextuelle qui correspond à ces phonèmes. Le nombre de diphones utilisés dans la langue française est de l'ordre de 1500.

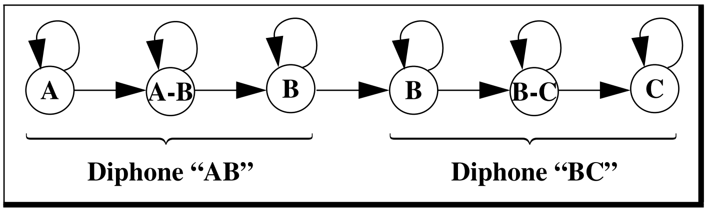

FIGURE 15. **Modèle de diphone.**

Le dernier type de modèle fréquemment utilisé est le triphone. On en compte environ 7300, dès lors ils nécessitent un entraînement fondé sur des bases de données conséquentes pour rester performant.

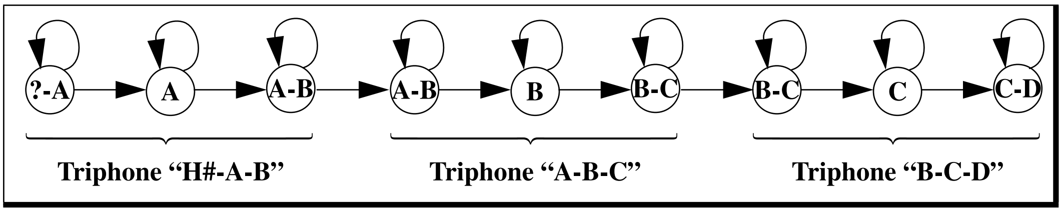

FIGURE 16. **Modèle de triphone.**

2.3.3 *Définition des paramètres*

Après avoir fixé la topologie des modèles de Markov, il faut encore déterminer les paramètres qui leur permettront de modéliser la parole.

Sachant qu'un modèle markovien correspond à un double phénomène probabiliste, on peut regrouper les paramètres du modèle en deux parties.

Les premiers paramètres, $\lambda_{\text{transition}} \in \Lambda_{\text{transition}}$, sont utilisés pour définir le processus markovien en tant que tel. Ils définissent des *probabilités de transition* entre les états du modèle : $P(q_k^{t+1} | q_j^t) \in \Lambda_{\text{transition}}$[1].

Ces probabilités de transition ont un rôle important dans la *modélisation du langage* car, suivant la construction du modèle, elles fixent les probabilités de transition entres phonèmes et par conséquent les probabilités de transitions entre les mots du vocabulaire utilisé. De plus,

1. Il est bon de rappeler ici qu'une notation rigoureuse des probabilités de transition mènerait plus exactement à $P(q_{\text{visité}}(t+1) = q_k | q_{\text{visité}}(t) = q_j)$, mais nous utiliserons la première notation qui est plus concise.

elles interviennent dans la *modélisation acoustique* car elle fixent la distribution de probabilité associée aux durées des phonèmes.

Les seconds paramètres, $\lambda_{\text{émission}} \in \Lambda_{\text{émission}}$, sont utilisés pour obtenir les *probabilités d'émission*. Pour chaque état, ils déterminent la probabilité qu'un vecteur acoustique x ait été émis par un état donné q_k : $P(x|q_k) \in \Lambda_{\text{émission}}$. On peut remarquer que ces paramètres interviennent uniquement dans la *modélisation acoustique*.

2.3.4 Critère de maximisation pour l'entraînement des modèles

Les valeurs des paramètres décrivant les modèles markoviens sont obtenus par entraînement. Lorsqu'on choisit de représenter la parole par des mots, il est possible de constituer une base de données permettant l'entraînement individuel des modèles de mots. Par contre, de telles bases ne peuvent exister pour des unités telles que les phonèmes, diphones ou triphones qui doivent être entraînés sur du texte continu (cf. section 2.5.3).

L'entraînement vise à maximiser un critère. Différents critères ont été proposés et sont examinés dans cette section.

On recherche, idéalement les valeurs de l'ensemble des paramètres, $\Lambda = \{\lambda_{\text{transition}}, \lambda_{\text{émission}}\}$, qui :

$$\text{maximisent } P(M_{\mathcal{E},i}|X_{\mathcal{E},i}, \Lambda) \quad ,\forall i, \tag{EQ 4}$$

$$\text{tout en minimisant } P(M_{\mathcal{E},j}|X_{\mathcal{E},i}, \Lambda) \quad ,\forall i, \forall j \neq i, \tag{EQ 5}$$

en supposant que ces contraintes conduisent à :

$$P(M_{\mathcal{E},i}|X_{\mathcal{E},i}, \Lambda) > P(M_{\mathcal{E},j}|X_{\mathcal{E},i}, \Lambda) \quad ,\forall i, \forall j \neq i. \tag{EQ 6}$$

Il suffit donc, pour la reconnaissance d'une séquence acoustique X, de choisir le modèle M_c ayant la probabilité $P(M_c|X, \Lambda)$ maximale.

Notons toutefois que cette approche recèle plusieurs hypothèses fortes.

- Le passage de **(EQ 6)** à la conclusion n'est assuré que si $X \subset \mathcal{E}$, autrement dit que la séquence acoustique X appartienne à la base de données d'entraînement, ce qui enlève tout intérêt au raisonnement. Il est cependant admis que si l'ensemble $\mathcal{E}$ est suffisamment grand et enregistré dans des conditions similaires à X, cette hypothèse est concevable. Cependant, la taille du “suffisamment grand” et les “conditions similaires” sont trop rarement étudiées et aboutissent souvent à des conclusions du type : “les résultats médiocres sont probablement dus à une base d’entraînement trop petite ou mal adaptée”.

- L’utilisation de **(EQ 4)** et **(EQ 5)** ne prouve nullement **(EQ 6)**, mais fournit des solutions successives qui se rapprochent d’une valeur vérifiant cette contrainte. En considérant les modèles indépendants (dont aucun état n’intervient dans la composition d’autres modèles) et les paramètres en nombre suffisant (ce qui est généralement le cas), on aboutit trivialement à **(EQ 6)**. Dans ce cas, nous perdons l’avantage des chaînes de Markov à utiliser des sous-modèles communs pour la modélisation de grands vocabulaires.

L’utilisation conjointe de **(EQ 4)** et **(EQ 5)** est rarement réalisée, car elle complique fortement l’apprentissage, et on délaisse souvent **(EQ 5)**. Cependant, il convient de noter que nombre de travaux tendent à exploiter au mieux le caractère discriminant que cette équation renferme, [ROSE92].

Le critère de maximisation se réduit donc en général à:

$$\max_{\Lambda} \prod_i P(M_{\mathcal{E},\,i} | X_{\mathcal{E},\,i}, \Lambda)\,.$$

Cependant, l’approche markovienne classique estime la probabilité $P(X_{\mathcal{E},\,i} | M_{\mathcal{E},\,i}, \Lambda)$, appelée généralement *vraisemblance*. On utilise donc la classique loi de Bayes pour en déduire $P(M_{\mathcal{E},\,i} | X_{\mathcal{E},\,i}, \Lambda)$:

$$P(M_{\mathcal{E},\,i} | X_{\mathcal{E},\,i}, \Lambda) = \frac{P(X_{\mathcal{E},\,i} | M_{\mathcal{E},\,i}, \Lambda)}{P(X_{\mathcal{E},\,i} | \Lambda)} P(M_{\mathcal{E},\,i} | \Lambda)\,. \qquad \textbf{(EQ 7)}$$

Le processus markovien suffit pour évaluer la probabilité $P(M_{\mathcal{E},\,i} | \Lambda)$, qui ne dépend donc que des paramètres $\lambda_{\text{transition}}$ décrivant le langage. C’est pour cette raison qu’on la décrit comme étant le *modèle de langage*. Le sous-groupe de $\lambda_{\text{transition}}$ concernant les transitions entre modèles est noté λ_{langage}.

A contrario le rapport $\dfrac{P(X_{\mathcal{E},\,i} | M_{\mathcal{E},\,i}, \Lambda)}{P(X_{\mathcal{E},\,i} | \Lambda)}$ est décrit comme étant le *modèle acoustique*, car il dépend principalement de $\lambda_{\text{émission}}$, mais aussi de $\lambda_{\text{transition}}$ en ce qui concerne la modélisation de la durée des phonèmes. L’ensemble des paramètres formé de $\lambda_{\text{émission}}$ et du sous-groupe de $\lambda_{\text{transition}}$ modélisant les durées des phonèmes est noté $\lambda_{\text{acoustique}}$.

Ces constatations induisent l'hypothèse que les deux modèles peuvent être maximisés séparément.

Cette remarque nous conduit à séparer le critère de maximisation   en deux autres critères :

$$\max_{\lambda_{\text{langage}}} \prod_i P(M_{\mathcal{E},i} | \lambda_{\text{langage}}) \qquad \text{(EQ 8)}$$

$$\text{et} \max_{\lambda_{\text{acoustique}}} \prod_i \frac{P(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \lambda_{\text{acoustique}})}{P(X_{\mathcal{E},i} | \lambda_{\text{acoustique}})}. \qquad \text{(EQ 9)}$$

Cette simple permutation de facteurs a des conséquences importantes au niveau de la maximisation de   vis-à-vis de **(EQ 9)**. En effet, on passe d'une *maximisation des probabilités a posteriori* à une *maximisation des vraisemblance*.

La maximisation de $\prod_i P(M_{\mathcal{E},i} | \lambda_{\text{langage}})$ se fait généralement par l'étude de la langue concernée et de l'application envisagée.

La maximisation de $\prod_i \frac{P(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \lambda_{\text{acoustique}})}{P(X_{\mathcal{E},i} | \lambda_{\text{acoustique}})}$ peut être effectuée de plusieurs manières dont les plus connues sont celle de maximisation de l'information mutuelle ("Maximum Mutual Information", "MMI") et celle de maximisation de vraisemblance ("Maximum Likelihood Estimation", "MLE").

2.3.4.1 Maximum d'information mutuelle

L'information mutuelle entre une séquence $X_{\mathcal{E},i}$ et un modèle $M_{\mathcal{E},i}$ est définie par :

$$I(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \Lambda) = \log \frac{P(X_{\mathcal{E},i}, M_{\mathcal{E},i} | \Lambda)}{P(X_{\mathcal{E},i} | \Lambda) P(M_{\mathcal{E},i} | \Lambda)}. \qquad \text{(EQ 10)}$$

En utilisant le fait que

$$P(X_{\mathcal{E},i}, M_{\mathcal{E},i} | \Lambda) = P(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \Lambda) P(M_{\mathcal{E},i} | \Lambda),$$

on obtient

$$I(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \Lambda) = \log \frac{P(X_{\mathcal{E},i} | M_{\mathcal{E},i}, \Lambda)}{P(X_{\mathcal{E},i} | \Lambda)}, \qquad \text{(EQ 11)}$$

dont la minimisation est identique à celle de (EQ 9).

Or, sachant que

$$P(X_{\mathcal{E},i}|\Lambda) = \sum_j P(X_{\mathcal{E},i}, M_{\mathcal{E},j}|\Lambda) = \sum_j P(X_{\mathcal{E},i}|M_{\mathcal{E},j}, \Lambda)P(M_{\mathcal{E},j}|\Lambda),$$

on obtient :

$$I(X_{\mathcal{E},i}|M_{\mathcal{E},i}, \Lambda) = log\frac{P(X_{\mathcal{E},i}|M_{\mathcal{E},i}, \Lambda)}{\sum_j P(X_{\mathcal{E},i}|M_{\mathcal{E},j}, \Lambda)P(M_{\mathcal{E},j}|\Lambda)}.$$

Le critère de maximisation utilisé pour l'entraînement des modèles se résume à :

$$\max_{\Lambda}\sum_i I(X_{\mathcal{E},i}|M_{\mathcal{E},i}, \Lambda).$$

Cependant, l'obtention de ces valeurs est loin d'être triviale. On a le plus souvent recours à des procédures d'optimisation générale telles que les algorithmes du gradient, [BAHL86][OKAN93][KAPA93].

Une solution à ce problème, [MERI88][NORM94], est d'approximer le dénominateur par un modèle générique M_g permettant toutes les transitions possibles du langage :

$$P(X_{\mathcal{E},i}|M_g) = \sum_j P(X_{\mathcal{E},i}|M_{\mathcal{E},j}, \Lambda)P(M_{\mathcal{E},j}|\Lambda).$$

2.3.4.2 Maximum de vraisemblance

Cette approche est la plus souvent utilisée car elle conduit à des algorithmes d'entraînement beaucoup plus simples.

Elle consiste à simplifier l'équation (EQ 9) en considérant que $P(X_{\mathcal{E},i}|\lambda_{\text{acoustique}})$ est constant, non seulement lors de la reconnaissance, mais aussi lors de l'entraînement. Le critère de maximisation peut dès lors se simplifier en :

$$\max_{\lambda_{\text{acoustique}}}\prod_i P(X_{\mathcal{E},i}|M_{\mathcal{E},i}, \lambda_{\text{acoustique}}), \quad \text{(EQ 12)}$$

qui définit, avec (EQ 8), le critère de maximum de vraisemblance.

2.4 Algorithme de Baum-Welch

Il n'existe aucune approche analytique permettant d'obtenir l'ensemble des paramètres Λ_{opt} assurant un maximum global de **(EQ 7)**. Cependant, Baum et ses collègues, [BAUM66] [BAUM70] [BAUM72] travaillèrent, à la fin des années 60, à l'obtention d'un algorithme itératif conduisant à un maximum local. Cet algorithme, nommé *réestimation avant-arrière* ("Forward Backward Reestimation") ou plus simplement *réestimation de Baum-Welch* est la méthode la plus couramment utilisée. D'autres méthodes, basées sur les optimisations par gradient ont également été utilisées, [LEV83].

Le développement mathématique des preuves de convergence vers un maximum local de l'algorithme de Baum-Welch, ainsi que la dérivation des formules de réestimation des paramètres, sont reportés à l'annexe A. Seules les formules de réestimation sont reproduites ici.

Dans cette section, on montre comment l'algorithme de Baum-Welch estime $P(X|M, \lambda_{\text{acoustique}})$ en partant de $P(x|q_k, \lambda_{\text{acoustique}})$ et de $P(q_k^{t+1}\middle|q_j^t, \lambda_{\text{acoustique}})$.

On allégera l'écriture en omettant systématiquement $\lambda_{\text{acoustique}}$ et en notant N la taille de la séquence acoustique X.

Les états q_i de $Q = \{q_1, \ldots, q_K\}$ étant disjoints, tous les q_i^t (visite de l'état q_i à l'instant t) sont des évènements mutuellement exclusifs et on peut écrire :

$$P(X|M) = \sum_{i=1}^{K} P(q_i^t, X\middle|M) \quad , \quad \forall t = 1, \ldots, N. \qquad \textbf{(EQ 13)}$$

Le terme de gauche peut aisément être développé autour du temps t en séparant les informations relatives aux temps précédents t (t compris), vis-à-vis des informations relatives aux temps suivant :

$$P(q_i^t, X\middle|M) = P(q_i^t, X_1^t\middle|M)P(X_{t+1}^N\middle|q_i^t, X_1^t, M). \qquad \textbf{(EQ 14)}$$

Le premier facteur, noté α_i^t, est appelé *probabilité avant* et correspond à la probabilité d'avoir observé le début de la séquence de vecteurs acoustiques X depuis le moment initial et jusqu'à l'instant t, et d'être arrivé à ce même instant t sur l'état q_i.

Le second facteur, noté β_i^t, est appelé *probabilité arrière* et correspond à la probabilité d'observer la fin de la séquence de vecteurs acoustiques X entre l'instant $t+1$ et le temps N correspondant à la durée totale de la séquence, sachant que l'on avait préalablement observé le début de la séquence acoustique X_1^t et que l'on était arrivé sur l'état q_i à l'instant t.

La récriture de l'équation **(EQ 13)** donne :

$$\boxed{P(X|M) = \sum_{i=1}^{K} \alpha_i^t \beta_i^t \quad , \quad \forall t = 1, \ldots, N} \tag{EQ 15}$$

On peut également étendre la définition de **(EQ 15)** pour $t = 0$ si l'on impose le premier état comme étant l'état initial : $P(q_i^0) = \delta_{i,1}$. Ceci entraîne :

$$\alpha_i^0 = \delta_{i,1}, \ \beta_i^0 = P(X_1^N | M) \text{ et}$$

$$P(X|M) = \beta_1^0. \tag{EQ 16}$$

En imposant l'état q_K le dernier état visité, $P(q_i^N) = \delta_{i,K}$, on a dans l'autre cas extrême, $t = N$:

$$\alpha_i^N = P(\delta_{i,K}, X | M), \ \beta_i^N = 1 \text{ et}$$

$$P(X|M) = \alpha_K^N. \tag{EQ 17}$$

2.4.0.1 Récurrence avant

On peut aisément déduire de la probabilité avant, une formule de récurrence :

$$\begin{aligned} P(q_i^t, X_1^t | M) &= \sum_{k=1}^{K} P(q_i^t, q_k^{t-1}, x_t, X_1^{t-1} | M) \\ &= \sum_{k=1}^{K} P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M) P(q_k^{t-1}, X_1^{t-1} | M), \end{aligned}$$

où le facteur $P(q_k^{t-1}, X_1^{t-1} | M)$ conduit à la récurrence, et le facteur $P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M)$ est défini comme la *contribution locale* et peut être séparé suivant :

$$P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M) = P(q_i^t | q_k^{t-1}, X_1^{t-1}, M) P(x_t | q_i^t, q_k^{t-1}, X_1^{t-1}, M) \tag{EQ 18}$$

où le premier facteur correspond à la *probabilité de transition* et le second en *probabilité d'émission.*

En faisant les hypothèses successives :

- le modèle de Markov est du premier ordre :

  $P(q_i^t | q_k^{t-1}, X_1^{t-1}, M) = P(q_i^t | q_k^{t-1}, M)$;
- les vecteurs acoustiques sont indépendants les uns des autres :

  $P(x_t | q_i^t, q_k^{t-1}, X_1^{t-1}, M) = P(x_t | q_i^t, q_k^{t-1}, M)$;
- les vecteurs acoustiques étant émis par les états visités au temps t, sont indépendants des états visités précédemment :

  $P(x_t | q_i^t, q_k^{t-1}, M) = P(x_t | q_i^t, M)$.

On trouve finalement :

$$P(q_i^t, X_1^t | M) = \sum_{k=1}^{K} P(q_i^t | q_k^{t-1}, M) P(x_t | q_i^t, M) P(q_k^{t-1}, X_1^{t-1} | M),$$

que l'on peut réécrire sous la forme :

$$\alpha_i^t = P(x_t | q_i^t, M) \sum_{k=1}^{K} P(q_i^t | q_k^{t-1}, M) \alpha_k^{t-1}. \tag{EQ 19}$$

2.4.0.2 Récurrence arrière

On peut également obtenir une formule de récurrence pour la probabilité arrière :

$$
\begin{aligned}
P(X_{t+1}^N \mid q_i^t, X_1^t, M) &= \sum_{k=1}^{K} P(q_k^{t+1}, x_{t+1}, X_{t+2}^N \mid q_i^t, X_1^t, M) \\
&= \sum_{k=1}^{K} P(q_k^{t+1}, x_{t+1} \mid q_i^t, X_1^t, M) P(X_{t+2}^N \mid q_k^{t+1}, q_i^t, X_1^{t+1}, M) \\
&= \sum_{k=1}^{K} P(q_k^{t+1} \mid q_i^t, X_1^{t+1}, M) \\
&\quad P(x_{t+1} \mid q_i^t, q_k^{t+1}, X_1^t, M) P(X_{t+2}^N \mid q_k^{t+1}, q_i^t, X_1^{t+1}, M),
\end{aligned}
$$

en utilisant successivement les mêmes hypothèses :

- le modèle de Markov est du premier ordre :

  $P(q_k^{t+1} \mid q_i^t, X_1^{t+1}, M) = P(q_k^{t+1} \mid q_i^t, M)$;
- les vecteurs acoustiques sont indépendants les uns des autres :

  $P(x_{t+1} \mid q_i^t, q_k^{t+1}, X_1^t, M) = P(x_{t+1} \mid q_i^t, q_k^{t+1}, M)$;
- les vecteurs acoustiques étant émis par les états visités au temps t, sont indépendants des états visités précédemment :

  $P(x_{t+1} \mid q_i^t, q_k^{t+1}, M) = P(x_{t+1} \mid q_k^{t+1}, M)$, et

  $P(X_{t+2}^N \mid q_k^{t+1}, q_i^t, X_1^{t+1}, M) = P(X_{t+2}^N \mid q_k^{t+1}, X_1^{t+1}, M)$.

On trouve finalement :

$$
P(X_{t+1}^N \mid q_i^t, X_1^t, M) = \sum_{k=1}^{K} P(q_k^{t+1} \mid q_i^t, M) P(x_{t+1} \mid q_k^{t+1}, M) P(X_{t+2}^N \mid q_k^{t+1}, X_1^{t+1}, M),
$$

que l'on peut également réécrire sous la forme :

$$
\boxed{\beta_i^t = \sum_{k=1}^{K} P(q_k^{t+1} \mid q_i^t, M) P(x_{t+1} \mid q_k^{t+1}, M) \beta_k^{t+1}} \qquad \textbf{(EQ 20)}
$$

2.4.1 Estimation des paramètres

Etant donné l'ensemble $\mathcal{E} = \{X_{\mathcal{E},1}, \ldots, X_{\mathcal{E},N_{\mathcal{E}}}\}$ des séquences acoustiques associées à la base de données d'entraînement, et l'ensemble $\mathcal{M} = \{M_{\mathcal{E},1}, \ldots, M_{\mathcal{E},N_{\mathcal{E}}}\}$ des modèles associés à ces séquences acoustiques, l'entraînement consiste à déterminer l'ensemble des paramètres Λ qui maximise les probabilités $P(X_{E,i}|M_{E,i}, \Lambda)$.

En considérant M la concaténation de tous les modèles $M_{\mathcal{E},i}$ et X la concaténation de toutes les séquence acoustiques $X_{\mathcal{E},i}$, l'entraînement revient à déterminer Λ qui maximise $P(X|M, \Lambda)$.

L'algorithme de Baum-Welch est un processus itératif où l'on estime à chaque itération de nouvelles valeurs des paramètres Λ à partir des anciennes valeurs Λ', de façon à assurer la relation $P(X|M, \Lambda) \geq P(X|M, \Lambda')$. Le processus itératif est assuré de converger vers un maximum local de $P(X|M, \Lambda)$ vu que cette probabilité est bornée.

Les paramètres $\lambda_i \in \Lambda$ sont l'ensemble

- des probabilités de transition,

$$\lambda_{tr(i,k)} = P(q_k|\bar{q_i}, M) \qquad ,\forall i = 1, \ldots, K \text{ et } \forall k = 1, \ldots, K,$$

où $P(q_k|\bar{q_i}, M)$ correspond à $P(q_k^t|q_i^{t-1}, M)$ supposée constante $\forall t$,

- et des paramètres associés aux probabilités d'émission.

Dans le cas où l'on suppose chaque état associé à une densité de probabilité gaussienne d'ordre N_a dont la matrice de covariance est supposée diagonale, la probabilité d'émission peut s'écrire sous la forme :

$$p(x|q_k) = \prod_{n=1}^{N_a} \frac{1}{\sigma_{k,n}\sqrt{2\pi}} e^{-\frac{(x_n - \mu_{k,n})^2}{2\sigma_{k,n}^2}}, \qquad \text{(EQ 21)}$$

où les paramètres sont :

$$\left.\begin{array}{l}\mu_{k,n}\\ \sigma_{k,n}\end{array}\right\} \quad ,\forall k = 1, \dots, K \text{ et } \forall n = 1, \dots, N_a . \qquad \textbf{(EQ 22)}$$

Dans ce cas, Les formules de réestimation extraites de l'annexe A sont respectivement :

$$\lambda_{tr(i,k)} = \frac{\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t | M, \Lambda')}{\sum_{t=1}^{T} P(X, q_i^{t-1} | M, \Lambda')},$$

$$\mu_{k,n} = \frac{\sum_{t=1}^{T} x_n^t P(X, q_k^t | M, \Lambda')}{\sum_{t=1}^{T} P(X, q_k^t | M, \Lambda')} \text{ et}$$

$$\sigma_{k,n} = \sqrt{\frac{\sum_{t=1}^{T} (x_n^t - \mu_{k,n})^2 P(X, q_k^t | M, \Lambda')}{\sum_{t=1}^{T} P(X, q_k^t | M, \Lambda')}}.$$

Exprimons maintenant les paramètres en fonction des probabilités avant et arrière.

Par définition de α_k^t et β_k^t on trouve directement $P(X, q_k^t | M, \Lambda') = \alpha_k^t \beta_k^t$.

Pour $P(X, q_i^{t-1}, q_j^t | M, \Lambda')$, on trouve successivement :

$$\begin{aligned}
P(X, q_i^{t-1}, q_j^t | M, \Lambda') &= P(X_1^{t-1}, x_t, X_{t+1}^N, q_i^{t-1}, q_j^t | M, \Lambda') \\
&= P(X_1^{t-1}, q_i^{t-1} | M, \Lambda') P(x_t, X_{t+1}^N, q_j^t | M, q_i^{t-1}, X_1^{t-1}, \Lambda') \\
&= \alpha_i^{t-1} P(q_j^t | M, q_i^{t-1}, X_1^{t-1}, \Lambda') P(x_t, X_{t+1}^N | M, q_i^{t-1}, q_j^t, X_1^{t-1}, \Lambda') \\
&= \alpha_i^{t-1} \lambda'_{tr(i,j)} P(x_t | M, q_i^{t-1}, q_j^t, X_1^{t-1}, \Lambda') P(X_{t+1}^N | M, q_i^{t-1}, q_j^t, X_1^t, \Lambda') \\
&= \alpha_i^{t-1} \lambda'_{tr(i,j)} P(x_t | q_j^t, M, \Lambda') \beta_j^t
\end{aligned}$$

en prenant les hypothèses habituelles.

Les formules peuvent donc se réduire à :

$$\lambda_{tr(i,k)} = \frac{\sum_{t=1}^{T} \alpha_i^{t-1} \lambda'_{tr(i,k)} P(x_t | q_j^t, M, \Lambda') \beta_j^t}{\sum_{t=1}^{T} \alpha_i^{t-1} \beta_i^{t-1}}, \qquad \text{(EQ 23)}$$

$$\mu_{k,n} = \frac{\sum_{t=1}^{T} x_n^t \alpha_k^t \beta_k^t}{\sum_{t=1}^{T} \alpha_k^t \beta_k^t} \quad \text{et}$$

$$\sigma_{k,n} = \sqrt{\frac{\sum_{t=1}^{T} (x_n^t - \mu_{k,n})^2 \alpha_k^t \beta_k^t}{\sum_{t=1}^{T} \alpha_k^t \beta_k^t}}$$

2.4.2 *Réseaux de neurones et algorithme de Baum-Welch*

L'utilisation simultanée de chaînes de Markov pour la modélisation du langage et des réseaux de neurones pour la modélisation acoustique est un sujet qui fût introduit à la fin des années 80, [BOU88].

Comme expliqué dans la section 2.2.5, les réseaux de neurones peuvent être utilisés pour estimer la probabilité $P(\varphi_i^t | x_t, \Lambda)$ qu'un phonème φ_i soit prononcé à l'instant t, sachant que l'on observe le vecteur acoustique x_t. Pour ce faire il suffit, comme le montre la figure 17, d'appliquer à l'entrée d'un réseau préalablement entraîné, le vecteur acoustique, pour obtenir cette probabilité sur les différentes sorties, chacune représentant un phonème précis. Dans ce cas, nous considérons maintenant un seul état markovien par phonème.

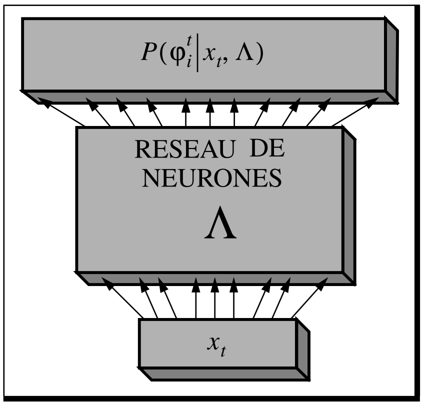

FIGURE 17. **Réseau de neurones et probabilité a posteriori**

Cependant, dans ses formules de récurrence, l'algorithme de Baum-Welch utilise les vraisemblances utilisées comme probabilités d'émission dans les chaînes de Markov. Ces vraisemblances sont calculées à partir des probabilités a posteriori grâce à la loi de Bayes :

$$p(x|\varphi_i) = \frac{P(\varphi_i|x)p(x)}{P(\varphi_i)}, \qquad \text{(EQ 24)}$$

où $P(\varphi_i|x)$ est la probabilité estimée à l'aide du réseau de neurones,

$p(x|\varphi_i)$, l'estimation de la vraisemblance utilisée par l'algorithme de Baum-Welch,

$P(\varphi_i)$, la probabilité a priori d'observer le phonème φ_i (estimée par dénombrement de la base de données),

$p(x)$, la probabilité a priori d'émettre le vecteur acoustique x. Lors de l'entraînement et de son utilisation, les influences de cette probabilité s'annulent.

2.4.2.1 Entraînement emboîté

L'entraînement d'un système de reconnaissance basé sur les deux méthodes nécessite quelques précautions. En effet les paramètres à estimer sont de deux types. Les uns sont les probabilités de transition entres états des modèles markoviens tandis que les autres sont les poids du réseau de neurones.

L'estimation des paramètres s'effectue donc de manière alternée. Comme le montre la figure 18, partant d'un réseau de neurone initiale (ou de tout autre modèle permettant l'estimation des probabilités d'émission), et d'une première estimation des probabilités de transitions entre phonèmes, l'utilisation de l'algorithme de Baum-Welch, permet d'obtenir une meilleure

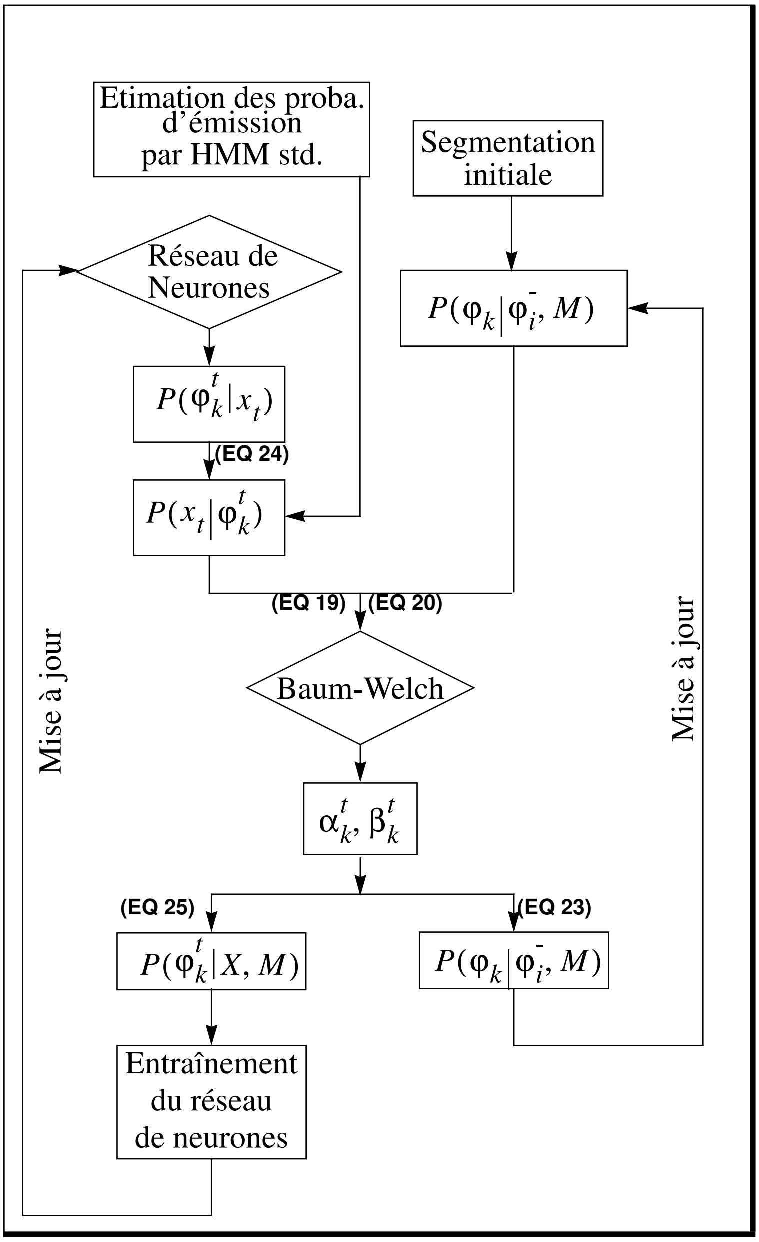

FIGURE 18. **Schéma bloc d'entraînement imbriqué pour l'algorithme de Baum-Welch.**

estimation des probabilités de transition ainsi que de nouveaux objectifs pour l'entraînement du réseau de neurones.

En effet, en reprenant (EQ 14) et (EQ 15), on trouve aisément :

$$P(\varphi_k^t|X, M) = \frac{P(\varphi_k^t, X|M)}{P(X|M)} = \frac{\alpha_k^t \beta_k^t}{\sum_{i=1}^{K} \alpha_i^t \beta_i^t}. \qquad \text{(EQ 25)}$$

Cette probabilité globale peut être utilisée comme objectif pour l'entraînement d'un nouveau réseau de neurones. Cette méthode assure la maximisation de la probabilité étant donné que chacun des deux processus d'optimisation fait croître la probabilité globale.

Cependant, le processus d'apprentissage est lourd, et comme dans le cas classique (estimation des probabilités d'émissions par distributions gaussiennes) on préfère généralement utiliser une simplification basée sur l'algorithme de Viterbi.

2.5 Algorithme de Viterbi

L'algorithme de Viterbi constitue une simplification de l'algorithme de Baum-Welch. La vraisemblance $P(X|M)$ est estimée non plus à l'aide de l'ensemble de tous les chemins possibles parcourant le modèle, mais uniquement à l'aide du meilleur chemin :

$$\tilde{P}(X|M) = \max_{\gamma \in \Gamma} P(X, \gamma|M) . \qquad \textbf{(EQ 26)}$$

Notons $\gamma^t_{max, i}$ le meilleur chemin de longueur t, associé à la séquence acoustique X^t_1 et finissant à l'état q_i.

En définissant :

$$\nu^t_i = P(X^t_1, \gamma^t_{max, i} \middle| M) = \max_j P(X^t_1, \gamma^{t-1}_{max, j}, q^t_i \middle| M) ,$$

on a trivialement :

$$\tilde{P}(X|M) = \nu^N_L , \qquad \textbf{(EQ 27)}$$

où l'on considère ici q_L le dernier état de la chaîne de Markov.

De même, on obtient aisément une *formule de récurrence* :

$$\begin{aligned}
\nu^t_i &= \max_k P(\gamma^{t-1}_{max, k}, q^t_i, X^t_1 \middle| M) \\
&= \max_k P(q^t_i, \gamma^{t-1}_{max, k}, x_t, X^{t-1}_1 \middle| M) \\
&= \max_k P(q^t_i, x_t \middle| \gamma^{t-1}_{max, k}, X^{t-1}_1, M) P(\gamma^{t-1}_{max, k}, X^{t-1}_1 \middle| M) \\
&= \max_k P(q^t_i \middle| \gamma^{t-1}_{max, k}, X^{t-1}_1, M) P\left(x_t \middle| q^t_i, \gamma^{t-1}_{max, k}, X^{t-1}_1, M\right) \nu^{t-1}_k
\end{aligned}$$

en effectuant les hypothèses :

- le modèle de Markov est du premier ordre :
  $P(q^{t+1}_k \middle| \gamma^{t-1}_{max, k}, X^{t+1}_1, M) = P(q^{t+1}_k \middle| q^t_i, M)$;
- les vecteurs acoustiques sont indépendants les uns des autres :

$$P\left(x_t \middle| q_i^t, \gamma_{max,k}^{t-1}, X_1^{t-1}, M\right) = P\left(x_t \middle| q_i^t, \gamma_{max,k}^{t-1}, M\right);$$

- les vecteurs acoustiques étant émis par les états visités au temps t, sont indépendants des états visités précédemment :

$$P\left(x_t \middle| q_i^t, \gamma_{max,k}^{t-1}, M\right) = P(x_t | q_i^t, M).$$

On a finalement :

$$\boxed{\nu_i^t = P(x_t | q_i^t, M) \max_k (\nu_k^{t-1} \quad P(q_i^t | q_k^{t-1}, M))} \qquad \textbf{(EQ 28)}$$

2.5.1 Estimation des paramètres

L'algorithme de Viterbi ne se basant que sur le meilleur chemin, les formules de réestimation des paramètres deviennent triviales.

Si l'on note γ_m, le meilleur chemin conduisant à l'estimation de $P(X|M)$, on peut écrire :

$\tilde{P}(X|M) = P(X, \gamma_m | M)$, et directement :

$$\begin{aligned} \tilde{P}(X, q_i^{t-1}, q_j^t | M, \Lambda) &= P(X, \gamma_m, q_i^{t-1}, q_j^t | M) \\ &= P(X|M)\delta_{\gamma_m, i, j}^t \end{aligned}$$

et

$$\begin{aligned} \tilde{P}(X, q_i^t | M, \Lambda) &= P(X, \gamma_m, q_i^t | M) \\ &= P(X|M)\delta_{\gamma_m, i}^t, \end{aligned}$$

$$\text{où} \begin{cases} \delta^t_{\gamma_m, i, j} = 1 \ \text{si } \gamma_m \text{ contient les évènements } q_i^{t-1} \text{ et } q_j^t \\ \qquad\qquad = 0 \ \text{sinon} \end{cases}$$

$$\text{et} \begin{cases} \delta^t_{\gamma_m, i} = 1 \ \text{si} \gamma_m \text{contient } q_i^t \\ \qquad = 0 \ \text{sinon} \end{cases} .$$

En partant des formules de réévaluation de Baum-Welch, on montre :

$$\lambda_{tr(i,k)} = \frac{\sum_{t=1}^{T} P(X|M)\delta^t_{\gamma_m, i, j}}{\sum_{t=1}^{T} P(X|M)\delta^t_{\gamma_m, i}} = \frac{\sum_{t=1}^{T} \delta^t_{\gamma_m, i, j}}{\sum_{t=1}^{T} \delta^t_{\gamma_m, i}} = \frac{\text{nombre de transitions entre } q_i \text{ et } q_j}{\text{nombre de transitions à partir de } q_i}, \quad \text{(EQ 29)}$$

$$\mu_{k,n} = \frac{\sum_{t=1}^{T} x_n^t P(X, q_k^t \middle| M, \Lambda')}{\sum_{t=1}^{T} P(X, q_k^t \middle| M, \Lambda')} = \frac{\sum_{t=1}^{T} x_n^t \delta^t_{\gamma_m, i}}{\sum_{t=1}^{T} \delta^t_{\gamma_m, i}} = \text{moyenne des } x_n \text{émis sur } q_k \text{ et}$$

$$\sigma_{k,n} = \sqrt{\frac{\sum_{t=1}^{T} (x_n^t - \mu_{k,n})^2 \delta^t_{\gamma_m, i}}{\sum_{t=1}^{T} \delta^t_{\gamma_m, i}}} = \sqrt{\text{moyenne des } (x_n^t - \mu_{k,n})^2 \text{ où } x_n \text{ est émis sur } q_k} .$$

2.5.2 Réseaux de neurones et algorithme de Viterbi

L'entraînement d'un système fondé conjointement sur l'algorithme de Viterbi et sur l'utilisation d'un réseau de neurones est similaire à celui décrit dans la section 2.4.2.

Cependant, l'algorithme de Viterbi nous procure ici uniquement le meilleur chemin.

Hors de ce chemin optimal, nous pouvons extraire aisément une estimation des probabilités de transition en effectuant un simple dénombrement.

De plus, ce chemin optimal associe chaque vecteur acoustique à un phonème particulier.

Il est montré, [BOU88], que l'utilisation de cette nouvelle classification pour l'entraînement d'un nouveau réseau entraînait une augmentation des probabilités a posteriori $P(M|X, \Lambda)$.

L' objectif utilisé lors de l'entraînement consistent simplement en un vecteur index dont toutes les composantes sont nulles à l'exception de celle désignant le phonème associé au vecteur acoustique placé à l'entrée du réseau.

La figure 19 reprend de manière synthétique le processus utilisé pour l'estimation successives des paramètres.

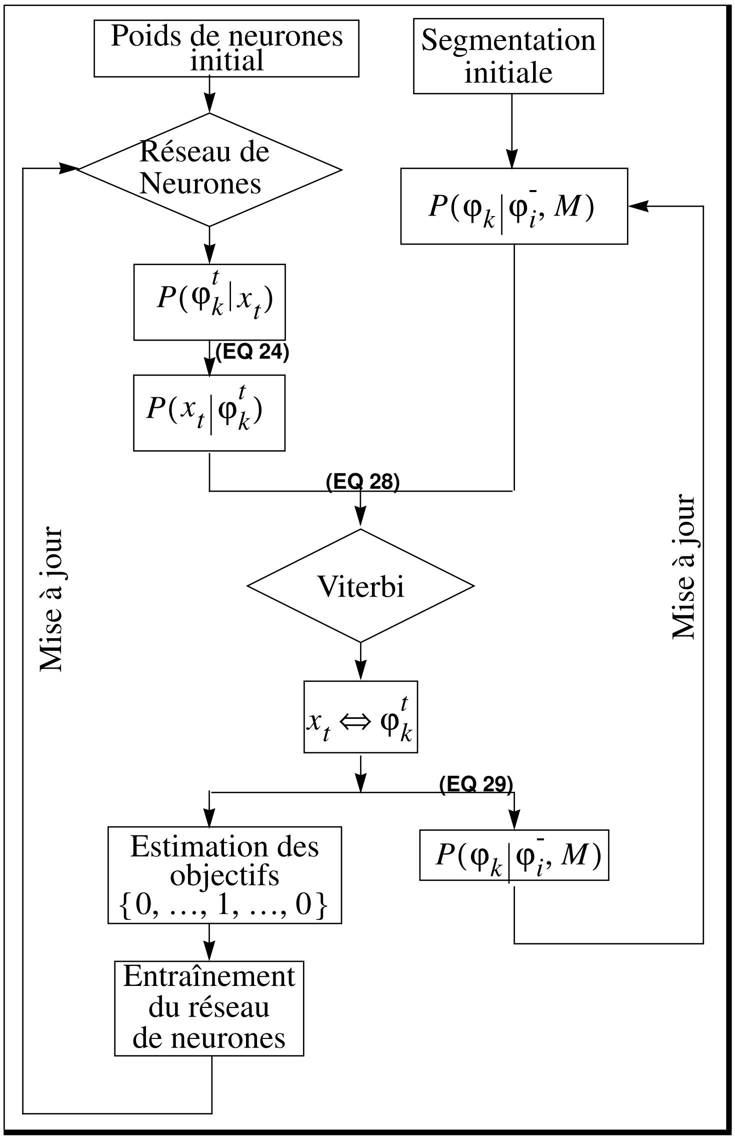

FIGURE 19. **Schéma bloc d'entraînement imbriqué pour l'algorithme de Viterbi.**

2.5.3 *Parcours rapide*

Comme nous venons de le voir, l'algorithme de Viterbi peut être utilisé pour estimer les paramètres du modèle markovien. Comme le montre la figure 20, moyennant une modification triviale de la définition des prédécesseurs d'un état, il peut également être utilisé pour rechercher la segmentation optimale d'une séquence acoustique. En effet l'algorithme de rétropropagation nous fournit la séquence d'états conduisant à la vraisemblance maximale, $P(X|M)$.

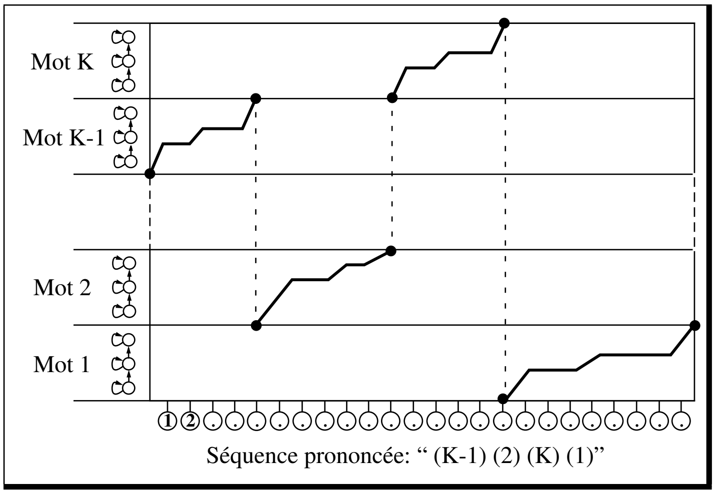

FIGURE 20. **Reconnaissance de mots enchaînés.**

Cependant, si l'objectif de l'utilisation de l'algorithme est la segmentation (en mots ou en phonèmes), il est sans intérêt de conserver le détail du chemin entre les états représentant les unités à segmenter.

D'un point de vue algorithmique, il est alors possible de minimiser l'espace mémoire nécessaire à une telle recherche. H. Ney, [NEY84], montre qu'il suffit à chaque instant de conserver le point d'entrée du dernier phonème correspondant au meilleur chemin. Pour ce faire (voir figure 21), il suffit, outre le vecteur vertical nécessaire pour le calcul des probabilités totales, d'un vecteur vertical supplémentaire conservant le point d'entrée du dernier phonème de chaque chemin, ainsi que de deux vecteurs horizontaux contenant, à chaque instant t l'indice du dernier mot correspondant au meilleur chemin (ici, '2 ') et l'indice temporel correspondant au début de ce mot (dans ce cas, 'n'). En pratique il est cependant nécessaire de doubler les vecteurs verticaux lors de la mise à jour des valeurs, si la grammaire entre les mots n'est pas triviale (dans le cas d'un simple alignement temporel où la séquence de mots est fixée, un ordre précis de mise à jour des valeurs permet d'éviter ce doublement).

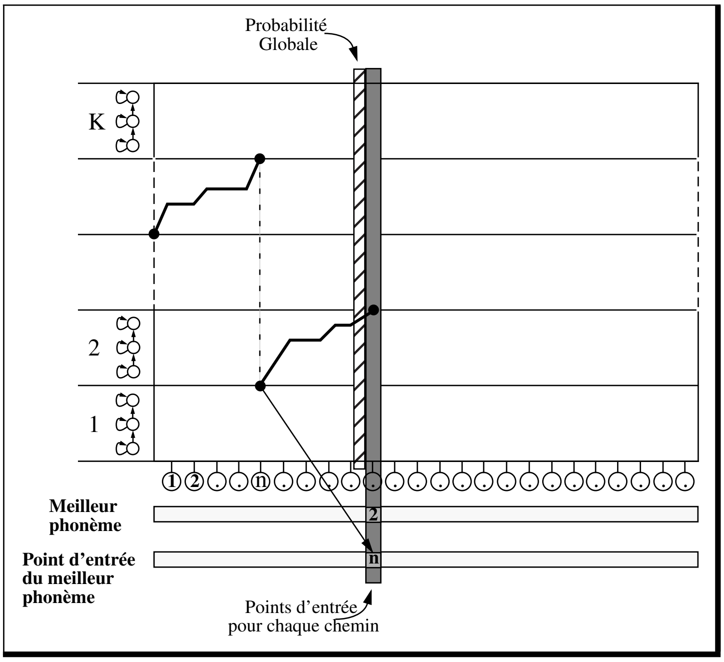

FIGURE 21. Récurrence rapide pour l'algorithme de Viterbi

2.5.4 *Parcours rapide pour les N meilleurs chemins*

Une extension triviale de l'algorithme de récurrence rapide peut être envisagée pour la recherche des N meilleures segmentations (voir, par exemple [STEIN91]). Le principe de base est de conserver à chaque étape de l'algorithme, non plus le meilleur choix, mais les N meilleurs. Pour ce faire, il est nécessaire de multiplier par N tous les vecteurs utilisés pour conserver l'information utile.

Comme le montre la figure 22, à chaque instant, et pour chaque état, nous conservons maintenant les N meilleurs chemins finissant sur cet état. Pour ce faire, parmi les $N(K+1)$ chemins possibles à chaque instant (en tenant compte du chemin finissant sur l'état en question), nous n'en conservons que les N meilleurs.

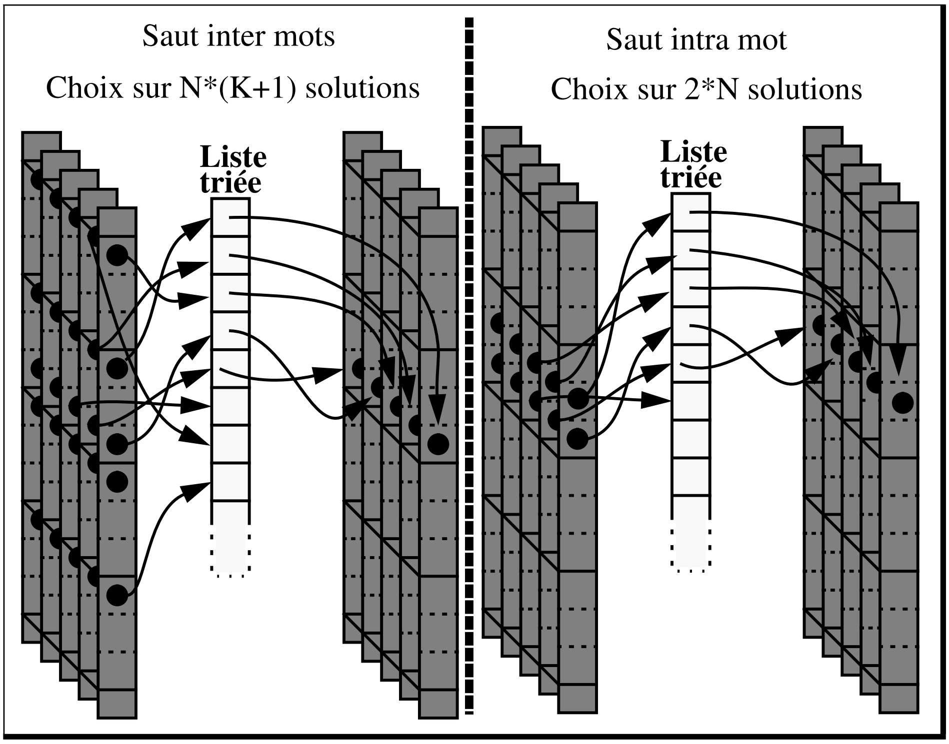

FIGURE 22. **Décision locale pour le N-Best strict.**

De même manière que lors de l'algorithme de Viterbi, il n'est pas nécessaire de conserver toutes les décisions locales, car seules les transitions entre les K mots, ainsi que leurs points d'entrée respectifs sont intéressants. Comme le montre la figure 23, il faut néanmoins conserver les N sauts possibles à chaque instant, ce qui implique l'utilisation de $2N$ vecteurs horizontaux pour retenir l'indice du mot choisi et son point d'entrée.

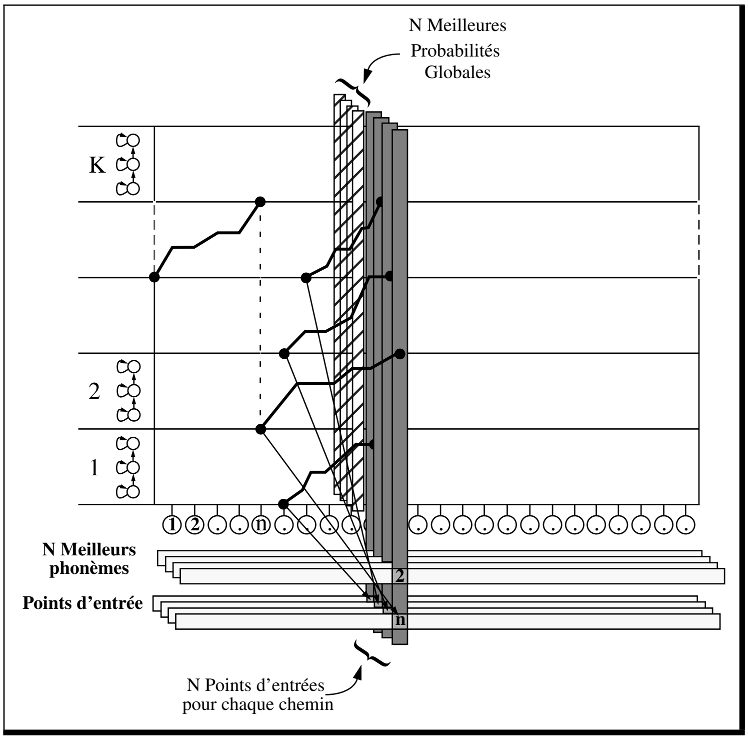

FIGURE 23. **Fast N-Best**

2.5.5 Parcours rapide pour l'approximation des N meilleurs chemins.

Le défaut intrinsèque de l'algorithme du N-Best parfait, est qu'il génère habituellement des chemins contenant de faibles variations. En effet, il se peut qu'il retourne des chemins correspondant à une séquence identique de mots où seules existent des variations dans les sauts internes entre états.

Pour pallier ce défaut, nombre de méthodes ont été envisagée[SOON91][STEIN91]. Présentons les plus courantes dans l'ordre croissant des hypothèses simplificatrices lors de leurs progressions avant.

La première, ("Exact sentence-dependent N-Best") [SCHW91], consiste à effectuer des modification dans le choix entre les $N(K+1)$ meilleurs chemins, de manière à ne conserver que des chemins possédants des séquences différentes de mots. Cependant, pour être tout à fait stricte, cette approche impose pour chaque chemin de conserver la séquence de mots préalablement détecté. Dans ce cas, on suppose donc les chemins identiques s'ils possèdent des séquences de mots identiques même si leurs segmentations sont différentes.

La seconde ("Word dependent N-Best") est une modification de la précédente, où l'on considère des chemins différents uniquement si leurs derniers mots rencontrés (hormis le mot courant) sont différents. Cette approche diminue la taille de l'historique nécessaire.

La dernière, appelée "N-Best" en treillis ("Lattice N-Best"), consiste, comme le montre la figure 24, à ne prendre en considération, lors des décisions locales, que les meilleurs chemins finissant sur les K mots différents (ainsi que le chemin finissant sur l'état lui même). Cet algorithme est identique à celui de Viterbi, à la différence près que l'on conserve en mémoire les N meilleures solutions qui seront utilisées pour la progression arrière. Comme le montre la figure 25, pour la conservation des différents sauts entre mots, il nous suffit maintenant de conserver les N meilleurs chemins finissant sur des mots différents.

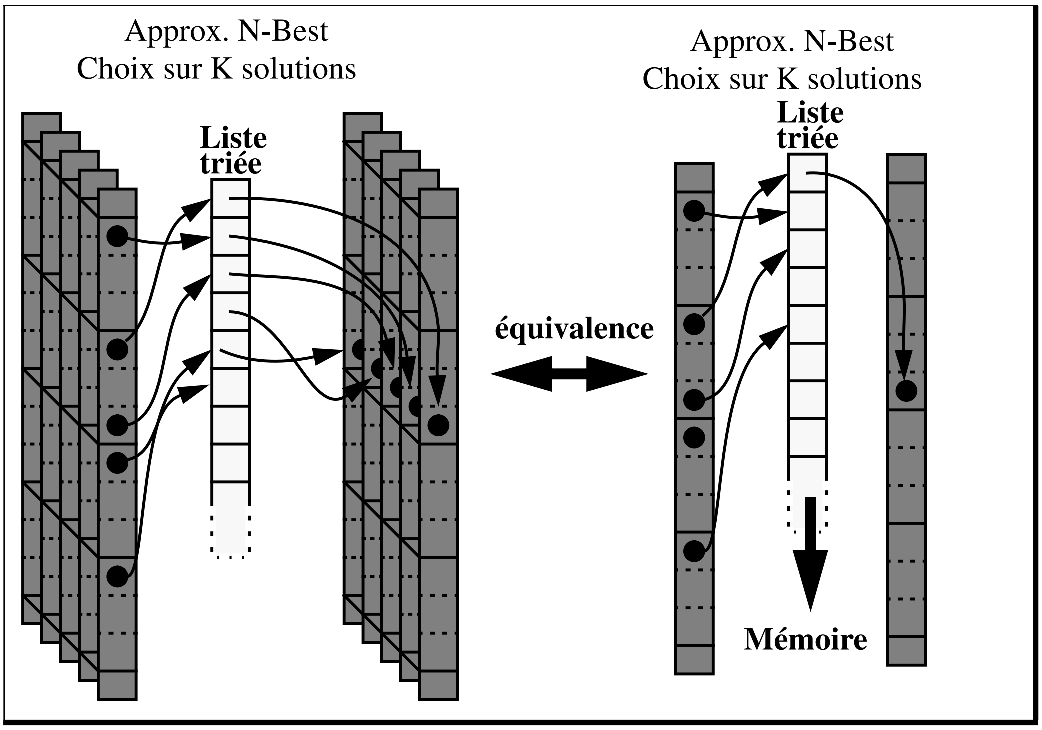

FIGURE 24. **Décision locale pour l'approximation du N-Best en treillis**

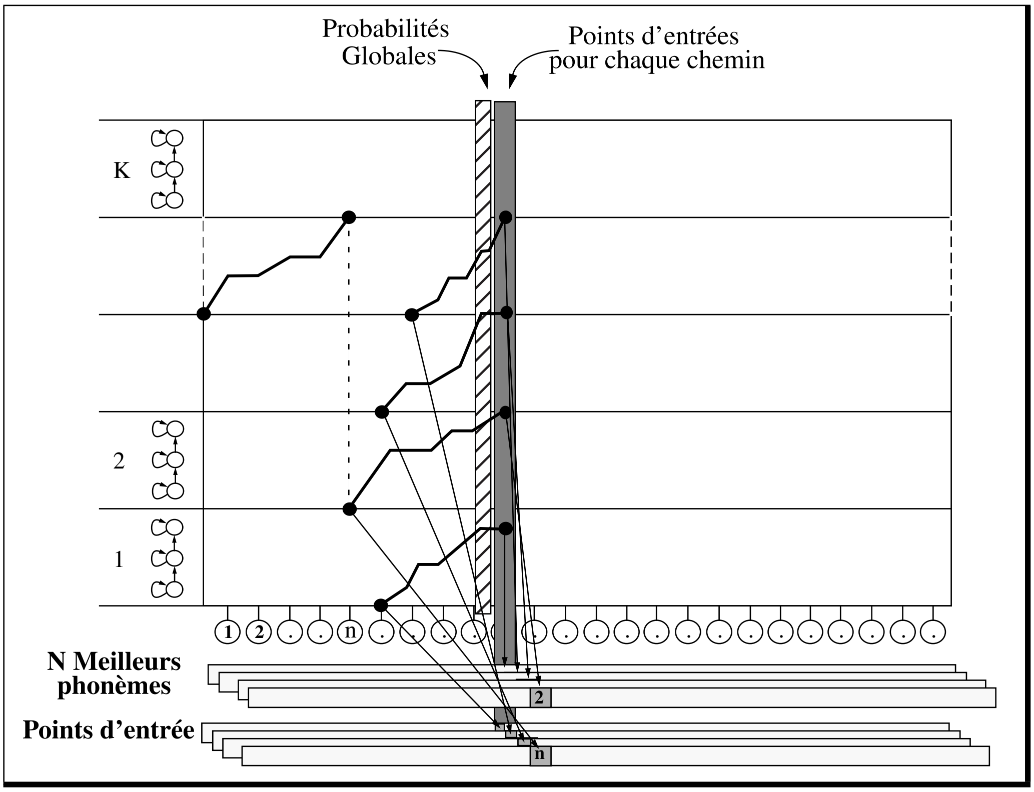

FIGURE 25. **N-Best en treillis.**

2.5.6 Propagation arrière automatique

2.5.6.1 Position du problème

La détection du meilleur chemin par l'algorithme de Viterbi nécessite deux passages (une progression avant et une progression arrière). La progression avant calcule les distances cumulées et conserve les décisions prisent localement dans le choix de différents chemins aboutissant en un point donné. La progression arrière reprend les décisions locales et détecte ainsi le chemin optimal.

Lors de l'indexation de longues séquences de parole, la place mémoire occupée par les vecteurs nécessaires à la progression avant peut devenir importante et poser des problèmes de gestion.

Dans un tout autre domaine d'application qui est la reconnaissance directe de parole, le temps de réponse du système de reconnaissance est critique, et nous ne pouvons nous permettre d'être en possession de tout l'enregistrement sonore (pouvant atteindre plusieurs dizaines de secondes) pour effectuer la progression arrière et donner la réponse.

2.5.6.2 Solution existante

Les méthodes utilisées jusqu'à présent pour éviter ces deux problèmes étaient basées sur le principe de *retour arrière prématuré*. Comme le montre la figure 26, ce principe repose sur l'hypothèse (qui s'avère expérimentalement correcte) qu'à n'importe quel instant t_2 donné, les K meilleurs chemins finissant sur les K états sont des chemins divergents provenant d'un même point situé à une "*distance de convergence*"correspondant à quelques phonèmes auparavant, soit t_1.

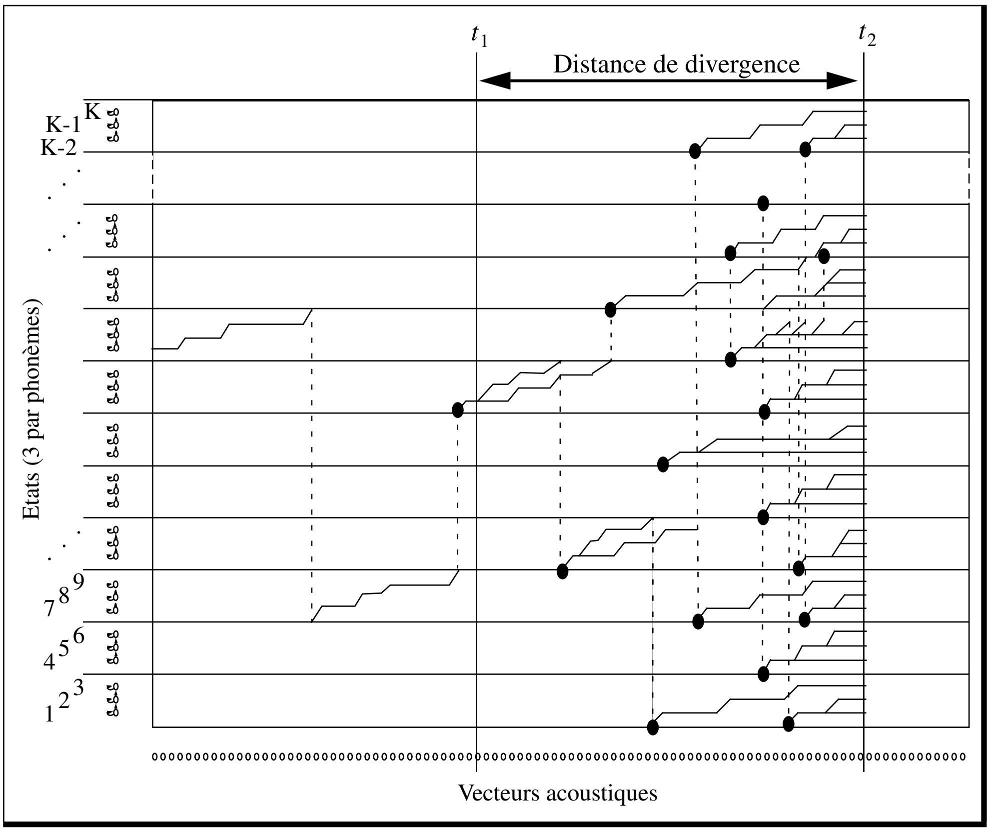

FIGURE 26. Divergence des chemins.

Comme le montre la figure 27, ces méthodes considéraient donc que l'on pouvait arrêter la

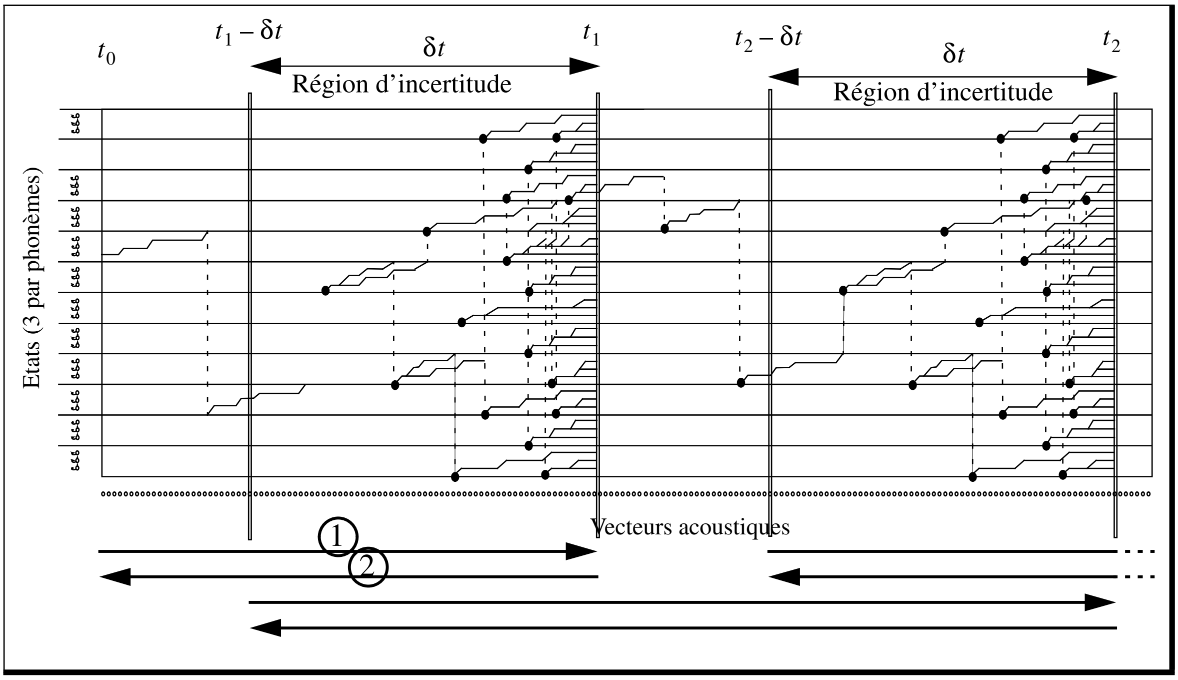

FIGURE 27. Schéma de rétropropagation prématuré.

progression avant, 1, à n'importe quel moment t_1, effectuer une progression arrière 2, à partir de n'importe quel état, considérer comme faux les quelques derniers phonèmes générés dans un intervalle de temps δt, fixé d'avance, et comme valide les phonèmes précédents (entre t_0 et $t_1 - \delta t$). La progression avant pouvait alors repartir du dernier phonème considéré valide, en $t_1 - \delta t$, jusqu'au prochain arrêt en t_2, et ainsi de suite.

Ce principe présente le désavantage de doubler inutilement les calculs dans les régions d'incertitude. De plus, la taille de ces régions est choisie arbitrairement (δt), et doit dès lors être prise suffisamment grande pour contenir la distance de convergence et ce dans toutes les configurations phonétiques envisageables.

2.5.6.3 Solution innovante

La solution présentée ici offre l'avantage de ne plus doubler les calculs sur ces régions d'incertitude. Sa mise en oeuvre impose uniquement l'utilisation d'un vecteur vertical supplémentaire, mais sans engendrer de prise de décision, ni de calcul supplémentaire.

Comme le schématise la figure 28., cette solution repose sur les étapes suivantes.

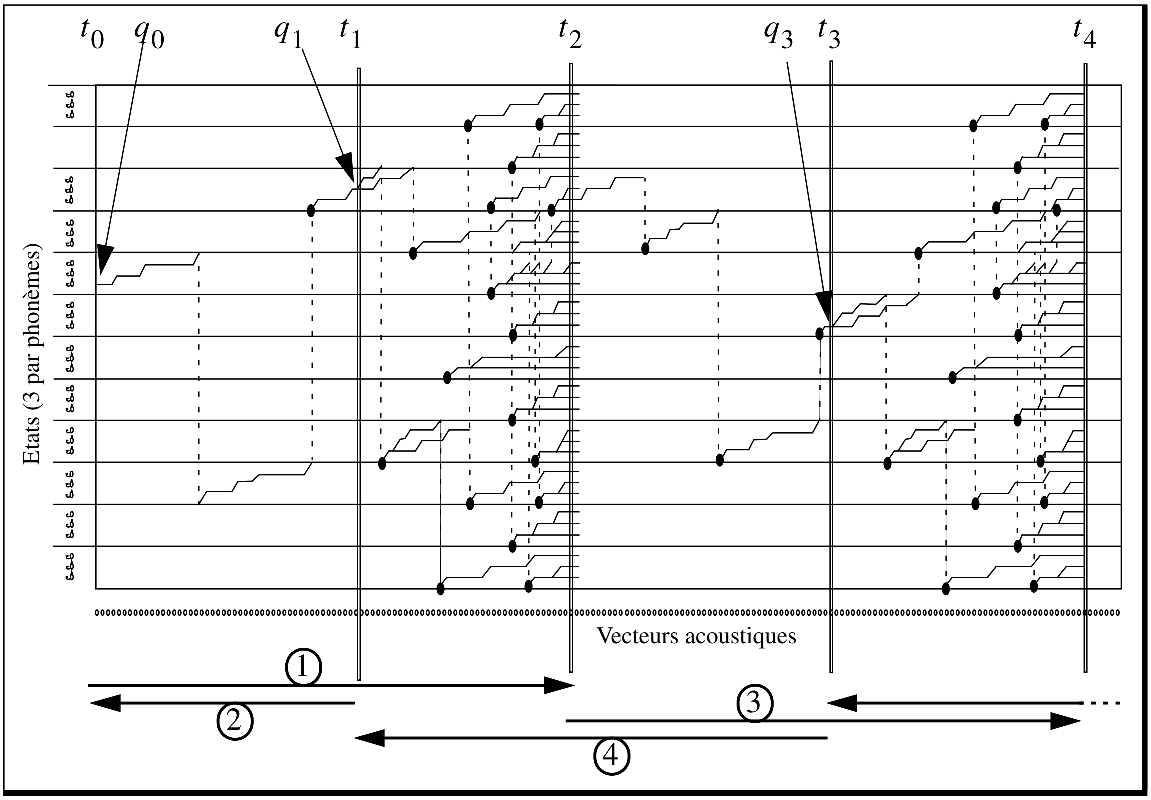

FIGURE 28. **Schéma optimisé.**

Lors de la progression avant, 1, on choisit un instant t_1 que l'on utilisera comme point de départ de la progression arrière, 2. Il suffit de déterminer l'état q_1 à partir duquel la progression arrière devra s'effectuer. Pour cela, on marque d'une étiquette différente chaque chemin finissant sur chaque état à l'instant t_1. Lors de la progression avant, on conserve à chaque décision l'étiquette du meilleur chemin (voir figure 29).

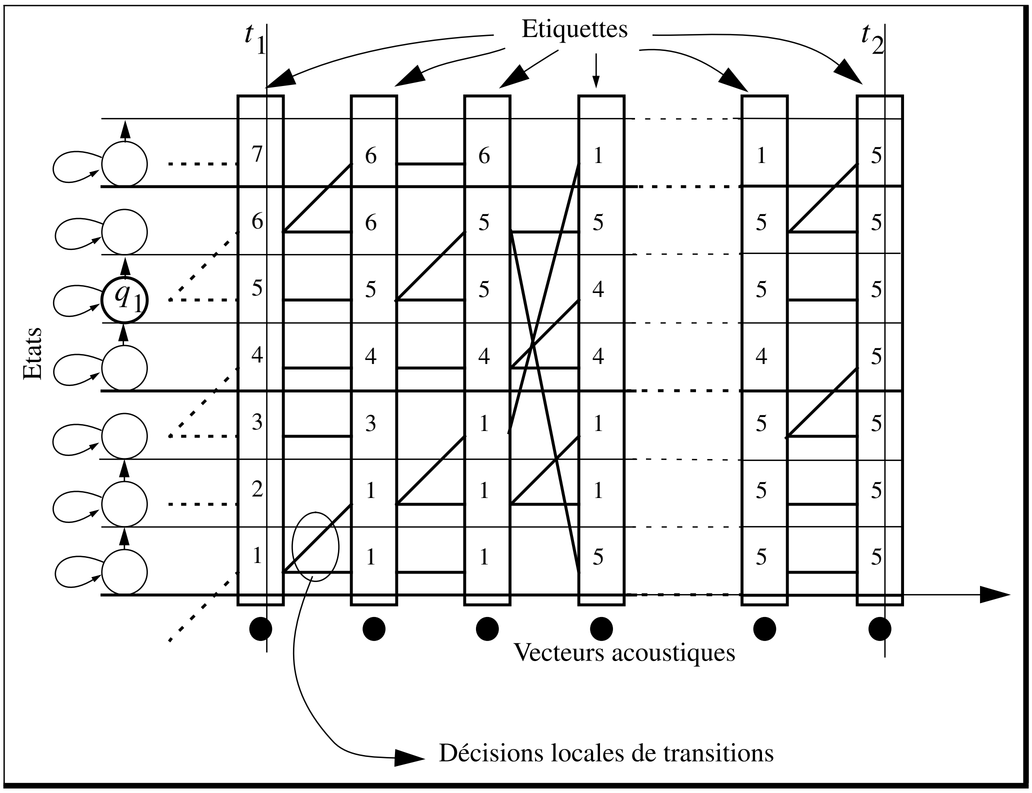

FIGURE 29. Convergence de l'étiquetage.

A chaque instant t, le nombre d'étiquettes reste identique ou se réduit. Nous savons, par expérience, que tous les chemins finissent par provenir d'un seul état, soit t_2 ce moment, généralement décalé par rapport à t_1 de quelques phonèmes. En t_2, on sait donc que tous les chemins contiennent le même état en t_1, soit q_1 cet état.

On peut donc ensuite, connaissant q_1, effectuer une progression arrière à partir de t_1, q_1 jusqu'en t_0, q_0. La segmentation générée ainsi est exacte (voir figure 28).

Ce principe peut être répété ultérieurement. Par exemple, en t_3, on associe de nouveau une étiquette différente pour chaque chemin finissant sur tous les états, et lors de la progression avant, 3, on recherche l'instant t_4 où tous les chemins finissent par provenir d'un seul état en t_3, soit q_3. On peut donc effectuer une progression arrière 4,à partir de (t_3, q_3) jusqu'en (t_1, q_1).

La fréquence maximale des propagations arrières est fixée par la vitesse de divergence des chemins. Si l'espace mémoire est une ressource critique, on peut optimiser son utilisation en imposant $t_3 = t_2$, ce qui revient à ré-initialiser le vecteur contenant les étiquettes des chemins dès la fin du processus de rétropropagation précédent.

Ce processus peut être aisément étendu à la recherche des N meilleurs chemins.

2.6 REMAP

Comme signalé plus haut (section 2.3.4), le défaut principal de l'approche classique est son critère de maximisation basé sur la vraisemblance, $P(X|M, \Lambda)$.

C'est pour cette raison que bon nombre de chercheurs tentent de supprimer cette contrainte en essayant de maximiser directement les probabilités a posteriori, $P(M|X, \Lambda)$.

H. Bourlard, Y. Konig et N. Morgan, [BOU95], ont introduit une approche originale pour la maximisation des probabilités a posteriori des modèles de parole. Cette approche, nommée REMAP ("Recursive Estimation and Maximization of A posteriori Probabilities"), s'appuie sur l'utilisation d'un réseau de neurones pour estimer efficacement les probabilités a posteriori.

Le principe inhérent à REMAP est de permettre l'évaluation de probabilités globales a posteriori $P(M|X, \Lambda)$, tout en évitant les hypothèses utilisées lors de l'approche du maximum de vraisemblance.

2.6.1 Modélisation

Partant de $P(M|X, \Lambda)$, montrons tout d'abord que cette probabilité peut être évaluée en tenant compte des probabilités "locales", $P(q_i^n \middle| X_{n-c}^{n+d}, q_j^{n-1}, \Lambda)$, appelées *probabilités de transition conditionnelles.*

Ces probabilités peuvent être estimées par un *réseau de neurones*, de type perceptron multicouches où le vecteur d'entrée contient, non seulement les vecteurs acoustiques, mais également une référence à l'état précédent, pour fournir à la sortie une estimation des probabilités de présence dans les différents états possibles, comme le montre la figure 30.

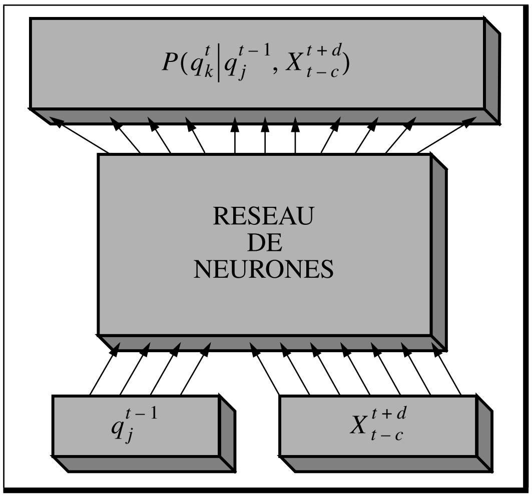

FIGURE 30. Configuration du réseau de neurones pour l'estimation des probabilités de transition conditionnelles.

En considérant tous les chemins, $\gamma \in \Gamma$ on peut écrire :

$$P(M|X, \Lambda) = \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda),$$

où la somme se réduit aux seuls chemins valides vis-à-vis de M.

En développant le membre de droite, on trouve :

$$P(M|X, \Lambda) = \sum_{\gamma \in \Gamma} P(\gamma|X, \Lambda) P(M|\gamma, X, \Lambda), \qquad \textbf{(EQ 30)}$$

et on montre que le premier facteur contribue au modèle acoustique et le deuxième facteur au modèle de langage.

2.6.1.1 Modèle acoustique

En décomposant plus loin $P(\gamma|X, \Lambda)$, on trouve facilement :

$$
\begin{aligned}
P(\gamma|X, \Lambda) &= P\left(q_{\gamma_1}^1, \gamma_2^N \middle| X, \Lambda\right) \\
&= P\left(q_{\gamma_1}^1 \middle| X, \Lambda\right) P\left(\gamma_2^N \middle| q_{\gamma_1}^1, X, \Lambda\right) \\
&= P\left(q_{\gamma_1}^1 \middle| X, \Lambda\right) P\left(q_{\gamma_2}^2, \gamma_3^N \middle| q_{\gamma_1}^1, X, \Lambda\right) \\
&= P\left(q_{\gamma_1}^1 \middle| X, \Lambda\right) P\left(q_{\gamma_2}^2 \middle| q_{\gamma_1}^1, X, \Lambda\right) P(\gamma_3^N | \gamma_1^2, X, \Lambda) \\
&= \dots \\
&= \prod_{n=1}^{N} P\left(q_{\gamma_n}^n \middle| \gamma_1^{n-1}, X, \Lambda\right),
\end{aligned}
$$

et en faisant successivement les hypothèses que :

- le modèle de Markov est du premier ordre :

$$P\left(q_{\gamma_n}^n \middle| \gamma_1^{n-1}, X, \Lambda\right) = P\left(q_{\gamma_n}^n \middle| q_{\gamma_{n-1}}^{n-1}, X, \Lambda\right);$$

- la probabilité de transition conditionnelle ne dépend que des vecteurs acoustiques temporellement proches (fenêtre $[n - c, n + d]$) :

$$P\left(q_{\gamma_n}^n \middle| q_{\gamma_{n-1}}^{n-1}, X, \Lambda\right) = P\left(q_{\gamma_n}^n \middle| q_{\gamma_{n-1}}^{n-1}, X_{n-c}^{n+d}, \Lambda\right);$$

on trouve directement que la probabilité conditionnelle d'un chemin est le produit des probabilités de transition conditionnelles :

$$P(\gamma|X, \Lambda) = \prod_{n=1}^{N} P\left(q_{\gamma_n}^n \middle| q_{\gamma_{n-1}}^{n-1}, X_{n-c}^{n+d}, \Lambda\right). \tag{EQ 31}$$

2.6.1.2 Modèle de langage

En reprenant la définition d'une chaîne de Markov cachée et de ses deux niveaux de processus aléatoires, il est clair, qu'une fois le chemin γ sélectionné, la séquence de vecteurs acoustiques X n'a plus d'effet sur la probabilité du modèle. On peut donc écrire, en toute généralité :

$$P(M|X, \gamma, \Lambda) = P(M|\gamma, \Lambda). \tag{EQ 32}$$

Considérons un ensemble de I modèles M_i, pouvant chacun générer un ensemble de chemins Γ_i. Si ces ensembles sont disjoints deux à deux, chaque chemin ne peut être généré que par un seul modèle, et on peut écrire :

$$P(M_i|\gamma, \Lambda) = \begin{cases} 1 \text{ si } \gamma \in \Gamma_i \\ 0 \text{ sinon} \end{cases} \quad \textbf{(EQ 33)}$$

Dans le cas plus général où un chemin peut être généré par différents modèles, on a :

$$P(M_i|\gamma, \Lambda) = \frac{P(\gamma, M_i|\Lambda)}{P(\gamma|\Lambda)} = \frac{P(\gamma|M_i, \Lambda)P(M_i|\Lambda)}{P(\gamma|\Lambda)}. \quad \textbf{(EQ 34)}$$

Remarquons que $P(\gamma|\Lambda)$ peut être obtenue à l'aide de :

$$P(\gamma|\Lambda) = \sum_j P(\gamma, M_j|\Lambda) = \sum_j P(\gamma|M_j, \Lambda)P(M_j|\Lambda). \quad \textbf{(EQ 35)}$$

Par ailleurs, en sommant **(EQ 34)** sur tous les modèles, et en y insérant **(EQ 35)** on retrouve :

$$\sum_i P(M_i|\gamma, \Lambda) = \frac{\sum_i P(\gamma|M_i, \Lambda)P(M_i|\Lambda)}{\sum_j P(\gamma|M_j, \Lambda)P(M_j|\Lambda)} = 1. \quad \textbf{(EQ 36)}$$

Cette dernière équation est valide pour tout chemin contenu dans au moins un des modèles. Pour les autres chemins, cette relation n'est cependant pas valide, et nous avons :

$$\sum_i P(M_i|\gamma, \Lambda) = 0. \quad \textbf{(EQ 37)}$$

En partant de **(EQ 34)** et en appliquant un développement similaire à celui utilisé lors de l'étude du modèle acoustique, on trouve :

$$P(M_i|\gamma,\Lambda) = \frac{\prod_{n=1}^{N} P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, M_i, \Lambda\right)P(M_i|\Lambda)}{\prod_{n=1}^{N} P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, \Lambda\right)} \tag{EQ 38}$$

$$= \prod_{n=1}^{N} \frac{P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, M_i, \Lambda\right)}{P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, \Lambda\right)} P(M_i|\Lambda),$$

où la seule hypothèse utilisée est :

- le modèle de Markov est du premier ordre.

2.6.1.3 Estimation de la probabilité a posteriori globale

En reprenant l'équation **(EQ 30)** et en y insérant la contribution du modèle acoustique **(EQ 31)** et du modèle de langage **(EQ 38)**, on trouve :

$$\boxed{P(M_i|X,\Lambda) = P(M_i|\Lambda) \sum_{\gamma \in \Gamma} \prod_{n=1}^{N} \frac{P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, X_{n-c}^{n+d}, \Lambda\right) P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, M_i, \Lambda\right)}{P\left(q_{\gamma_n}^{n}\middle|q_{\gamma_{n-1}}^{n-1}, \Lambda\right)}} \tag{EQ 39}$$

qui met en évidence le rôle de la séquence acoustique X par rapport à la probabilité a priori du mot M_i.

2.6.2 Critère discriminant

2.6.2.1 Hypothèse

Il est évident que l'entraînement est discriminant si l'on peut imposer lors de celui-ci :

$$\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j}|X_{\mathcal{E},i},\Lambda) = 1 \qquad ,\forall i = 1,\ldots,N_{\mathcal{E}}, \tag{EQ 40}$$

tout en maximisant :

$$P(M_{\mathcal{E},i} | X_{\mathcal{E},i}, \Lambda) \qquad ,\forall i = 1, \ldots, N_{\mathcal{E}}.$$

2.6.2.2 *Thèse*

Montrons qu' il suffit d'imposer :

$$\sum_{k=1}^{K} P(q_k^n | q_i^{n-1}, X_{n-c}^{n+d}, \Lambda) = 1 \qquad ,\forall i, n \qquad \textbf{(EQ 41)}$$

pour retrouver **(EQ 40)**.

2.6.2.3 *Démonstration*

En effet, **(EQ 40)** peut se développer en :

$$\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \Lambda) = \sum_{\gamma \in \Gamma} \sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j}, \gamma | X_{\mathcal{E},i}, \Lambda),$$

où Γ peut être réduit à l'ensemble de tous les chemins valides dans tous les modèles M_i, et de longueur identique à $X_{\mathcal{E},i}$.

En développant le membre de droite, on trouve successivement :

$$| X_{\mathcal{E},i}, \Lambda) = \sum_{\gamma \in \Gamma} \sum_{j=1}^{\mathcal{E}} P(\gamma | X_{\mathcal{E},i}, \Lambda) P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \gamma,$$

$N_{\mathcal{E}}.$

Etant donné **(EQ 36)**, on a directement :

$$\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \Lambda) = \sum_{\gamma \in \Gamma} P(\gamma | X_{\mathcal{E},i}, \Lambda). \qquad \textbf{(EQ 42)}$$

En effectuant le même développement que lors de la description du modèle acoustique **(EQ 31)**, et en utilisant les mêmes hypothèses (modèle de Markov du premier ordre et dépendance des probabilités de transition conditionnelles restreintes aux vecteurs acoustiques temporellement proches), on obtient :

$$\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \Lambda) = \sum_{\gamma \in \Gamma} \prod_{n=1}^{N} P\left(q_{\gamma_n}^{n} \middle| q_{\gamma_{n-1}}^{n-1}, X_{\mathcal{E},i,n-c}^{n+d}, \Lambda\right). \quad \textbf{(EQ 43)}$$

En écrivant la somme sur les chemins en fonction de la somme sur les différents états aux différents instants, on trouve :

$$\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \Lambda) = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \dots \sum_{k_N=1}^{K} \prod_{n=1}^{N} P\left(q_{k_n}^{n} \middle| q_{k_{n-1}}^{n-1}, X_{\mathcal{E},i,n-c}^{n+d}, \Lambda\right). \quad \textbf{(EQ 44)}$$

Le passage de l'équation **(EQ 43)** à l'équation **(EQ 44)** implique une hypothèse :

- l'ensemble des paramètres, Λ, doit contenir la description syntaxique de tous les modèles impliqués lors de l'entraînement, de façon à déterminer si la transition est contenue dans au moins un chemin valide.

En mettant successivement en évidence les probabilités conditionnelles relatives aux états visités aux instants N, $N-1$,..., on trouve :

$$\begin{aligned}
\sum_{j=1}^{N_{\mathcal{E}}} P(M_{\mathcal{E},j} | X_{\mathcal{E},i}, \Lambda) &= \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \dots \sum_{k_{N-1}=1}^{K} \left(\prod_{n=1}^{N-1} P\left(q_{k_n}^{n} \middle| q_{k_{n-1}}^{n-1}, X_{\mathcal{E},i,n-c}^{n+d}, \Lambda\right)\right) \\
&\qquad \sum_{k_N=1}^{K} P\left(q_{k_N}^{N} \middle| q_{k_{N-1}}^{N-1}, X_{\mathcal{E},i,N-c}^{N}, \Lambda\right) \\
&= \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \dots \sum_{k_{N-1}=1}^{K} \prod_{n=1}^{N-1n} P\left(q_{k_n}^{n} \middle| q_{k_{n-1}}^{n-1}, X_{\mathcal{E},i,n-c}^{n+d}, \Lambda\right) \\
&= \dots \\
&= \sum_{k_1=1}^{K} P\left(q_{k_1}^{1} \middle| X_{\mathcal{E},i,1}^{n+d}, \Lambda\right) \\
&= 1.
\end{aligned}$$

2.6.2.4 Conséquence des hypothèses sur l'entraînement.

On peut déduire des développements précédents qu'une estimation correcte de $P(q_k^n | q_i^{n-1}, X_{n-c}^{n+d}, \Lambda)$ permet d'assurer un caractère discriminant à l'apprentissage. Cependant, nous avons vu plus haut que l'ensemble des paramètres, Λ, devait contenir la description des modèles utilisés lors de l'entraînement, de façon à ne pas tenir compte des chemins non valides.

En d'autres termes, si le modèle markovien réduit au temps n les accès à un sous ensemble d'états $\tilde{Q}$, l'estimation de $P(q_k^n | q_i^{n-1}, X_{n-c}^{n+d}, \Lambda)$ doit tenir compte de la contrainte :

$$\sum_{q_k \in \tilde{Q}} P(q_k^n | q_i^{n-1}, X_{n-c}^{n+d}, \Lambda) = 1 .$$

Lors de l'entraînement, il faut donc rester vigilant et normaliser de manière à vérifier cette dernière équation.

2.6.3 Formules de récurrence

Comme pour l'approche Baum-Welch, on se base sur des formules de récurrence avant et arrière pour permettre l'évaluation des probabilités a posteriori.

Pour pouvoir séparer X en X_1^t et X_{t+1}^N, il est nécessaire de revenir aux probabilités jointes, conditionnées par M :

$$P(X|M) = \sum_{k=1}^{K} P(q_k^t, X | M) \qquad ,\forall t . \qquad \textbf{(EQ 45)}$$

De la même manière que dans l'approche Baum-Welch, nous pouvons définir le premier facteur comme étant la *probabilité avant* :

$$\alpha_k^t = P(q_k^t, X_1^t | M) , \qquad \textbf{(EQ 46)}$$

tandis que le deuxième facteur, nommé *probabilité arrière* est :

$$\gamma_k^t = P(X_{t+1}^N | q_k^t, X_1^t, M) . \qquad \textbf{(EQ 47)}$$

Nous avons donc :

$$P(q_k^t, X \middle| M) = \alpha_k^t \gamma_k^t. \qquad \textbf{(EQ 48)}$$

En sommant cette dernière sur k, on retrouve :

$$P(X|M) = \sum_{k=1}^{K} \alpha_k^t \gamma_k^t \qquad ,\forall t. \qquad \textbf{(EQ 49)}$$

Finalement, en utilisant la loi de Bayes, on peut écrire :

$$\boxed{P(M|X) = \frac{P(M) \sum_{k=1}^{K} \alpha_k^t \gamma_k^t}{P(X)}} \qquad \textbf{(EQ 50)}$$

Recherchons maintenant les formules de récurrence conduisant à l'estimation des différents α_k^t et γ_k^t.

2.6.3.1 *Récurrence avant*

Montrons que $\alpha_i^t = P(q_i^t, X_1^t \middle| M)$ peut être développée sous une forme récurrente.

En effet, on a :

$$\begin{aligned}
P(q_i^t, X_1^t \middle| M) &= \sum_{k=1}^{K} P(q_k^{t-1}, q_i^t, x_t, X_1^{t-1} \middle| M) \\
&= \sum_{k=1}^{K} P(q_k^{t-1}, X_1^{t-1} \middle| M) P(q_i^t, x_t \middle| q_k^{t-1}, X_1^{t-1}, M) \\
&= \sum_{k=1}^{K} P(q_k^{t-1}, X_1^{t-1} \middle| M) P(x_t \middle| q_k^{t-1}, X_1^{t-1}, M) P(q_i^t \middle| q_k^{t-1}, X_1^t, M).
\end{aligned}$$

En faisant les hypothèses successives :

- les probabilités de transition conditionnelles ne dépendent que des vecteurs acoustiques temporellement proches, et l'on peut également leurs adjoindre les vecteurs acoustiques appartenant au voisinage s'étendant vers le futur :

$$P(q_i^t | q_k^{t-1}, X_1^t, M) = P(q_i^t | q_k^{t-1}, X_{t-c}^{t+d}, M);$$

- la probabilité d'émettre x_t, connaissant X_1^{t-1} et l'état précédent q_k^{t-1}, peut être soit estimée par un simple algorithme de prédiction acoustique dépendant de l'état précédent, soit supposée constante. En notant c_k^t cette probabilité nous avons :

$$P(x_t | q_k^{t-1}, X_1^{t-1}, M) = c_k^t, \quad \text{(EQ 51)}$$

on obtient :

$$\boxed{\alpha_i^t = \sum_{k=1}^{K} \alpha_k^{t-1} P(q_i^t | q_k^{t-1}, X_1^t, M) c_k^t} \quad \text{(EQ 52)}$$

Cette récurrence avant est semblable à celle de l'algorithme Baum-Welch, **(EQ 19)**. La différence fondamentale réside dans l'interprétation donnée à la *contribution locale* $P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M)$ qui, dans REMAP, est décomposée en *probabilité de transition conditionnelle* et en *prédiction acoustique* dépendant de l'état courant :

$$\boxed{P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M) = P(q_i^t | q_k^{t-1}, X_1^t, M) P(x_t | q_k^{t-1}, X_1^{t-1}, M)} \quad \text{(EQ 53)}$$

tandis que dans Baum-Welch, **(EQ 18)**, elle était décomposée en probabilité de transition (supposée par la suite non conditionnelle) et en probabilité d'émission.

2.6.3.2 Récurrence arrière

Montrons que $\gamma_i^t = P(X_{t+1}^N | q_i^t, X_1^t, M)$ peut être également développée sous une forme récurrente.

En effet, on a :

$$
\begin{aligned}
P(X_{t+1}^{N}|q_i^t, X_1^t, M) &= \sum_{k=1}^{K} P(q_k^{t+1}, x_{t+1}, X_{t+2}^{N}|q_i^t, X_1^t, M) \\
&= \sum_{k=1}^{K} P(q_k^{t+1}, x_{t+1}|q_i^t, X_1^t, M) P(X_{t+2}^{N}|q_k^{t+1}, q_i^t, X_1^{t+1}, M) \\
&= \sum_{k=1}^{K} P(x_{t+1}|q_i^t, X_1^t, M) P(q_k^{t+1}|q_i^t, X_1^t, M) P(X_{t+2}^{N}|q_k^{t+1}, q_i^t, X_1^{t+1}, M).
\end{aligned}
$$

En faisant les hypothèses successives :

- le modèle de Markov est du premier ordre :

  $P(X_{t+2}^{N}|q_k^{t+1}, q_i^t, X_1^{t+1}, M) = P(X_{t+2}^{N}|q_k^{t+1}, q_i^t, X_1^{t+1}, M)$;

- les probabilités de transition conditionnelles ne dépendent que des vecteurs acoustiques temporellement proches, et l'on peut également leurs adjoindre les vecteurs acoustiques appartenant au voisinage s'étendant vers le futur :

  $P(q_k^{t+1}|q_i^t, X_1^t, M) = P(q_k^{t+1}|q_i^t, X_{t-c}^{t+d}, M)$;

- la probabilité d'émettre x_{t+1}, connaissant X_1^t et l'état précédent q_k^t, peut être soit estimée par un simple algorithme de prédiction acoustique dépendant de l'état précédent, soit supposée constante. Nous noterons par c_i^{t+1} cette probabilité :

  $P(x_{t+1}|q_i^t, X_1^t, M) = c_i^{t+1}$,

on obtient, après avoir mis en évidence le facteur indépendant de la somme :

$$
\boxed{\gamma_i^t = c_i^{t+1} \sum_{k=1}^{K} P(q_k^{t+1}|q_i^t, X_{t-c}^{t+d}, M) \gamma_k^{t+1}} \qquad \textbf{(EQ 54)}
$$

De nouveau, cette équation de récurrence arrière ne diffère de celle de Baum-Welch que par l'interprétation donnée à la probabilité locale.

Etant en possession des formules de récurrences pour l'estimation de la probabilité a posteriori, nous pouvons maintenant rechercher une méthode d'entraînement nous permettant d'estimer les paramètres du modèle qui, à la différence d'une approche mixte HMM-réseau de neurones, sont tous contenus dans le réseau de neurones, y compris les probabilités de transition.

2.6.4 Estimation des paramètres

Le grand intérêt de REMAP réside dans sa méthode d'entraînement qui conserve un caractère discriminant.

En effet, comme le montre H. Bourlard dans [BOU95], il existe une méthode de réévaluation des paramètres Λ, qui assure la convergence (vers un minimum local) de :

$$\prod_i P(M_{\mathcal{E},\, i} | X_{\mathcal{E},\, i}, \Lambda)\,. \qquad \text{(EQ 55)}$$

La démonstration de cette propriété est reportée à l'annexe B. Seuls les résultats sont reproduits ici.

On y montre d'un part que chaque couple $\left\{X_{t-c}^{t+d}, q_j^{t-1}\right\}$ doit être présenté pour l'apprentissage du réseau de neurones avec une fréquence proportionnelle à la *probabilité de visite* :

$$P(q_j^{t-1} | X, M, \Lambda)\,, \qquad \text{(EQ 56)}$$

et d'autre part que l'objectif associé à l'état de sortie q_k doit être égal à :

$$O_k(X_{t-c}^{t+d}, q_j^{t-1}) = P(q_k^t | X, q_j^{t-1}, M, \Lambda)\,. \qquad \text{(EQ 57)}$$

Ainsi, les valeurs obtenues à la sortie du réseau de neurones après convergence tendent vers les probabilités :

$$P(q_k^t | X_{t-c}^{t+d}, q_j^{t-1}, M, \Lambda') = P(q_k^t | X, q_j^{t-1}, M, \Lambda)\,, \qquad \text{(EQ 58)}$$

où Λ' est le nouvel ensemble de paramètres, et Λ l'ancien.

Il nous suffit donc d'estimer à chaque itération $P(q_j^{t-1} | X, M, \Lambda)$ et $P(q_k^t | X, q_j^{t-1}, M, \Lambda)$ en fonction des probabilités de transition conditionnelles locales.

2.6.4.1 Estimation des probabilités de visite

En développant **(EQ 56)** par la loi de Bayes, on trouve :

$$P(q_k^{t-1}\big|X, M, \Lambda) = \frac{P(q_k^{t-1}, X\big|M, \Lambda)}{P(X|M, \Lambda)}.$$

En utilisant **(EQ 48)** pour le numérateur et **(EQ 49)** pour le dénominateur, on trouve :

$$\boxed{P(q_k^{t-1}\big|X, M, \Lambda) = \frac{\alpha_k^{t-1}\gamma_k^{t-1}}{\sum_{k=1}^{K}\alpha_k^{t-1}\gamma_k^{t-1}}} \quad \text{(EQ 59)}$$

2.6.4.2 Calcul des objectifs

Il est aisé de voir que la probabilité recherchée $P(q_k^t\big|X, q_j^{t-1}, M, \Lambda)$ peut être développée suivant :

$$P(q_k^t\big|X, q_j^{t-1}, M, \Lambda) = \frac{P(q_k^t, q_j^{t-1}, X\big|M, \Lambda)}{P(X, q_j^{t-1}\big|M, \Lambda)}. \quad \text{(EQ 60)}$$

En utilisant les mêmes hypothèses que lors de la récurrence arrière, le numérateur peut être estimé par :

$$\begin{aligned}
P(q_k^t, q_j^{t-1}, X\big|M, \Lambda) &= P(q_k^t, q_j^{t-1}, X_1^{t-1}, X_t^N\big|M, \Lambda) \\
&= P(q_j^{t-1}, X_1^{t-1}\big|M, \Lambda)P(q_k^t, X_t^N\big|q_j^{t-1}, X_1^{t-1}, M, \Lambda) \\
&= \alpha_j^{t-1}P(q_k^t, x_t, X_{t+1}^N\big|q_j^{t-1}, X_1^{t-1}, M, \Lambda) \\
&= \alpha_j^{t-1}P(q_k^t, x_t\big|q_j^{t-1}, X_1^{t-1}, M, \Lambda)P(X_{t+1}^N\big|q_k^t, q_j^{t-1}, x_t, X_1^{t-1}, M, \Lambda) \\
&= \alpha_j^{t-1}P(x_t\big|q_j^{t-1}, X_1^{t-1}, M, \Lambda)P(q_k^t\big|q_j^{t-1}, X_1^{t-1}, M, \Lambda)P(X_{t+1}^N\big|q_k^t, X_1^t, M, \Lambda) \\
&= \alpha_j^{t-1}c_j^tP(q_k^t\big|q_j^{t-1}, X_{t-c}^{t+d}, M, \Lambda)\gamma_k^t.
\end{aligned}$$

En utilisant **(EQ 48)** pour le dénominateur, on trouve directement :

$$P(q_k^t\big|X, q_j^{t-1}, M, \Lambda) = \frac{\alpha_j^{t-1}c_j^tP(q_k^t\big|q_j^{t-1}, X_{t-c}^{t+d}, M, \Lambda)\gamma_k^t}{\alpha_j^{t-1}\gamma_j^{t-1}},$$

et en simplifiant les α_j^{t-1}, on obtient :

$$P(q_k^t \mid X, q_j^{t-1}, M, \Lambda) = \frac{c_j^t P(q_k^t \mid q_j^{t-1}, X_{t-c}^{t+d}, M)\gamma_k^t}{\gamma_j^{t-1}} \qquad \text{(EQ 61)}$$

2.6.4.3 *Schéma d'entraînement*

En regroupant les informations précédentes, nous pouvons résumer la méthode d'entraînement par le schéma montré à la 31 :

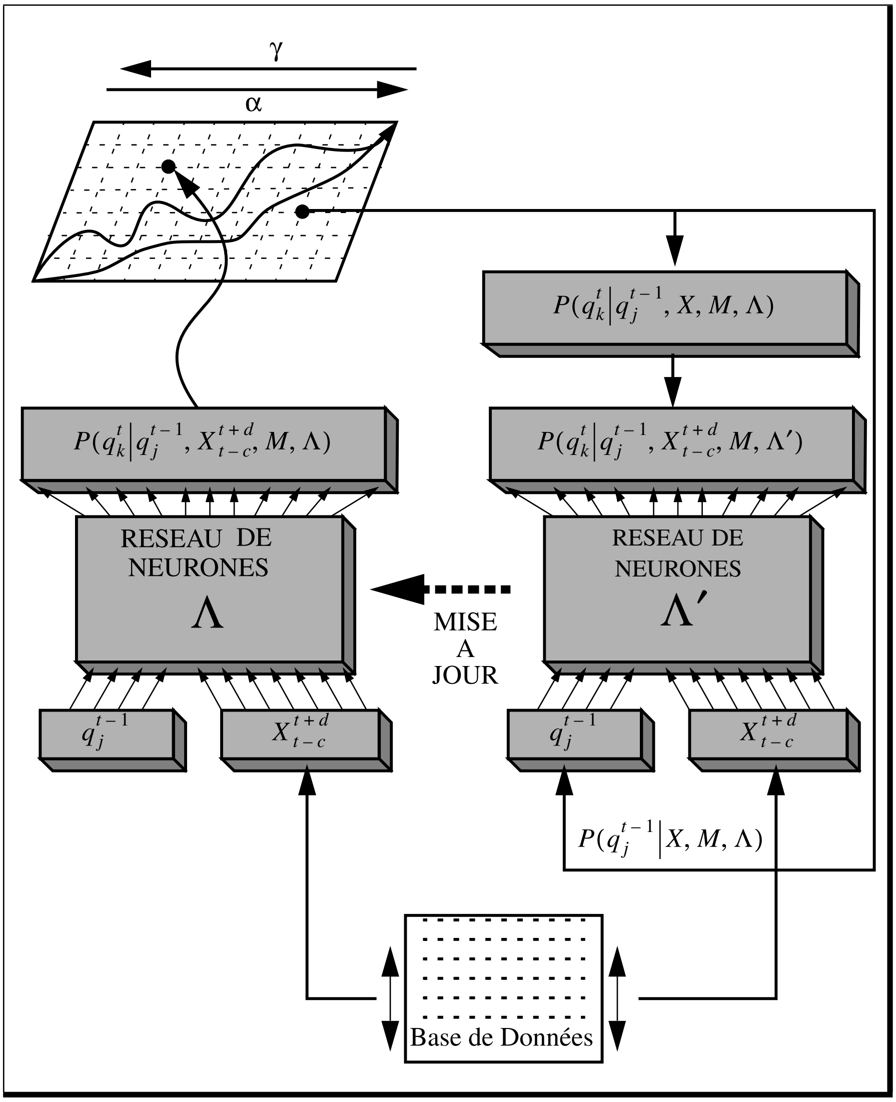

FIGURE 31. **Schéma d'entraînement.**

Le principe d'entraînement est semblable à celui expliqué dans l'algorithme de Baum-Welch, section 2.4.2. Cependant, ils diffèrents en deux points. Tout d'abord, le réseau de neurone prend maintenant également en entrée une référence au dernier état visité. De plus nous ne devons plus estimer les probabilités de transitions qui sont modèlisés intrinsèquement dans le réseau lui-même.

Comme le montre cette figure, nous utilisons un réseau de neurones pour analyser chaque phrase de la base de données. Cette analyse effectue une progression avant α et arrière γ de manière à associer à chaque X_{t-c}^{t+d}, les couples possibles $\left\{X_{t-c}^{t+d}, q_j^{t-1}\right\}$, leur probabilité d'apparition $P(q_j^{t-1} \mid X, M, \Lambda)$ et les objectifs à atteindre $P(q_k^t \mid X, q_j^{t-1}, M, \Lambda)$.

Les paires $\left\{X_{t-c}^{t+d}, q_j^{t-1}\right\}$ n'étant pas équiprobables, il est nécessaire de les pondérer lors de l'apprentissage en fonction de $P(q_j^{t-1} \mid X, M, \Lambda)$. Deux méthodes sont envisageables. La première consiste à pondérer la correction apportée au réseau en fonction de cette probabilité. La deuxième, que nous utilisons, consiste à présenter cette paire à l'entrée du réseau avec une fréquence proportionnelle à sa probabilité d'apparition.

CHAPITRE 3

Etat de l'art

Dans ce chapitre, nous présentons tout d'abord les méthodes d'évaluation des systèmes de reconnaissance en vue de comparer leur efficacité dans le contexte d'applications précises.
Ensuite, nous soulignons les idées novatrices qui ont contribué à l'évolution des systèmes de reconnaissance de mots clés aux cours de ces dernières années.
Nous montrons finalement les travaux récemment développés en détection de mots clés pour le tri automatique de documents en fonction des thèmes abordés par ceux-ci.

3.1 Les méthodes d'évaluation

3.1.1 Problématique

Comparer deux systèmes de reconnaissance de parole est difficile.

On peut classifier les systèmes suivant différents critères : la qualité du modèle acoustique employé, la grammaire appliquée, la taille du vocabulaire traité, la rapidité du traitement, l'indépendance vis-à-vis des locuteurs, la robustesse aux bruits,... Cependant le critère définitif pour comparer deux systèmes reste le taux de reconnaissance pour la tâche envisagée.

Toutefois, on ne peut comparer deux systèmes par ce taux de reconnaissance, que si ils sont évalués dans des conditions d'application identiques. C'est pourquoi la définition de bases de données universellement reconnues est indispensable.

Si les systèmes ont été élaborés dans une optique d'utilisation différente, la comparaison est plus compliquée. En effet, on ne peut que difficilement adapter les systèmes à une tâche de référence. C'est pourquoi on s'est appliqué, au cours du temps, à développer des méthodes d'évaluation qualitative les plus indépendantes possibles des applications envisagées.

3.1.2 La perplexité

Il est clair, qu'un système de reconnaissance qui doit choisir entre deux mots, est plus facile à mettre en oeuvre et est plus robuste qu'un système d'écriture automatique basé sur un vocabulaire de plus de dix mille mots. Pour tenir compte de la complexité de la tâche à accomplir par le système de reconnaissance et la chiffrer, on a recours à la mesure de la *perplexité*.

Imaginons un vocabulaire de M mots. Ces mots s'enchaînent pour former des phrases, à l'aide de règles définies par une grammaire.

Au premier abord, on peut voir la perplexité d'une grammaire comme le nombre moyen de mots qui peuvent concourir comme successeur d'un mot.

Plus précisément, si l'on note j le point de décision où concourent M_j mots, et $P(m|j)$ la probabilité que l'un de ces mot, m, soit choisi au point j, l'entropie associée à ce point est définie par :

$$H_j = \sum_{m=1}^{M_j} P(m|j)log_2(P(m|j)) .$$

La *perplexité* en un point j est définie comme : $Q_j = 2^{H_j}$.

L' *entropie* de la grammaire est définie par : $H = \sum_{j=1}^{J} P(j) H_j$.

où J est le nombre total de points de décision de la grammaire et $P(j)$ la probabilité a priori de passer sur ce point de décision.

Dans ce cas, la *perplexité de la grammaire* est : $Q = 2^H$.

La mesure de perplexité s'applique généralement aux systèmes basés sur une grammaire de mots. Dans ce cas, les valeurs rencontrées sont de l'ordre de 10 pour un système ayant de fortes contraintes, comme un système de reconnaissance de commandes vocales, et pouvant aller jusqu'à 60 pour des systèmes plus larges, tels que la grammaire par paires de mots de DARPA-RM.

Dans le cas de l'indexation, nous utilisons un vocabulaire ouvert et sommes contraints de travailler au niveau phonétique. Nous basons donc notre méthode uniquement sur une grammaire de paires de phonèmes.

Si l'on suppose le système sans contraintes sur les séquences phonétiques, on obtient $P(m|j) = P(m)$, ce qui conduit à une valeur approximative de 40 pour la perplexité.

Si l'on tient compte des probabilités de transition a priori entre phonèmes, la perplexité chute aux environs de 16.

Il est cependant bon de noter que les valeurs associées à une modélisation phonétique ne peuvent être comparées directement à une modélisation par mots. La durée des modèles est en effet différente et par conséquent le nombre de décisions par seconde est également différent. Il faudrait, pour obtenir une mesure comparable, pondérer la perplexité par rapport au temps moyen entre deux décisions.

3.1.3 Estimation du taux d'erreur

La reconnaissance de parole, implique une gestion rigoureuse des erreurs de décision. Ces erreurs d'origines diverses doivent être estimées de façon identique pour permettre la comparaison des différents systèmes de reconnaissance.

Dans le cas d'un système de reconnaissance de parole continue, on distingue trois types d'erreurs :

- l'erreur de substitution (subst) : un mot est confondu avec un autre ;
- l'erreur de suppression (supp) : un mot n'est pas trouvé ;
- l'erreur d'insertion (ins) : un mot est trouvé alors qu'il n'est pas prononcé.

Pour la reconnaissance de mots isolés, l'estimation du taux d'erreur est directe car il ne peut y avoir que des erreurs de substitution.

$$\text{Taux d'erreur} = 100 \cdot \frac{\text{nbr de subst.}}{\text{nbr total de mots}}$$

En reconnaissance continue, le problème se complique car les trois types d'erreur existent. De plus il est possible de considérer une substitution comme la combinaison simultanée d'une suppression et d'une insertion ce qui reviendrait alors à considérer cette erreur comme double.

Kai-Fu Lee, dans [LEE89], introduit d'une part la définition du *pourcentage correct* qui ne tient pas compte des insertions et d'autre part la définition de l'*exactitude au niveau du mot* qui en tient compte :

$$\text{Pourcentage correct} = 100 \cdot \frac{\text{nbr correct}}{\text{nbr total de mots}}.$$

$$\text{Taux d'erreur} = 100 \cdot \frac{\text{nbr de subst} + \text{nbr de supp} + \text{nbr d ins}}{\text{nbr total de mots}}$$

$$\text{Exactitude au niveau du mot} = 1 - \text{Taux d erreur} = 100\frac{\text{nbr correct} - \text{nbr d ins}}{\text{nbr total de mots}},$$

Où nbr correct correspond au nombre de mots reconnus correctement.

3.1.4 Mots clés

Dans le cas de la reconnaissance de mots clés, les définitions précédentes ne sont plus valables, car la nature des erreurs a changé. On distingue maintenant deux types d'erreurs :

- la *non détection* du mot clé, à laquelle est associée la probabilité de ne pas détecter le mot clé alors qu'il est prononcé ;
- la *fausse alarme*, à laquelle est associée la probabilité de détecter un mot clé alors qu'il n'est pas prononcé.

De plus, la mesure d'efficacité du système doit rester indépendante de la fréquence des mots clés dans le texte à analyser.

En accord avec ces contraintes, on utilise la probabilité de détection des mots clés en fonction de la probabilité de fausses alarmes par mot clé et par heure pour mesurer l'efficacité du système. Cette courbe est appelée *courbe caractéristique d'opération du récepteur ("Reciever Operating Curves", "R.O.C.")*. Nous représentons ci-dessous un exemple de courbes caractéristiques issus de [ROH93].

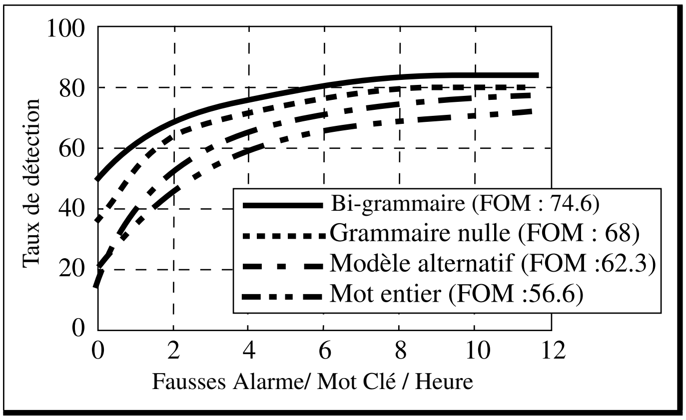

FIGURE 32. **Courbes caractéristiques d'opération du récepteur et valeurs de mérites.**

Pour obtenir une valeur unique décrivant cette courbe, on utilise la moyenne des probabilités de détection pour un taux de fausses alarmes par mot clé et par heure variant entre 0 et 10. Cette valeur est appelée *Valeur de mérite* ("Figure Of Merit", "FOM")

3.1.5 Tri par le contenu

Quand on recherche, dans une base de données, les séquences associées au sujet que l'on désire analyser, on s'appuie généralement sur la recherche de plusieurs mots clés de façon à trier ces phrases. On introduit alors une mesure d'efficacité du système plus complexe que ceux utilisés pour la détection de mot-clés.

Le système d'extraction trie les phrases de la base de données en fonction décroissante de leur probabilité d'appartenance au sujet recherché. Parmi cette liste triée, il ne sélectionnera que les premières. C'est la raison pour laquelle certains auteurs, [JAME94][YOUN94], introduisent le concept de précision.

La *précision* du système vis-à-vis d'une requête est définie en fonction du nombre de phrases retenues dans cette liste. Pour un nombre fixé de phrases sélectionnées, la précision correspond au rapport du nombre de phrases traitant réellement au sujet sur le nombre total de phrases sélectionnées.

Par exemple, si le résultat d'une requête fournit 120 phrases triées suivant leur probabilité de satisfaire cette requête et si les phrases traitant réellement du sujet se situent aux places repérées dans le tableau (1),

Contenu	X		X	X			X	X	X		X		
Place	1	2	3	4	5	6	7	8	9	...	88	...	120

TABLEAU 1. **Position des phrases traitant du sujet recherché.**

alors la précision prendra les valeurs suivantes en fonction du nombre de phrases sélectionnées :

Place	1	2	3	4	5	6	7	8	9	...	88	...	120
Précision	1	1/2	2/3	3/4	3/5	3/6	4/7	5/8	6/9	...	7/88	...	7/120

TABLEAU 2. **Valeur de la précision.**

Appelons “précision standard” (cfr tableau (3))pour une requête donnée, la moyenne de la précision sur les places où une phrase correctement classée apparaît (les zones grisées du tableau (2))

Occurrence	1	2	3	4	5	6	7	Préc. Std.
Place	1	3	4	7	8	9	88	
Précision	1	2/3	3/4	4/7	5/8	6/9	7/88	0,6

TABLEAU 3. **Précision standard.**

La *précision moyenne* est alors définie par la moyenne des “précisions standards” pour toutes les requêtes de la précision.

Dans l'exemple ci-dessus, n'ayant qu'une seule requête, cette précision moyenne vaudra donc 0,6.

Cette mesure met en évidence la capacité du système à extraire les informations pertinentes des phrases. Cependant, elle est sensible à la qualité de la base de données dans laquelle les recherches sont effectuées. En effet, si les phrases ne contiennent aucune information pertinente permettant leur classification en différents sujets, les résultats seront mauvais, et cela indépendamment des performances intrinsèques du système d'extraction.

C'est pour cette raison que l'on compare cette mesure, lorsque c'est applicable, avec la précision moyenne obtenue quand on se base sur la transcription textuelle exacte des phrases et des mots clés. Car même une transcription phonétique correcte ne conduit pas toujours à une bonne répartition en sujets.

Cette mesure peut également mettre en évidence l'importance des modèles de langages vis-à-vis des modèles phonétiques. En effet, en comparant la précision moyenne obtenue par la transcription textuelle avec celle obtenue par la transcription phonétique, on peut aisément déterminer la perte de qualité résultant de l'utilisation d'un modèle phonétique par rapport à l'utilisation des modèles de langages.

3.1.6 Indexation par le contenu

Cette dernière mesure, quoique très utile dans le cadre de tri de messages dans un nombre fini de catégories présente néanmoins un désavantage lorsqu'elle est appliquée à la détection de mots clés.

En effet, cette méthode d'évaluation est sensible à la fréquence d'apparition des mots clés. Si on demande à un même système d'extraction, de détecter 7 occurrences de mots clés parmi 120 ou parmi 1200, il est clair que les résultats de la requête, en terme de précision, vont être dégradés. Cependant, seule la tâche a été modifiée, et le système utilisé est resté inchangé.

Pour le tri de message en un nombre fini de catégories, la mesure reste insensible à la taille de la base de données car les fréquences de chaque sujet restent stables.

Pour avoir une mesure à la fois indépendante de ce phénomène et spécifique à l'indexation, nous avons introduit un critère d'évaluation original fondé sur le temps nécessaire à un utilisateur pour effectuer une recherche spécifique avec ou sans outil d'indexation.

Nous définissons, pour une requête donnée, le *gain de temps* comme le rapport entre le nombre de phrases que l'utilisateur ne doit pas écouter en utilisant l'outil d'indexation, et le nombre de phrases qu'il devrait écouter s'il ne possédait pas l'outil.

En considérant l'exemple cité plus haut, nous pouvons aisément estimer le temps nécessaire à une personne ne possédant pas d'outil d'indexation, pour trouver 5 occurrences d'un mot clé dans les 120 phrases (hors de 7 occurrences totales). En moyenne, il devra écouter 88 phrases. En utilisant l'outil dont les résultats seraient comme indiqués dans le tableau (1), il ne devrait écouter que 8 phrases, D'où une économie de 80 phrases et un gain de temps de $\frac{80}{88}$%.

Pour la totalité des mots clés, estimons le temps gagné par la personne utilisant cet outil d'indexation et regroupons ces résultats au tableau (4), suivant le nombre d'occurrences de mots clés qu'elle recherche.

Indexation	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
Sans	18	35.5	53	70.5	88	105.5	123	70
Avec	1	3	4	7	8	9	88	17.14
Economie	17	32.5	49	63.5	80	96.5	35	49.63
Gain	94.5	91.5	92.5	90	90.9	65.9	28.5	79.11

TABLEAU 4. **Gain de temps de l'utilisateur en fonction du nombre de mots clés recherchés (approche neuronale).**

Nous pouvons alors définir le *gain de temps moyen* comme la moyenne des gains par rapport aux nombres d'occurrences recherchés dans la base de données.

3.1.7 Lien entre "courbe caractéristique" et "position"

A partir des mesure de position d'occurrence, il est possible d'estimer la courbe caractéristique du récepteur. pour chaque position d'occurrence, nous pouvons estimer un point de la courbe.

En effet, comme le montre le tableau (5), nous pouvons, pour chaque nouvelle occurrence du

	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ
Position	2	4	8	13	21	31	57
Proba. détection	1/7	2/7	3/7	4/7	5/7	6/7	7/7
Fausses Alarmes	1	2	5	9	16	25	50

TABLEAU 5. **Lien entre fausses alarmes et position d'occurrence.**

mot clé, obtenir le nombre de fausse alarme à partir des positions, car le nombre de fausses alarmes est simplement la différence entre le nombre de phrases sélectionnées (position) et le nombre de phrases contenant réellement le mot clé (nombre d'occurrence jusqu'à l'instant précis). De plus, connaissant, a posteriori, le nombre total d'occurrences du mot clé, nous pouvons estimer, pour chaque occurrence, la probabilité de détection associé (la première occurrence correspond à une détection de 1/7, la seconde à 2/7, etc.).

Il est alors aisé de reconstruire la courbe comme le montre la figure 33.

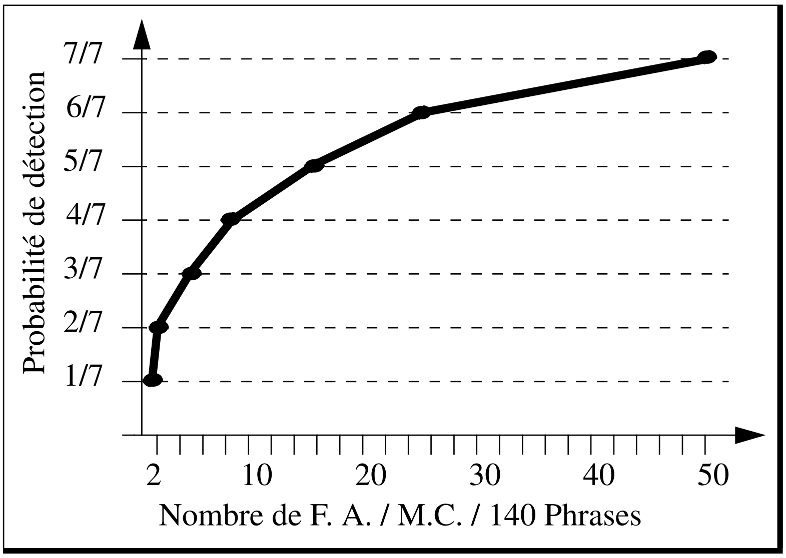

FIGURE 33. **Courbe caractéristique reconstruite à partir des positions**

3.2 La recherche de mots clés

3.2.1 Raison d'être

Dans son ensemble, la recherche de mots clés n'est généralement pas plus rapide que la reconnaissance de mots enchaînés.

Pourquoi privilégions-nous l'utilisation d'un détecteur de mots clés?

Comme le fait remarquer Higgins, [HIGG85], les raisons sont multiples :

- *L'entraînement du locuteur :* La reconnaissance de mots enchaînés est souvent appliquée à un individu en particulier et dans ce cas les modélisations utilisées lui sont propres. Il est cependant vrai que les améliorations apportées ces dernières années aux systèmes de reconnaissance vis-à-vis de leurs indépendance par rapport aux locuteurs sont remarquables.
- *Le vocabulaire :* La reconnaissance de mots enchaînés implique que chaque mot prononcé appartienne à un vocabulaire fixé, et qu'au moins un échantillon de chaque mot soit disponible pour l'apprentissage. Même si l'usage aujourd'hui très généralisé de la représentation des mots par leur transcription phonétique permet l'insertion de nouveaux mots de vocabulaire, la reconnaissance de mots clés garde l'avantage dans le cas de la parole spontanée.
- *La syntaxe :* La reconnaissance de mots enchaînés implique que la séquence de mots contenue dans la phrase à analyser puisse être générée par un automate statistique à états finis. En présence de grands vocabulaires, cette contrainte est indispensable pour obtenir des taux de reconnaissances acceptables, mais exclut la parole spontanée qui échappe par définition à toute modélisation syntaxique.
- *L'attitude du locuteur :* Lors d'une reconnaissance de mots enchaînés, le locuteur sait généralement qu'il parle à une machine, tandis que la reconnaissance de mots clés est souvent appliquée à des locuteurs ne le sachant pas et parlant donc de manière spontanée.

3.2.2 Applications

La nécessité d'un outil permettant la détection de mots clés a été mise en évidence en grande partie par J. G. Wilpon, L. R. Rabiner, C-H. Lee et E. R. Goldman, [WIL90], qui firent une étude détaillée sur la reconnaissance de mots isolés appliquée à la téléphonie. La base de données sur laquelle ils travaillèrent fut construite par AT&T en 1988 et contenait approximativement 70000 appels où les personnes étaient sensées dire un des 5 mots pré-définis ("Collect", "Calling-card", "Third-number", "Person", "Operator") de façon à obtenir le service désiré. L'étude montrait clairement que bon nombre de personnes (17%) ne citaient pas uniquement un des cinq mots clés, mais incluaient également dans leur message d'autres mots ("*Collect call* please"). Par ailleurs, ils montraient, [BOSS88], que le taux de reconnaissance par mot isolé chutait de 97% à 90% lorsque ces paroles parasites étaient présentes. Pour remédier à ce problème, ils proposèrent d'utiliser un modèle poubelle qui permit d'améliorer ces taux respectivement à 99.3% et 95.1%.

Cependant, la reconnaissance de mots clés peut être utile dans un grand nombre d'applications. R.C. Rose, [ROSE91], proposa en 1991 d'utiliser la reconnaissance de mots clés pour le tri des messages vocaux en 6 classes différentes en fonction de leur contenu. En se basant sur 120 mots clés, il obtenait un taux moyen de détection de 69% avec un taux de fausses alarmes de 5.4 par mot clé et par heure. Ces détections subissaient un post-traitement en fonction du taux de reconnaissance de chaque mot clé. En associant chacune de ces 6 classes avec 20 mots clés, il parvenait à classifier les phrases, d'une durée moyenne de 30 secondes, avec un taux de réussite moyen de 82.4%.

S. Nakamura, [NAKA93], proposa pour sa part, l'utilisation de la reconnaissance de mots-clés, pour la numérotation téléphonique dans les GSMs, chaque nom propre correspondant à un numéro à composer. Les contraintes étaient fortes, car le système de reconnaissance devait pouvoir fonctionner sur le DSP du téléphone et dans des environnements très bruités (voiture), mais il pouvait cependant être dépendant du locuteur. Il utilisait une méthode de pré-segmentation basée sur la puissance acoustique, puis appliquait un algorithme de programmation dynamique pour comparer les mots-clés. Les résultats obtenus étaient de l'ordre de 90% de détections correctes pour un vocabulaire de 100 mots.

3.2.3 Méthodes existantes

Depuis les années 70, bien des progrès et innovations ont contribué à l'amélioration des systèmes de recherche. La liste présentée ici, est loin d'être exhaustive, mais elle montre la richesse des idées déployées dans ce domaine en seulement une vingtaine d'années.

En 1977, R.W. Christiansen et C.K. Rushforth, [CHRIS77], utilisèrent une méthode dépendante du locuteur, basée sur la *mise en correspondance entre un flux de parole continue et des séquences de référence* (“reference templates”) pour en extraire les mots clés. Chaque mot de référence était comparé avec une fenêtre de taille identique, glissant le long du flux d'entrée. Un algorithme d'alignement temporel (“DTW”) était utilisé pour déterminer la distance entre cette fenêtre et le mot de référence. Un mot clé était détecté lorsque la distance enregistrée était suffisamment faible pour plusieurs fenêtres successives. Il est évident que cette méthode de fenêtre glissante nécessitait un grand nombre de calculs. En outre, l'utilisation de multiples occurrences de mots de référence pour augmenter la robustesse alourdissait encore le procédé d'apprentissage. Pour alléger la charge de calculs, les auteurs utilisaient à chaque étape de décision une méthode de seuillage empirique.

En 1985, A.L. Higgins et R.E. Wohlford, [HIGG85], introduisirent le concept de “*mot poubelle*” (“filler templates”). Ils partirent d'un système de reconnaissance continu basé sur l'alignement temporel pour détecter 25 mots clés dans de la parole générés à partir d'un vocabulaire fermé de 100 mots. Tout d'abord, ils utilisèrent les 100 mots du vocabulaire en tant que mots de référence pour effectuer l'alignement temporel, extraire la séquence de mots prononcés et détecter ainsi le passage par les mots clés. Ensuite, ils introduisirent l'utilisation de mots poubelles pour modéliser les 75 non mots clés. Ces mots poubelles étaient soit composés des mots dits de liaison ("the", "of", "for", etc...) soit composés de l'ensemble des phonèmes basés sur une segmentation manuelle de non mots clés (61 mots poubelles), soit estimés par quantification vectorielle sur les données utilisées par les deux modélisation précédentes (15 ou 7 classes étant générées).

En 1989, J.G. Wilpon, C.H. Lee et L. R. Rabiner, [WIL89], utilisèrent une *approche markovienne* pour détecter les mots clés émis de façon quasi isolée lors d'appels téléphoniques. La position du mot clé était basée sur le même principe que celui de Christiansen et Rushforth. Imaginant toutes les paires de début et de fin possibles du mot clé, on effectuait un alignement temporel, mais cette fois-ci en s'appuyant sur le modèle markovien du mot clé recherché. Un post-traitement basé sur la durée moyenne du mot clé, ainsi que sur l'énergie du signal était ensuite effectué de façon à réduire le taux d'erreurs.

En 1990, ces mêmes auteurs, [WIL90], introduisirent le concept de mot poubelle dans la modélisation markovienne et *évitèrent* ainsi *le parcours exhaustif* de toutes les paires point d'entrée-point de sortie des mots clés, tout en améliorant le taux de reconnaissance. Le mot poubelle était ici entraîné sur toutes les sections ne contenant pas de mots clés, et était composé d'une dizaine d'états interconnectés.

La même année, R. C. Rose et D. B. Paul, [ROSE90], firent une *étude approfondie sur l'utilisation des mots poubelles* pour la reconnaissance de mots clés. La base de données d'entraînement, issue de textes lus par 15 personnes différentes, était composée d'un vocabulaire de 100 mots, y compris 20 mots clés qui apparaissaient, en moyenne 59 fois. Le test, était pour sa part, composé d'extraits de conversations entre 8 personnes prononçant au total 353 mots clés. Pour construire les modèles de mots clés, ils enchaînaient des triphones, chacun composé de trois états successifs préalablement entraînés sur la base de données. Ils purent ainsi étudier différents types de mots poubelles. Leur première approche consista à créer le modèle poubelle à partir des 80 mots composant le reste du vocabulaire, chaque mot étant représenté par sa succession de triphones. La valeur de mérite, ("FOM"), associée à ce test fut de 59. Ils la comparèrent ensuite à un modèle poubelle essentiellement composé des triphones qui n'intervenaient pas dans la composition des mots clés (268 triphones, 804 états). Les résultats atteignirent 61. Ensuite ils essayèrent les modèles de phonèmes à trois états de manière à réduire la complexité du système (45 phonèmes, 135 états). Ils obtinrent un résultat de 60,6. Ils simplifièrent encore plus le système en utilisant des modèles d'états générés à partir de quantification vectorielle (128 états), mais les résultats chutèrent à 41,4. Finalement, ils essayèrent de détecter les mots clés en partant d'un entraînement sur une base de données indépendante de la tâche. Les mots clés, composés de triphones étaient mis en concurrence avec le modèle poubelle composé de phonèmes. Le taux de détection moyen fut dans ce cas de 44. La figure 34, extraite de [ROSE90] représente la syntaxe utilisé.

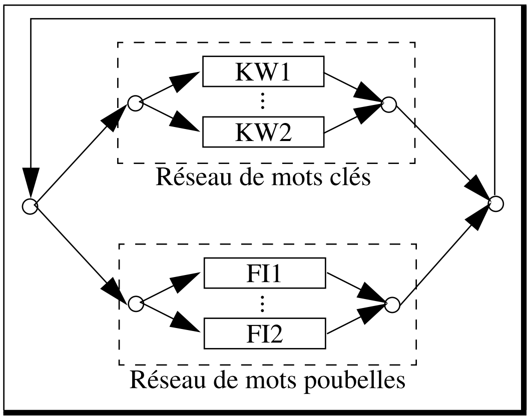

FIGURE 34. **Description du système de détection de mots clés basé sur l'utilisation de mots poubelles.**

En 1991, D. P. Morgan, C. L. Scofield et J. E. Adcock, [MORG91], introduisirent l'utilisation des *réseaux de neurones* dans la recherche de mots clés. Ils se basèrent tout d'abord sur un système standard de détection de mots clés pour obtenir les régions susceptibles de contenir un mot clé. Ensuite, ils utilisèrent un réseau de neurones pour valider ou non la détection. Pour ce faire, ils plaçaient à l'entrée du réseau un vecteur de taille fixe représentant la région où le mot clé était susceptible d'apparaître. Ce vecteur contenait les variations basses fréquences de chaque coefficient acoustique le long de cette région.

L'année suivante, R. C. Rose, [ROSE92], introduisit une méthode d'*entraînement discriminante* pour mieux séparer les modèles de mots clés vis-à-vis des modèles de non mots clés. Ce système était basé sur un ensemble discret de distributions gaussiennes. Chaque état du mot clé était représenté par une pondération de ces distributions. Il proposa une méthode de maximisation globale de la probabilité de détection du mot clé en modifiant les pondérations des différents états du mot clé. De plus, il utilisa un processus de retour prématuré de l'algorithme de programmation dynamique, de manière à déceler le mot clé avant d'avoir parcouru la totalité de la phrase.

Dans un esprit identique à ce dernier, Y. Komori et D. Rainton, [KOMO92], utilisèrent une *méthode du gradient* pour minimiser les erreurs de détection. Ici cependant, tous les paramètres des modèles markoviens étaient pris en compte.

G. J. Clary et J. H. Hansen, [CLAR92], s'appuyèrent sur un réseau de neurones à couche unique pour combiner les vecteurs acoustiques successifs afin de créer un nouveau vecteur représentatif qui était par la suite utilisé dans un modèle de Markov semi-continu. Le nombre de vecteurs combinés à l'entrée du réseau était variable en fonction du type de signal. La taille de la combinaison augmentait quand les coefficients acoustiques étaient stables et diminuait quand ces coefficients variaient rapidement. Pour un taille de combinaison fixée, ils regroupaient l'ensemble de vecteurs utilisant ce type de combinaison et entraînaient le réseau de neurones de manière à fournir pour chaque combinaison un vecteur unique représentatif.

H. Gish, K. Ng et J. R. Rohlicek, [GISH92] et [GISH93], retravaillèrent la décision de présence ou non de mots clés en utilisant la sortie d'un *décodeur acoustico-phonétique pour segmenter* le signal en phonèmes. Pour chaque segment phonétique, ils modélisaient la trajectoire des coefficients acoustiques contenus dans ce segment, de façon à obtenir un nombre fixe de coefficients caractérisant la trajectoire pour chaque phonème. Ces nouveaux coefficients étaient ensuite utilisés par un détecteur de mots-clés basé sur une approche Markovienne. Pour chaque mot clé, ils modélisaient le mot clé ainsi que le non-mot clé correspondant de façon à minimiser le taux de fausses alarmes.

En 1992 T. Zeppenfeld et A. H. Waibel, [ZEPP92], combinèrent l'utilisation d'un *réseau de neurones temporel* (TDNN) avec les méthodes classiques de programmation dynamique pour la recherche de mots clés. Ils utilisèrent un réseau de neurones spécifique pour chaque mot clé qui était représenté par un nombre fixe d'états correspondant aux sorties du réseau. Ces sorties étaient utilisées dans l'algorithme de programmation dynamique pour détecter les occurrences. Cependant, ils durent considérer chaque trame comme point de départ possible du mot clé. De ce fait, ils rejoignaient le principe utilisé par J. Wilpon en 1989, [WIL89]. En 1993, [ZEPP93], ils améliorèrent leur système de reconnaissance en utilisant, pour les petits mots clés, les contextes gauche et droit les plus communs, ceci ayant pour effet d'améliorer le taux de détection par rapport aux fausses alarmes. De même, ils utilisèrent une loi de Poisson pour adapter les courbes de probabilité de détection en fonction de la longueur moyenne des mots clés.

En 1993, R.C. Rose et E. M. Hofstetter, [ROSE93a], proposèrent un système de détection de mots clés qui se voulait indépendant de la tâche. Dans ce but, ils se basaient sur les triphones et utilisaient un *arbre de décision* pour réduire le nombre de classes d'apprentissage. Pour le modèle poubelle, ils utilisèrent le même arbre de décision, mais à une profondeur moindre de façon à rendre le modèle plus général que les modèles de mots clés.

J. R. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, M. Siu, [ROH93], travaillèrent sur la détection de mots clés en utilisant un entraînement phonétique basé sur les triphones, chacun modélisé par 3 états associés à une distribution discrète de probabilités d'émission. L'entraînement fut effectué séparément pour les hommes et les femmes à l'aide d'un *détecteur de sexe*. En se basant sur un modèle de langage composé de l'ensemble des mots du vocabulaire (2024 mots) représentés en séquences de triphones et reliés par une grammaire de type bi-gramme (perplexité de 42), les valeurs de mérite obtenues pour la détection de 20 mots clés prises dans ce vocabulaire étaient respectivement de 70.2 pour les femmes, 79.6 pour les hommes et 74.6 pour le modèle issu du mélange.

J. Alvarez-Cercadillo et L. A. Hernandez-Gomez, [ALVA93], présentèrent un système de reconnaissance de mots clés construit en deux parties. La première partie consistait en un détecteur de phonèmes utilisant une approche markovienne comprenant une grammaire nulle (chaque sortie de phonème étant relié à toutes les entrées des phonèmes). Ce détecteur fournissait à chaque instant une probabilité associée à chaque phonème. Dans la deuxième partie, ces différentes probabilités étaient alors présentées à l'entrée de *réseaux de neurones récurrents*. Chaque réseau avait pour but de simuler l'exécution d'un automate à états finis représentant un mot clé spécifique. Pour ce faire, il était constitué d'une couche contextuelle permettant le phénomène de récurrence.

J.-M. Boite, H. Bourlard et B. D’Hoore, [BOIT93], pour réduire la charge de calcul inhérente aux mots poubelles classiques, utilisèrent un modèle de *mot poubelle dynamique*. En se basant sur une approche phonétique (avec contexte ou non), ils construisirent les mots clés en enchaînant les phonèmes, et ils composèrent un état dynamique modélisant le mot poubelle dont la probabilité d’émission était la moyenne de celles des N meilleurs états à l’instant considéré. Ils permettaient ainsi à l’état dynamique de modéliser tout le vocabulaire, mais avec une qualité moindre que le modèle de mot clé.

En 1994, R. P. Lippmann, E. I. Chang et C. R. Jankowski, [LIPP94] mirent au point une méthode d’*optimisation* de leur système de reconnaissance de mots clés *basé directement sur la valeur de mérite* (“Figure Of Merit”,”F.O.M.”).Lors de l’entraînement ils considèrent les occurrences supposées des mots, les trient suivant leurs scores et en détermine la valeur de mérite. L’ajout d’une occurrence correcte augmente cette valeur de mérite. Traçant cette augmentation en fonction du score de l’occurrence, on obtient une courbe des gradients en fonction des scores, comme schématise à la figure 35. Une même raisonnement peut être tenu pour les occurrences erronées. Utilisant ces gradients, ils pouvaient ainsi pondérer l’apprentissage de chaque occurrence de mots clés de manière à favoriser les occurrences de mots clés pouvant générer une variation plus forte de valeur de mérite, et inversement.

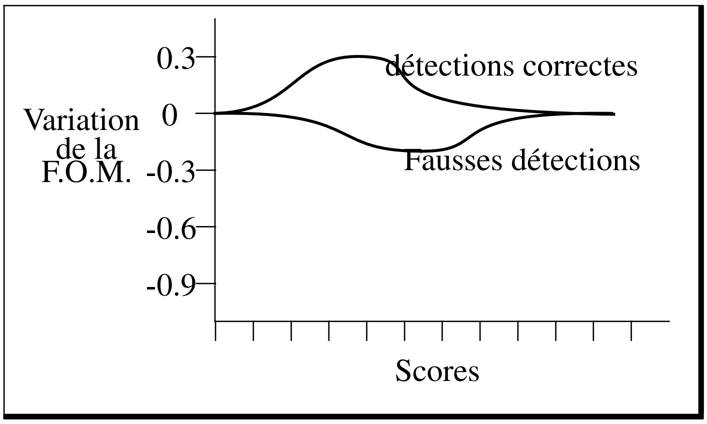

FIGURE 35. Estimation des gradients de la valeur de mérite.

En 1995, R. El Meliani et D. O’Shaughnessy, [ELM95], se basant sur le fait que la différence entre un mot clé et un mot hors vocabulaire était uniquement un problème lexical et non phonétique, proposèrent un *modèle poubelle*, non pas au niveau phonétique, mais au *niveau lexical*. Pour construire leur modèle poubelle, ils regroupèrent les syllabes ayant des fréquences d’apparition semblables, construisant ainsi un modèle moins robuste que les modèles de mots clés, tout en restant basé sur les mêmes modèles acoustiques. Les avantages d’un tel procédé étaient important, car d’une part, il permettait un traitement plus rapide (pas de modèle acoustique propre au mot poubelle), et d’autre part, il permettait aussi d’effectuer un entraînement des modèles acoustiques indépendamment de la tâche à effectuer.

En 1996, E. I. Chang et R. P. Lippmann, [CHAN96], confrontés au manque de données pour l’apprentissage des modèles de mots, *augmentèrent artificiellement leur base de données* de façon à effectuer un apprentissage plus efficace. En se basant sur l'étude effectuée par Wakita, [WAKK77], sur la relation entre la variation du conduit vocal et la position des formants, ils déduisirent qu'une compression ou un élargissement linéaire de l'enveloppe spectrale à court terme permettaient de modéliser une variation inter-locuteur. En appliquant cette transformation à leurs données, ils multiplièrent leurs données par cinq, et augmentèrent leur valeur de mérite de 18 pourcents.

3.3 Identification de sujets

Au début des années 1990, une nouvelle utilisation des systèmes de reconnaissance de mots clés fit son apparition. Avec l'avènement du multimédia, la diffusion et le stockage du son et des images disposèrent de supports numériques permettant des traitements de plus en plus sophistiqués. Les utilisateurs furent submergés par la masse d'information et la notion de navigateur de bases de données s'imposa. Voici quelques escales dans cette croisière vers l'indexation automatique.

3.3.1 Lincoln Laboratory

Le premier laboratoire à se préoccuper de ce problème fut sans doute le Lincoln Laboratory du MIT. R.C. Rose, E. I. Chang et R.P.Lippman, [ROSE91] présentèrent un procédé de classification de messages vocaux en fonction de leur contenu. Cette méthode fut inspirée des travaux de A.L. Gorin, S.E. Levinson. L.G. Miller et A.N. Gertner effectués en 1990 aux Bell Labs, [GOR90]. Le système de reconnaissance de mots clés sur lequel ils se basèrent était un modèle identique à celui qu'ils avaient utilisé en 1990, [ROSE90]. Partant de 510 messages de durée moyenne de 30 secondes, ils tentèrent de les répartir en 6 classes différentes ("description de jouets", "description d'objets abstraits", "discussion générale", "lecture de carte", "Interprétation d'une photo" et "description d'un dessin animé"). Pour ce faire ils choisirent pour chaque classe 40 mots clés contenant l'information mutuelle maximale.

Rappelons que si l'on note par C_i les différentes classes, M un message composé de K mots w supposés indépendants et pris hors d'un vocabulaire V, alors l'*information mutuelle* entre un mot w_k et une classe C_i est donnée par :

$$I(C_i, w_k) = \log\frac{P(C_i, w_k)}{P(C_i)P(w_k)} = \log P(C_i|w_k) - \log P(C_i).$$

Ils définissaient alors le facteur de détection :

$$s_k(M) = \begin{cases} 1 \text{ si } w_k \text{ détecté dans } M \\ 0 \text{ sinon} \end{cases},$$

et $v_{k,i} = I(C_i, w_k) + \log P(C_i) = \log P(C_i|w_k)$, de façon à pouvoir écrire la sortie du classificateur $c_i = \sum_{k=1}^{K} v_{k,i} s_k(M)$.

Le taux de classifications correctes atteignait une moyenne de 82%, alors que le taux de détection de mots clés était de 69% avec un taux de 5.4 fausses alarmes par mot clé et par heure.

Pour améliorer ces performances, ils modifièrent le facteur de détection, $s_k(M)$, pour tenir compte du score, y_k, obtenu lors de la détection de chaque mot clé par le système :

$$s_k(M) = \frac{1}{1 + exp(u_{k,2} - u_{k,1} y_k)},$$

où les paramètres $u_{k,1}$ et $u_{k,2}$ étaient estimés par rétro-propagation de l'erreur de classification :

$$E = \frac{1}{2} \sum_{i=1}^{6} (d_i - \hat{c}_i)^2,$$

où $d_i = 1$ pour la classe correcte et $d_i = 0$ pour les autres classes.

D'après leurs estimations, cette modification du système permit d'améliorer le taux de classifications correctes de 25%.

Une des contraintes de ce système était la nécessité d'entraîner des mots clés spécifiques. Il imposait donc la connaissance d'une cinquantaine d'occurrences de chaque mot clé pour pouvoir le modéliser correctement, et ainsi être capable de le détecter.

E. M. Hofstetter et R. C. Rose, [HOFS92], apportèrent en 1992 quelques modifications au système pour le rendre plus dépendant des tâches qu il pouvait effectuer. Pour ce faire, en partant d'un entraînement totalement indépendant du locuteur et de la tâche, ils modifièrent progressivement le système pour le rendre dépendant de la tâche, puis dépendant du locuteur.

Chaque triphone était classiquement représenté par 3 états markoviens et chaque état était défini par une pondération de 128 distributions gaussiennes f_m :

$$b_i(x_t) = \sum_{m=1}^{128} b_{i,m} f_m(x_t).$$

L'utilisation de ces pondérations permettait aisément de passer d'un système indépendant de la tâche à un système dépendant.

Pour l'adaptation à la tâche spécifique, ils considérèrent chaque distribution comme une combinaison linéaire entre les distributions indépendantes et dépendantes de la tâche.

$$b_{j,m} = \lambda_{it} b_{j,m}^{it} + \lambda_{dt} b_{j,m}^{dt}.$$

La mise en commun des deux informations se révélait nécessaire car les modèles indépendants, quoique non optimaux pour la tâche envisagée, étaient suffisamment définis grâce à la grande taille de la base de données, tandis que les modèles dépendants de la tâches étaient construits à partir d'une base de données trop petite pour être fiable.

La détermination des λ_{it} et λ_{dt} s'appuyait empiriquement sur les résultats de reconnaissance obtenus. Cette modification augmenta la probabilité de détection à 73% pour le modèle hybride alors qu'elle atteignait 44% pour le modèle indépendant et 68% pour le modèle dépendant.

Pour l'adaptation à un locuteur précis, ils modifièrent les moyennes et variances associées aux 128 probabilités gaussiennes.

Une première approche consistait à d'abord estimer les probabilités $P_{m,k,t}$ d'utilisation de chaque distribution gaussienne f_m pour chaque état q_k et chaque vecteur acoustique x_t, à partir de la séquence phonétique de la base de données mono-locuteur. Ensuite, il suffisait de pondérer les nouveaux paramètres en tenant compte de ces probabilités. Par exemple, pour les moyennes, cela donnait :

$$\mu_m = \frac{\sum_{t=1}^{T} \sum_{k=1}^{K} P_{m,k,t} x_t}{\sum_{t=1}^{T} \sum_{k=1}^{K} P_{m,k,t}}$$

Cette méthode permettait de passer à une probabilité de détection de 69% pour le modèle mixte alors qu'elle atteignait 65% pour le modèle indépendant du locuteur et 57,8% pour le modèle entraîné sur la base mono-locuteur.

3.3.2 Dragon System

La société "Dragon System" se pencha aussi, en 1993, sur le problème de l'indexation de sujets, mais en partant d'un système de reconnaissance de parole continue sur large vocabulaire, [GILL93] [PESK93]. Cette approche lui permettait d'utiliser la séquence de mots obtenue par le système de reconnaissance pour classer les phrases.

Leur but était de classer correctement 120 phrases de 4.5 minutes parmi 10 sujets d'actualité ("pollution","musique","délinquance",...).

Chaque sujet, S_i, possédait son propre modèle markovien, M_i, construit à partir de mots clés représentatifs du sujet et d'un sous-modèle utilisé pour les autres mots. Pour la sélection des mots clés, ils choisirent 2 approches différentes.

La première reposait sur les hypothèse que chaque mot avait la même probabilité d'occurrence dans chaque sujet et que l'apparition de chacun de ces mots suivait une distribution binomiale. Ensuite, ils effectuèrent le test de χ^2 pour trier ces hypothèses en fonction de leur validité, et ils gardèrent les mots ayant le plus faible taux de validité. L'inconvénient de cette méthode était qu'elle conservait les mots de liaison ("the","of",...) qu'ils enlevèrent manuellement.

La seconde méthode utilisait le même schéma, mais chaque mot était préalablement trié suivant sa fréquence d'apparition ("rare","moyenne","fréquente") dans chaque phrase d'entraînement. Ensuite, le test de χ^2 était utilisé pour trier les mots suivant la validité de l'hypothèse que ces classes de fréquences étaient identiques pour chaque sujet.

Pour estimer le sujet S_i associé à une séquence acoustique test, X, les auteurs se basèrent sur le critère suivant :

$$i_{max} = \underset{i}{\operatorname{argmax}} P(M_i)P(X|M_i).$$

Or, si l'on injecte les séquences de mots W_j^i pouvant être générées par le modèle M_i, on peut écrire :

$$P(X|M_i) = \sum_j P(X, W_j^i \middle| M_i).$$

Pour réduire la charge de calculs, ils firent alors successivement l'hypothèse que la somme pouvait être approximée par la meilleure séquence obtenue avec le modèle associé au sujet donné :

$$P(X|M_i) = P(X, W_{max}^i \middle| M_i),$$

puis l'hypothèse que cette meilleure transcription pouvait être obtenue par un modèle générique :

$$\begin{aligned} P(X|M_i) &= P(X, W_{max}|M_i) \\ &= P(W_{max}|M_i)P(X|M_i, W_{max}) \end{aligned},$$

et enfin l'hypothèse que l'émission des vecteurs acoustiques, connaissant la séquence de mots associée, était indépendante du sujet :

$$P(X|M_i) = P(W_{max}|M_i)P(X|W_{max})$$

Les résultats obtenus étaient les suivants :

- en utilisant tous les mots du vocabulaire pour générer les modèles, ils atteignaient un taux de 72% de bonne classification pour les 120 phrases;
- lorsqu'ils utilisaient la première méthode de sélection des mots clés, ils obtenaient un taux de 67,5% avec 211 mots clés;
- la deuxième méthode de sélection conduisit à des taux de 70% avec 203 mots clés et de 74% avec 4600 mots clés.

3.3.3 Ensigma

La société "Ensigma Ltd." présenta en 1995 [CAR95], un système d'identification de sujets reposant sur l'utilisation de mots clés.

La génération de modèles de sous mots était basée sur l'emploi d'un arbre de décision utilisant des mesures de dissimilarité pour séparer les classes ayant la plus grande dispersion.

La construction des modèles représentant les mots clés utilisait les contextes gauche et droit de chaque mot clé pour obtenir une meilleure représentation. Chaque prononciation différente générait un modèle spécifique.

La détection de mots clés s'effectuait en regardant le cheminement dans le modèle de langage. Ce dernier était construit à partir de modèles phonétiques simples, représentant les non mots clés et les modèles contextuels représentant les mots clés. La pondération des transitions entre modèles phonétiques simples permettait de faire varier le taux de détection vis-à-vis du taux de fausses alarmes.

Le système de sélection des sujets utilisait la métrique suivante :

$$V_m = \sum_j n_j \log\frac{P(w_j|S_m)}{P(w_j|S_{\overline{m}})},$$

où n_j était le nombre d'occurrences du j ème mot clé dans la phrase à classer, $P(w_i|S_m)$ la probabilité, a priori, de détecter ce mot clé dans le m$^{\text{ème}}$ sujet et $P(w_i|S_{\overline{m}})$ la probabilité, a priori, de détecter ce mot clé dans les autres sujets.

Les tests étaient réalisés sur une base de données générée à partir des journaux parlés de la BBC contenant 15 heures de parole. Chaque sujet était composé de 10 mots clés choisis arbitrairement. Le taux de reconnaissance des modèles de sous mots était d'environ 30%, tandis que le taux moyen de reconnaissance de mots clés était de 46%.

Les détections correctes de sujet variaient comme indiqué à la figure 36.

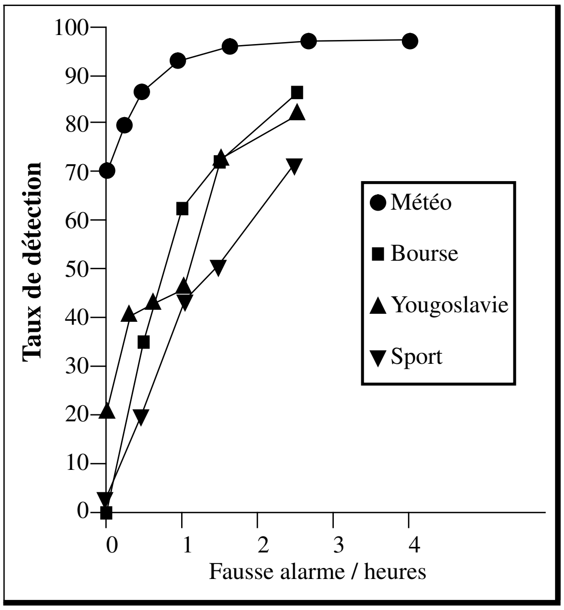

FIGURE 36. **Reconnaissance de sujets**

Le bon comportement de la détection du sujet "météo" était dû à la fréquence d'occurrence élevée des mots clés sélectionnés pour ce sujet.

3.3.4 Cambridge University

Le laboratoire de l'université de Cambridge (UK) a également consacré de nombreux travaux au tri de messages par leur contenu. En effet, en 1995, ils présentèrent un article sur ce sujet, [FOOT95] [JON95]. Leur tâche était de mettre au point un système de recherche d'informations à partir du courier vidéo ("video mail") installé dans leur laboratoire. Pour ce faire, ils développèrent un détecteur de mots clés "indépendant" du locuteur.

La base de données récoltée fut construite comme suit : 10 sujets étaient définis a priori. Pour un sujet donné, on construisait 5 descriptions de situations auxquelles les personnes devaient réagir en envoyant un message, et on enregistrait 6 réponses émanant de 6 personnes choisies au hasard parmi 15 personnes. Ceci totalisait donc 300 messages répartis parmi 10 sujets, décrits par 5 situations différentes. Pour chaque sujet, on choisit arbitrairement 35 mots clés définissant au mieux ce sujet.

Le système utilisait des modèles triphoniques de mots clés mis en concurrence avec un modèle monophonique de langage. Un premier usage de l'algorithme de Viterbi sur la phrase analysée à l'aide du modèle monophonique permettait d'obtenir les scores, trame par trame. Ces scores étaient utilisés pour normaliser les résultats obtenus lors de l'utilisation d'un second algorithme de Viterbi, basé sur les modèles triphoniques de mots clés confrontés d'une part à un modèle de silence et d'autre part aux modèles monophoniques.

La base de données récoltée par le système de courrier ne comportait que 300 messages de parole issus de 15 personnes différentes, l'apprentissage des modèles de triphones utilisait la base de données du "Wall Street Journal". Une adaptation aux locuteurs fut utilisée pour augmenter les taux de reconnaissance. Elle consistait en la prononciation, par chaque locuteur de 35 ou 75 phrases contenant les mots clés utilisés. Ces phrases permettaient de modifier les distributions gaussiennes associées aux probabilités d'émission, mais uniquement pour les modèles de triphones utilisés dans la construction des modèles de mots clés, tandis que le modèle de langage restait indépendant du locuteur.

Les taux de détection de mots clés, mesurés en "Valeur De Mérite" augmentèrent donc de 69, sans entraînement spécifique au locuteur, à 77 pour un modèle réestimé avec 35 phrases, à 79 lors de l'utilisation de 75 phrases et atteignait 81 lorsque les modèles employés étaient totalement dépendants du locuteur.

Pour mesurer l'efficacité du détecteur de mots clés pour l'extraction d'information, ils sélectionnèrent dans chaque description de situation les mots clés présents. Pour une description donnée, on utilisait alors ces mots clés pour effectuer une requête en espérant extraire des 300 phrases, uniquement les 6 phrases correspondant à cette situation.

Les mots clés associés à la requête, étaient pondérés pour tenir compte de leur fréquence d'apparition dans les 300 phrases.

La métrique utilisée pour trier les messages en fonction de la requête était quasi identique à celle utilisée par "Ensigma Ltd". En effet, la fréquence d'apparition des mots clés recherchés était pondérée par une mesure de la pertinence de chaque mot clé à définir le sujet.

La métrique associée à la phrase m était :

$$V_m = \sum_j n_{m,\,j} \log \frac{N}{n_j},$$

où N représentait le nombre total de phrases (300), n_j le nombre total de fois où le j$^{\text{ème}}$ mot clé apparaissait dans les toutes phrases, et $n_{m,\,j}$ le nombre de fois que ce mot clé était détecté dans la m$^{\text{ème}}$phase.

La *précision moyenne* du système était comparée aux résultats que l'on aurait obtenu en connaissant la transcription textuelle des phrases, et aux résultats que l'on aurait obtenu en connaissant la séquence phonétique exacte. Ces résultats sont repris dans le tableau (6).

	Précision moyenne	**Taux vis-à-vis texte**	**Taux vis-à-vis séq. phon. exacte**
Ind. Loc.	0.263	79.0%	82.9%
35 Phrases	0.271	82.2%	86.3%
75 Phrases	0.290	87.3%	92.9%
Dép. Loc.	0.295	88.8%	93.2%
Phonétique	0.317	95.3%	100%
Texte	0.332	100%	

TABLEAU 6. **1ers résultats Cambridge**

En 1996, [JON96] [BROW96] proposèrent une évolution consistant à utiliser le système de reconnaissance de parole continue HTK, entraîné sur un vocabulaire de 20 milles mots (WSJ). Malheureusement, le modèle de langage utilisé était construit à partir d'articles de journaux et ne correspondait pas à de la parole spontanée.

La comparaison des deux systèmes (mots clés et large vocabulaire) au niveau des détections de mots clés tourna au désavantage du système de reconnaissance de parole continue qui n'obtint que 53 comme valeur de mérite, par rapport à 69,9 pour le détecteur de mots clés. Cependant, les résultats obtenus restaient intéressants étant donnée la flexibilité apportée par ce système de reconnaissance de parole continue.

Pour estimer plus finement la qualité du système, ils modifièrent la création de requête pour représenter plus fidèlement la réalité. Ils demandèrent à 10 personnes de générer, pour chaque sujet 5 questions contenant au moins un des 35 mots clés prédéfinis. De ces questions, ils supprimèrent les mots de liaison ("the", "who", "in", "be", etc.) qui n'apportaient aucune information quant au sujet recherché.

Pour le tri, ils utilisèrent une métrique plus évoluée que précédemment, dans le but de rendre les mesures indépendantes de la longueur des phrases :

$$V_m^e = \sum_j n_{m,j} \frac{(K+1)}{(KL_m + n_{m,j})} \log \frac{N}{n_j},$$

où K était une constante déterminée empiriquement et L_m la longueur de la phrase, normalisée par la longueur moyenne des phrases.

Ils comparèrent alors les résultats obtenus avec le détecteur de mots-clés (MC) et ceux obtenus avec le système de reconnaissance de parole continue (PC). Ils en profitèrent par ailleurs pour combiner ces deux détecteurs pour améliorer les résultats.

Deux combinaisons possibles ont été étudiées. La première consistait à sommer les deux résultats obtenus pour chaque phrase puis à trier en fonction de cette somme (SOM). La seconde combinaison consistait à normaliser chaque liste de valeurs, pour un détecteur donné et une requête donnée, par la valeur maximale de cette liste, puis de les sommer phrase par phrase et enfin de les trier (NORM).

Les résultats obtenus sont explicités dans le tableau (7).

	Anciennes Requêtes		**Nouvelles Requêtes**	
	V_m	V_m^e	V_m	V_m^e
MC	0.249	0.287	0.284	0.309
PC	0.523	0.576	0.246	0.263
SOM	0.538	0.591	0.312	0.335
NORM	0.482	0.521	0.319	0.342

TABLEAU 7. **2ème résultats Cambridge**

Les valeurs importantes relevées pour les anciennes requêtes et l'utilisation du large vocabulaire s'expliquent aisément par la longueur moyenne de la requête, qui passait de 5.7 à 18.7 mots, quand on utilisait un large vocabulaire plutôt que les mots clés lors de la création des requêtes.

CHAPITRE 4

Solutions proposées

Nous avons jusqu'à présent décrit les outils utilisés en reconnaissance de parole ainsi que les méthodes précédemment développées en vue de la détection de mots clés.
Dans ce chapitre, nous précisons tout d'abord les contraintes imposées à la détection de mots clés par l'indexation de documents multimédia.
Par la suite, nous décrivons ***trois méthodes développées au cours de cette thèse*** *respectant ces contraintes spécifiques.*

- La première solution envisagée utilise une détection de phonèmes fondée sur un étiquetage trame par trame du signal à indexer.
- La deuxième méthode proposée s'appuie sur les chaînes de Markov pour extraire du signal les éléments nécessaires à l'élaboration de séquences phonétiques qui seront ultérieurement exploitées lors de l'indexation.
- La dernière technique tire parti du caractère discriminant de l'approche REMAP pour améliorer le système d'indexation.

4.1 Les contraintes de l'indexation

La détection de mots clés ne représente qu'un outil parmi d'autres pour réaliser l'indexation de documents multimédia. Comparé aux outils cités préalablement, il offre l'avantage de rester applicable dans de multiples configurations pour autant qu'il soit indépendant du locuteur, indépendant du vocabulaire et d'utilisation rapide.

4.1.1 Indépendance vis-à-vis du locuteur

Il va sans dire que dans le cadre d'une indexation de bande vidéo, l'utilisation d'un système multilocuteur s'impose.

On pourrait cependant imaginer, dans le cas de journaux télévisés par exemple, que certains locuteurs (les présentateurs réguliers) induisent un entraînement spécifique du système de reconnaissance, de façon à augmenter les taux de reconnaissance, qui sont bien sur plus faibles dans le cas multilocuteur que dans le cas monolocuteur. Cette dernière remarque doit cependant être nuancée par le fait que l'on possède généralement moins de données d'apprentissage pour des personnes particulières, que de données issues de locuteurs quelconques (voir à ce sujet la discussion sur le travail de E. M. Hofstetter et R. C. Rose [HOFS92]).

4.1.2 Indépendance du contenu lexical du signal à indexer

Si l'on désire que l'outil d'indexation puisse être appliqué sur des tâches les plus diverses possible et non sur des tâches précises contenant un vocabulaire restreint, nous devons élaborer un détecteur de mot clé pouvant fonctionner sur le plus grand vocabulaire possible.

Dans ce cas, deux solutions existantes sont envisageables.

La première consiste à utiliser un détecteur de mots clés dont les modèles associés aux mots clés concourent contre les modèles de poubelles. Effectivement, dans ce cas, le système est indépendant du vocabulaire utilisé et ne dépend que des mots clés recherchés.

La seconde solution consiste à utiliser un reconnaisseur de parole continue travaillant sur de grand vocabulaire. De grandes améliorations ont étés apportées sur ces systèmes qui obtiennent maintenant de bons résultats, même pour des vocabulaires de plusieurs dizaines de milliers de mots. De plus, on voit arriver depuis quelques années des systèmes à grand vocabulaire pouvant gérer les mots hors vocabulaire ("Out Of Vocabulary words","O.O.V."). Dans le cas ou un tel mot est présent, il est simplement classé en tant que mot hors vocabulaire. Dans le cas de reconnaissance de parole continue, vus la taille du vocabulaire, il n'est plus envisageable de travailler en modèle de mots, et on est obliger d'utiliser des modèles plus courts comme les modèles de triphones, de diphones ou de phonèmes. Le choix entre les différents modèles repose le plus souvent sur la taille de la base de données d'apprentissage et sur la complexité du système que l'on peut mettre en oeuvre.

4.1.3 Connaissance du mot clé à rechercher au moment même de la requête.

Dans le problème d'indexation, on peut différencier trois étapes importantes : La premier consiste à l'entraînement du modèle de parole qui sera utilisé lors de la reconnaissance, la seconde est le moment ou l'on réceptionne le signal de parole et la dernière étape est celle où on lance la requête d'indexation proprement dite.

D'autre part, nous sommes informés du mot clé à rechercher à des moments divers. Soit au moment de l'entraînement, soit au moment ou on rentre en possession de la bande sonore, soit au moment même de la requête. En fonction de ces trois cas, les solutions envisageables sont différentes.

Si nous connaissons le mot clé avant l'entraînement, nous pouvons utilisé un modèle de mot spécifique à ce mot clé, pour autant que nous ayons en notre possession un nombre suffisant d'occurrence pour entraîner ce mot. Dans ce cas, la détection standard par mot clé concourant avec des mots poubelles est suffisant. Nous pouvons également envisager un reconnaisseur de parole continue dont le vocabulaire contient le mot clé recherché.

Si nous ne connaissons pas le mot clé avant l'entraînement mais lorsque nous possédons la bande sonore et avant la requête, nous pouvons toujours effectuer la recherche au préalable, mais nous devons nous baser sur une description du mot clé en terme d'unités plus petites que le mot et qui peuvent ainsi être entraîné indépendamment du mot clé. Il est clair que cette contrainte nous conduira dans le cas de détecteur de mots clé à des résultats plus faible, mais nous acceptons ce prix contre la flexibilité offerte. Dans le cas de la reconnaissance de parole continue, il suffit d'ajouter dans le vocabulaire du langage, la transcription phonétique représentant le nouveau mot recherché. Cette modification de grammaire peut être rapide, et la qualité des résultats ne se trouvent pas modifiée.

Maintenant, si nous ne connaissons le mot clé qu'au moment même de la requête, nous ne pouvons pas nous permettre de relancer la reconnaissance de parole continue, qui est un processus long si l'on désire une bonne qualité. De même, la détection de mots clés demande un effort de calcul non négligeable qui implique une durée de traitement importante.

Pour conserver un outil capable d'indexer rapidement dès que l'on a choisi le mots clés, il est nécessaire de séparer la tâche d'indexation en deux parties. La première partie doit effectuer le maximum de travail possible tant que l'on ne connaît pas le mot clé et peut être effectué dès réception de la bande sonore. La deuxième partie est effectuée dès que l'on connaît le mot clé, et doit indexer le signal sonore en conséquence et le plus rapidement possible.

4.1.4 Solution

Nous nous sommes donc attachés à la création d'un système d'indexation rapide, indépendant du locuteur et de la tâche. Ce système est basé sur la recherche de mots clés inconnus avant la requête de l'utilisateur, en considérant que l'on disposait du signal sonore préalablement à la requête.

Pour atteindre ce but, la seule possibilité était de travailler à l'aide de modèles inférieurs aux mots, et de séparer la tâche en deux parties : la première consistait à pré-traiter le signal de parole, et la seconde, au moment de la requête, consistait à rechercher le mot clé sur le signal pré-traité.

Pour rejeter le maximum de calcul dans le pré-traitement, nous avons décidé de générer lors de ce pré-traitement un treillis d'hypothèse phonétique qui est utiliser lors de la requête pour trouver la séquence acoustique du mot clé désiré.

Nous avons développé trois méthodes différentes, caractérisées par la modélisation du signal et du langage utilisé :

- La première méthode consiste à analyser le signal de parole trame par trame.
- La seconde utilise un approche markovienne.
- La troisième utilise une approche basée sur l'algorithme REMAP.

4.2 Indexation par étiquetage de trames

4.2.1 Principe de la méthode

Le but de cette approche est de déterminer les performances que l'on peut obtenir en évitant l'utilisation de chaînes de Markov. Le système peut être scindé en plusieurs parties. La première partie estime la probabilité qu'un vecteur acoustique ait été généré lors de la prononciation d'un phonème. Cette recherche sera décrite à la section 4.2.2. La deuxième partie isole, en utilisant ces probabilités, les régions où certains phonèmes ont pu être prononcés et indexe à l'aide d'un treillis. Elle sera exposée en 4.2.3. Le principe de la recherche des mots clés, sera explicité en 4.2.4., et l'algorithme correspondant sera décrit en 4.2.5. Finalement, les expérimentations en vue de l'évaluation de la méthodes seront explicitées et comparées en section 4.2.6.

4.2.2 Probabilités locales

On associera à chaque vecteur acoustique $x \in X$, une seule probabilité $P(x|\varphi)$. Elle représente la probabilité que le vecteur acoustique x ait été émis lors de la prononciation d'un phonème $\varphi \in \phi$.

Cette approche peut être mise en parallèle avec une approche markovienne où l'on ne considérerait qu'un seul état par phonème. Cette probabilité mesure, comme les probabilités d'émission sur état, le degré d'appartenance d'un vecteur acoustique à une classe phonétique. Cependant, l'alignement temporel de type Viterbi n'étant pas utilisé dans ce système de reconnaissance, la comparaison avec l'approche markovienne s'arrête là.

Trois méthodes différentes sont utilisées pour estimer cette probabilité locale :

- la modélisation par une distribution monogaussienne pour chaque classe phonétique ;
- la modélisation multigaussienne ;
- l' utilisation d'un réseau de neurones pour estimer la probabilité a posteriori $P(\varphi|x)$.

Dans ce dernier cas, nous utilisons la loi de Bayes pour relier les probabilités conditionnelles et a posteriori.

4.2.2.1 Monogaussienne

Cette approche est fréquemment utilisée pour l'estimation des probabilités d'émission et consiste à évaluer :

$$P(x|\varphi) = (2\pi)^{-\frac{N_a}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_\varphi)\Sigma^{-1}(x-\mu_\varphi)^T},$$

où la matrice de covariance, Σ, est supposée diagonale pour les raisons évidentes de simplicité :

$$P(x|\varphi) = \prod_{n=1}^{N_a} \frac{1}{\sigma_{\varphi,n}\sqrt{2\pi}} e^{-\frac{(x_n-\mu_{\varphi,n})^2}{2\sigma_{\varphi,n}^2}}.$$

Il faut toutefois rappeler que cette approche, quoique courante, correspond à une grossière approximation de la probabilité par sa densité de probabilité. Une approximation plus correcte nécessite l'utilisation de la fonction :

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt,$$

et implique que l'on suppose le vecteur acoustique, non plus comme un point dans l'espace $\Re^{N_a}$, mais plutôt comme une région de cet espace.

Pour estimer l'ensemble des paramètres $\Lambda = (\mu_{\varphi_1}, \sigma_{\varphi_1}, \ldots, \mu_{\varphi_K}, \sigma_{\varphi_K})$, deux méthodes de segmentation phonétique sont envisagées. La première est obtenue par le travail d'experts qui détectent manuellement les frontières entre phonèmes. La seconde est issue d'un processus d'optimisation automatique, basée sur l'utilisation répétée d'un algorithme de Viterbi qui ne requiert, au départ, que la séquence de phonèmes présents et non la segmentation.

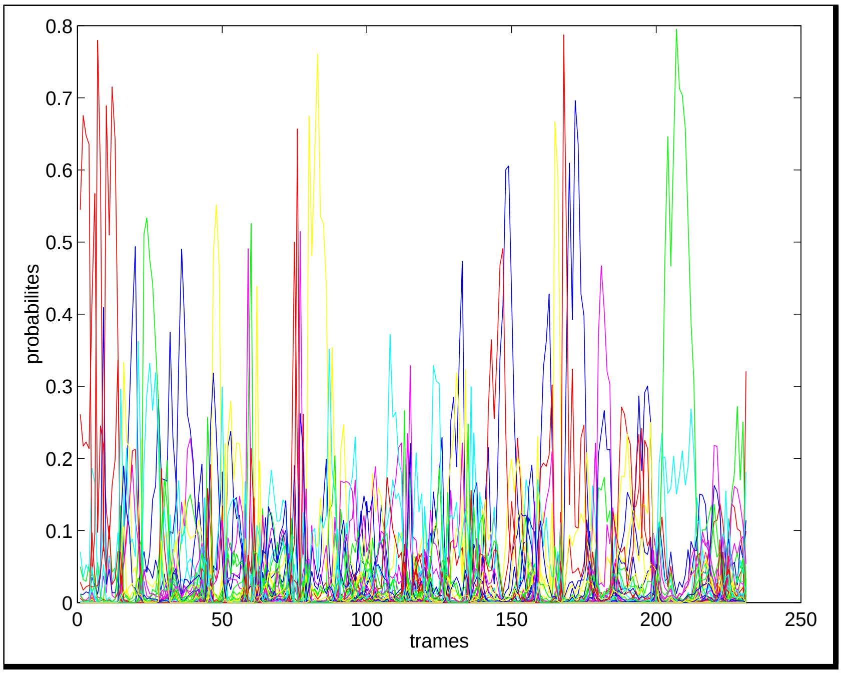

FIGURE 37. **Probabilité trame par trame, obtenue par modélisation monogaussienne, basée sur une segmentation manuelle de la base de données.**

4.2.2.2 Multigaussienne

Une autre approche, tout aussi classique, est de relaxer la contrainte d'unimodalité de l'hypersurface gaussienne associée à la densité de probabilité des différents phonèmes, au profit d'une utilisation de gaussiennes multimodales offrant l'avantage de modéliser plus fidèlement la distribution réelle des vecteurs acoustiques représentant le phonème. Dans ces conditions, la probabilité locale que le vecteur x ait été généré lors de la prononciation du phonème φ devient :

$$P(x|\varphi) = \sum_{d=1}^{D_\varphi} n_d P(x|G_{\varphi,d}), \tag{EQ 62}$$

où n_d est le facteur de pondération non négatif respectant la contrainte :

$$\sum_{d=1}^{D_\varphi} n_d = 1,$$

et où

$$P(x|G_{\varphi,d}) = \prod_{n=1}^{N_a} \frac{1}{\sigma_{\varphi,d,n}\sqrt{2\pi}} e^{-\frac{(x_n - \mu_{\varphi,d,n})^2}{2\sigma^2_{\varphi,d,n}}} \quad \text{(EQ 63)}$$

La remarque concernant l'approximation d'une probabilité par sa densité de probabilité reste de mise.

La détermination des paramètres de la multigaussienne s'effectue à l'aide de deux méthodes imbriquées, et ce pour chaque phonème pris séparément.

Un premier processus itératif rapide de classification est effectué de manière à associer chaque vecteur acoustique à la distribution gaussienne la plus probable :

$$x \in G_{\varphi,d} \text{ telle que } P(x|G_{\varphi,d}) \geq P(x|G_{\varphi,j}) \qquad ,\forall j, d\,.$$

Une fois tous les vecteurs acoustiques classés parmi les différentes distributions, on détermine les paramètres de chacune de ces distributions en prenant en compte, de manière identique, tous les vecteurs acoustiques associés à cette classe :

$$\mu^{new}_{\varphi,d,n} = \sum_{x \in G_{\varphi,d}} \frac{x_n}{N_{G_{\varphi,d}}} \text{ et } \sigma^{new}_{\varphi,d,n} = \sqrt{\sum_{x \in G_{\varphi,d}} \frac{(x_n - \mu_{\varphi,d,n})^2}{N_{G_{\varphi,d}}}}\,,$$

où $N_{G_{\varphi,d}}$ correspond au nombre de vecteurs associés à la distribution $G_{\varphi,d}$.
Cette approche rapide permet d'obtenir une première classification des vecteurs acoustiques. Cette approche n'est pas optimale, mais permet d'accélérer le processus optimal. En effet, elle fournit un point de départ nettement plus proche de la configuration finale qu'une répartition initiale aléatoire.

Un second processus itératif est alors utilisé pour affiner les distributions. On y considère que chaque vecteur acoustique contribue à l'élaboration de chaque distribution et ce, avec une pondération proportionnelle à sa probabilité d'appartenance. Cette répartition pondérée des vecteurs acoustiques permet de réévaluer les paramètres des différentes distributions et ainsi de considérer une nouvelle itération :

$$\mu^{new}_{\varphi,d,n} = \frac{\sum_{x \in G_{\varphi,d}} x_n P(x_n|G_{\varphi,d})}{\sum_{x \in G_{\varphi,d}} P(x_n|G_{\varphi,d})} \text{ et } \sigma^{new}_{\varphi,d,n} = \sqrt{\frac{\sum_{x \in G_{\varphi,d}} (x_n - \mu_{\varphi,d,n})^2 P(x_n|G_{\varphi,d})}{\sum_{x \in G_{\varphi,d}} P(x_n|G_{\varphi,d})}}\,.$$

Le nombre de distributions associées à chaque phonème est choisi en fonction du nombre de vecteurs associés au phonème. De plus, si le nombre de vecteurs acoustiques définissant une distribution descend en dessous d'un seuil, fixé à 30, cette distribution est supprimée.

4.2.2.3 Réseau de neurones

Comme expliqué précédemment, les réseaux de neurones peuvent être utilisés en classification et permettent d'obtenir une estimation des probabilités a posteriori $P(\varphi|x)$, [BOU88]. L'utilisation de la loi de Bayes, permet de comparer ces probabilités avec les deux méthodes précédentes :

$$P(x|\varphi) = P(\varphi|x)\frac{P(x)}{P(\varphi)},$$

où l'on considère $P(x)$ constante pour tout x et où l'on estime $P(\varphi)$ par dénombrement de la base d'entraînement.

Le réseau de neurones utilisé ici est un perceptron multicouches avec une seule couche cachée contenant 200 neurones. L'entraînement standard de rétropropagation de l'erreur est utilisé. En parallèle, nous évaluons de façon continue la qualité du réseau de neurones sur une partie de la base de d'entraînement non utilisée pour l'apprentissage, de manière à éviter le surentraînement ("validation croisée"). Les objectifs imposés à chaque vecteur acoustique sont binaires (1 pour la sortie représentant le phonème à associer au vecteur et 0 aux autres sorties). La segmentation choisie pour l'entraînement est identique à celle des deux autres méthodes.

Rappelons que l'avantage d'une approche neuronale consiste à ne pas imposer de contraintes au niveau de la forme de la distribution de probabilité. De plus, les probabilités a posteriori sont entraînées de manière discriminante.

4.2.3 Génération du treillis

Pour chacune des trois méthodes précédentes nous pouvons estimer la probabilité, à chaque instant, d'être en présence d'une prononciation du phonème φ, connaissant le vecteur acoustique émis à cet instant, x_t. Cette probabilité, $P(\varphi|x_t)$, n'est rien d'autre que la probabilité a posteriori qui peut être estimée, soit directement par la méthode neuronale, soit en passant par la loi de Bayes pour les deux autres méthodes. Ces probabilités fluctuent au cours du temps. Il existe des zones dans lesquelles ces probabilités sont particulièrement élevées (supérieures à la moyenne) et il est fort probable que ces zones correspondent à l'émission d'un phonème. Ces zones, une fois repérées, sont appelées *hypothèses*. Elles sont définies par les bornes délimitant ces zones, par la probabilité qui leur est associée ainsi que le phonème supposé émis.

4.2.3.1 Intégration temporelle

Lorsque l'on considère, pour un même phonème, l'évolution de la probabilité a posteriori $P(\varphi|x_t)$ au cours du temps, on remarque de fortes variations entre les valeurs de trames successives. Ces variations sont principalement dues au fait que l'on n'a pas tenu compte de la durée des phonèmes lors de la construction de cette courbe. Ceci met en évidence une différence majeure avec l'approche markovienne qui modélise, bon gré mal gré, la durée des phonèmes. Pour atténuer cet effet, nous filtrons ces courbes de probabilité suivant une fréquence de coupure dépendant de la durée moyenne de chaque phonème. Pour ce faire, nous effectuons une intégration temporelle des valeurs contenues dans un intervalle de longueur égale à la durée moyenne du phonème :

$$P_{int}(t, \varphi) = \frac{1}{d_\varphi} \sum_{n = t - \frac{d_\varphi}{2}}^{t + \frac{d_\varphi}{2}} P(\varphi|x_n), \qquad \text{(EQ 64)}$$

où d_φ représente la durée moyenne du phonème φ.

En effectuant un tel traitement, nous réduisons fortement les pics erratiques pour les phonèmes de durée moyenne élevée, tout en conservant la possibilité de détecter les phonèmes de courte durée. Ce filtrage offre en outre l'avantage de faciliter la détection des bornes de début et de fin de phonème.

Pour illustrer l'efficacité de cette méthode, nous traçons ici les probabilités non filtrées et filtrées issues des 50 premiers vecteurs acoustiques de la base de données pour l'approche multigaussienne.

En comparant les deux approches de la figure 38 , il est aisé de constater que les probabilités associées aux phonèmes “longs” sont effectivement moins sensibles aux différences entre deux vecteurs acoustiques successifs. Nous pouvons également remarquer que les probabilités associées aux phonèmes “courts” conservent un temps de réponse court. Observons par exemple le comportement du premier silence (“h#” entre la première et la 15ème trame) et du “t” aux environs de la 22ème trame.

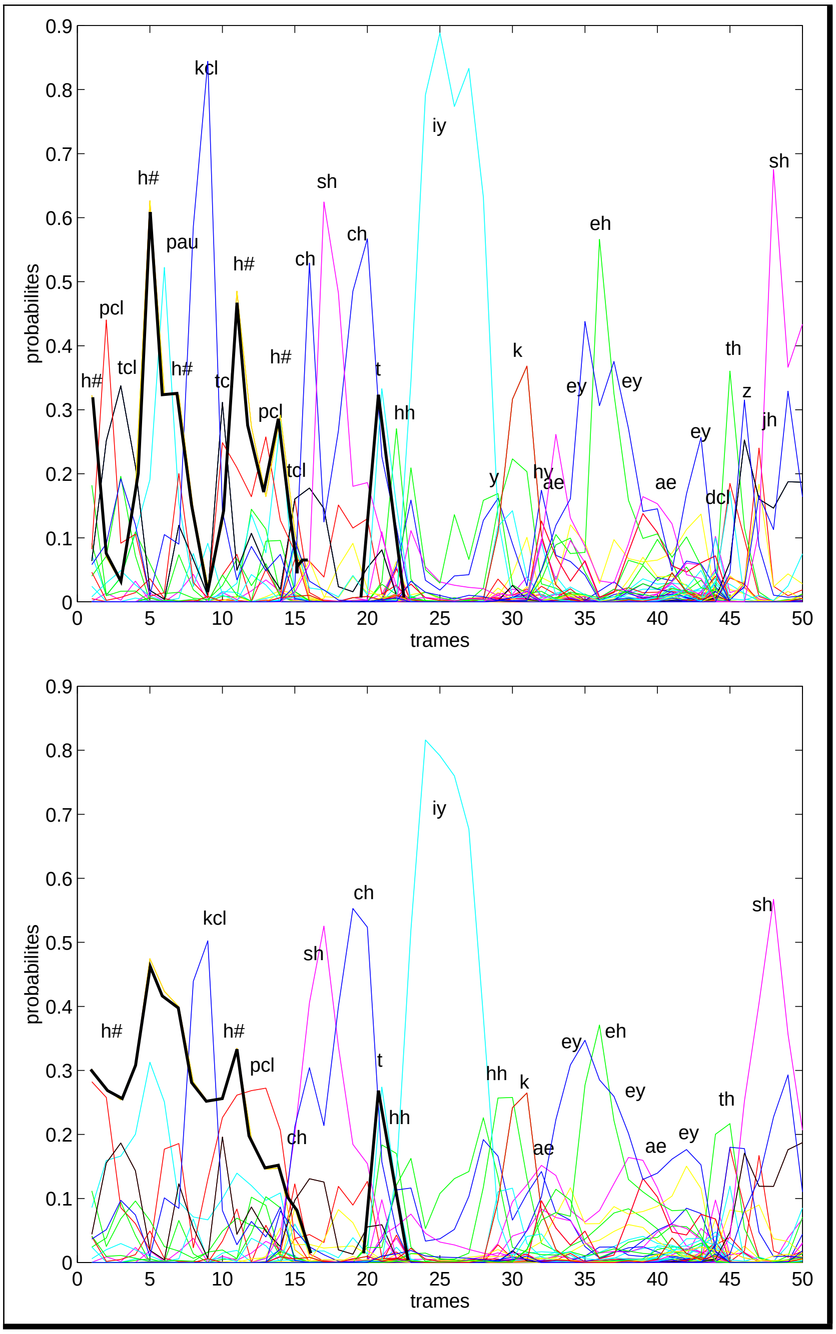

FIGURE 38. **Probabilités non filtrées (figure supérieure) et filtrées (figure inférieure), approche multigaussienne.**

4.2.3.2 *Niveau multiple d'hypothèses*

Comme on peut le constater sur les dernières figures, il n'est pas aisé de déterminer le début et la fin des hypothèses, ni de déterminer les transitions entre deux phonèmes successifs, surtout dans le cas d'une transition entre un phonème court et un long. Pour permettre plus de flexibilité vis-à-vis de la transition entre deux hypothèses, nous générons celles-ci à l'aide de différents seuils.

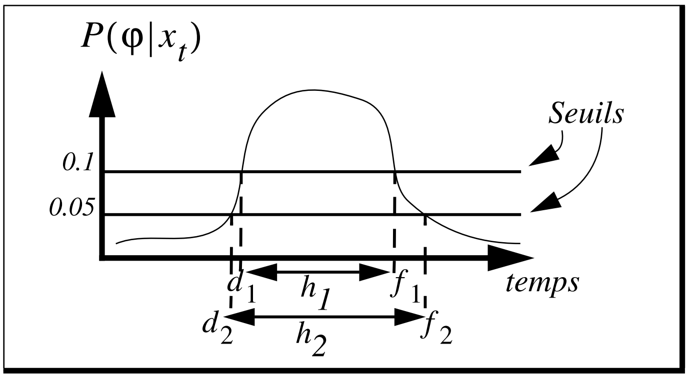

FIGURE 39. **Génération multiple d'hypothèses.**

Pour un seuil donné et pour chaque phonème φ, nous pouvons détecter les zones où la courbe $P(\varphi|x_t)$ passe au dessus de ce seuil.

Si l'on dénote par $X_b^e = \{x_b, \ldots, x_e\}$ un segment où la probabilité est supérieure au seuil, la probabilité que l'on soit en présence de la prononciation d'un phonème φ alors que la séquence de vecteurs acoustiques X_b^e a été émise s'écrit $P(\varphi|X_b^e)$. En utilisant la lois de Bayes, on peut développer cette probabilité sous la forme :

$$P(\varphi|X_b^e) = \frac{P(X_b^e|\varphi)P(\varphi)}{P(X_b^e)} \qquad \text{(EQ 65)}$$
$$= \frac{P(x_b, \ldots, x_e|\varphi)P(\varphi)}{P(x_b, \ldots, x_e)}$$

En faisant l'hypothèse que les vecteurs acoustiques sont indépendants :

$$P(x_1, x_2) = P(x_1)P(x_2),$$

Nous pouvons développer (EQ 65) en :

$$P(\varphi|X_b^e) = \prod_{t=b}^{e} P(x_t|\varphi)\frac{P(\varphi)}{\prod_{t=b}^{e} P(x_t)}.$$

Et en appliquant de nouveau la lois de Bayes sur chaque facteur, nous avons :

$$P(\varphi|X_b^e) = \frac{\prod_{t=b}^{e} P(\varphi|x_t)\prod_{t=b}^{e} P(x_t)}{\prod_{t=b}^{e} P(\varphi)}\frac{P(\varphi)}{\prod_{t=b}^{e} P(x_t)},$$

qui se réduit finalement à :

$$\boxed{P(\varphi|X_b^e) = \frac{\prod_{t=b}^{e} P(\varphi|x_t)}{P(\varphi)^{e-b}}} \quad \textbf{(EQ 66)}$$

Cette dernière relation a pour avantage d'offrir des valeurs peu sensibles à la longueur du segment, et peut donc être utilisée pour comparer des segments de différentes longueurs. Cette probabilité est notée P lorsque qu'aucune confusion n'est possible.

L'hypothèse générée par la détection d'un tel segment se compose donc de l'indice de la trame délimitant le début de l'hypothèse b, l'indice de la trame délimitant la fin de l'hypothèse e, le phonème associé φ, et la probabilité que le segment ait été émis lors de la prononciation du phonème P. Nous notons cette hypothèse :

$$h(\varphi, P, b, e).$$

Nous notons, par ailleurs, $P(h)$, la probabilité P associée à l'hypothèse h, et $\varphi(h)$, le phonème φ associé à l'hypothèse h.

En appliquant cette méthode de détection d'hypothèses pour différents seuils et pour chaque phonème, on récolte un ensemble d'hypothèses qui, une fois triées suivant un ordre croissant de leur trame initiale, b, constituent un treillis d'hypothèses $L = \{h_1, \ldots, h_M\}$ avec M, le nombre d'hypothèses générées. Ce nombre d'hypothèses peut être contrôlé en modifiant la valeur des seuils.

Ce treillis contient finalement toute l'information nécessaire à l'extraction future de mots clés. Il sera conservé jusqu'à son utilisation pour la recherche de mots clés.

Dans le tableau ci-dessous, nous affichons le contenu des hypothèses générées dans les 50 premières trames, dans le cas de l'utilisation de modèle monogaussien.

φ	**b**	**e**	**P**	**φ**	**b**	**e**	**P**
pcl	0	6	0.449	jh	28	30	0.143
h#	0	19	0.186	t	29	31	0.113
pau	3	10	0.051	k	29	32	0.115
epi	4	6	0.110	ey	32	44	0.195
pcl	5	16	0.506	iy	33	37	0.086
kcl	8	10	0.236	ae	34	44	0.119
s	13	21	0.148	ih	35	44	0.102
z	14	19	0.112	eh	36	42	0.100
t	14	18	0.189	hv	39	45	0.136
sh	15	22	0.104	dh	43	45	0.172
ch	15	23	0.235	th	43	47	0.105
jh	17	21	0.150	pcl	43	47	0.08
hh	20	23	0.082	z	44	51	0.155
t	20	23	0.301	s	44	52	0.244
y	21	30	0.368	tcl	45	47	0.116
iy	22	30	0.187	t	48	51	0.189
hv	24	36	0.193	ch	48	51	0.135
hh	27	33	0.115	jh	49	51	0.108

TABLEAU 8. **Extrait de treillis.**

4.2.4 *Analyse du contenu du treillis*

4.2.4.1 *Détection des transitions*

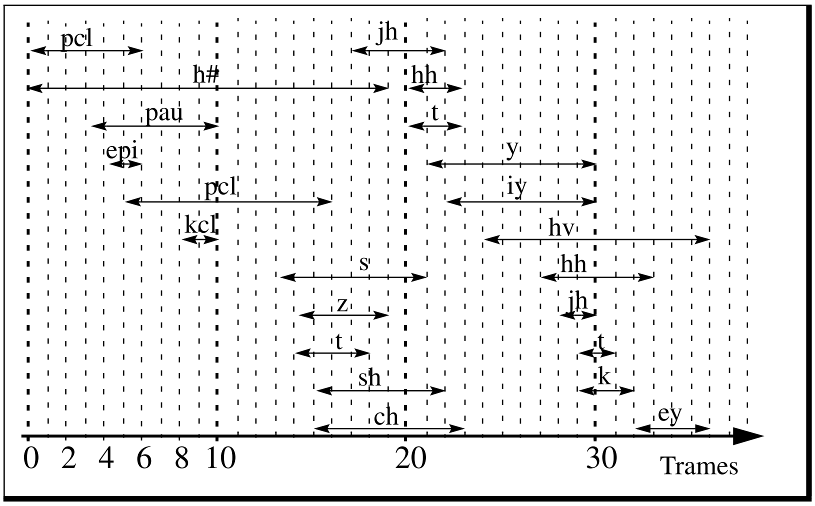

FIGURE 40. **représentation graphique d'un treillis.**

Comme le montre la figure 40, l'utilisation de seuils implique la génération d'hypothèses dont les bornes ne respectent aucune contrainte syntaxique, contrairement à une approche markovienne.

Pour palier ce défaut intrinsèque à ce type de segmentation, on envisage trois cas pour permettre l'enchaînement des séquences phonétiques :

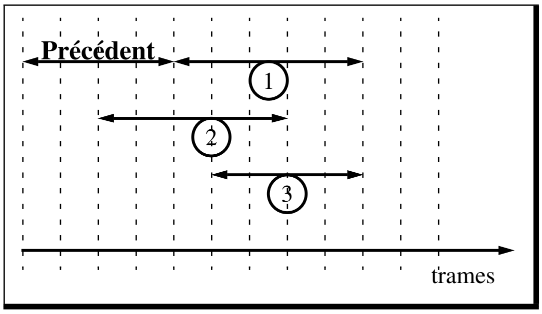

FIGURE 41. **Différents types de transitions.**

1. Le phonème suivant commence exactement à la trame suivante. C'est le cas idéal et aucune contrainte n'est imposée.
2. Le phonème commence entre le début et la fin du précédent. Dans ce cas, la seule contrainte à imposer consiste à ne prendre les phonèmes que s'ils se terminent après la fin du précédent.
3. Le phonème commence après la fin du précédent. Dans ce cas, son début ne peut être éloigné de la fin du précédent que d'un nombre fixe de trames.

D'autres méthodes ont été envisagées. On peut, par exemple, tenir compte de la durée moyenne des deux phonèmes pour estimer si la transition est possible. En considérant, par exemple, qu'il est possible d'avoir la transition si la distance entre la fin du précédent $f(h_1)$ et le début du suivant $d(h_2)$ est inférieure à une fraction α de la durée moyenne des phonèmes constituant les deux hypothèses :

$$|d(h_2) - f(h_1)| < \alpha \frac{|\mu(\varphi(h_1)) + \mu(\varphi(h_2))|}{2},$$

où $\mu(\varphi(h_1))$ correspond à la durée moyenne du phonème associé à l'hypothèse.

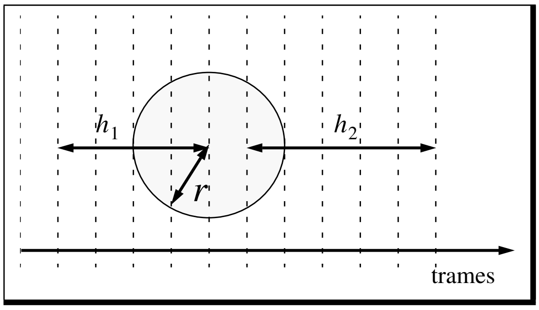

FIGURE 42. **Domaine de transitions.**

Cependant, cette approche, outre la surcharge de calculs qu'elle entraîne, conduit à estimer, à l'aide de modèles statistiques simples, les régions probables de transition, alors que l'on possède, dès l'estimation de $P(\varphi|x_t)$, une estimation réelle de ces régions de transition, étant donnés les différents seuils utilisés.

4.2.4.2 Matrice de confusion

Les hypothèses générées sont obtenues à travers un processus de seuillage. Tous les phonèmes ne sont donc pas représentés à tout moment dans le treillis, comme cela serait le cas pour une détection de mots clés basée sur une approche markovienne, fussent ces probabilités très faibles. Si dans la séquence de phonèmes recherchés, un de ceux-ci ne se retrouve pas dans le treillis (en cas de mauvaise prononciation, ou simplement à cause d'un accent marqué), le processus ne trouvera aucune séquence valable, même si tous les autres phonèmes de la séquence sont correctement détectés et présents dans le treillis. Deux méthodes ont été envisagées pour pallier ce défaut.

Première méthode

La première solution consiste à enrichir les hypothèses originales en leur ajoutant des hypothèses similaires (même départ et même fin) mais associées aux phonèmes qui se confondent le plus souvent aux phonèmes originaux. La probabilité associée à ces nouvelles hypothèses peut être estimée en accord avec les considérations suivantes.

En considérant φ_p le phonème réellement prononcé et $\tilde{\varphi}_d$ le phonème détecté par le système de reconnaissance sur un séquence donnée X_b^e, on peut écrire :

$$\begin{aligned} P(\varphi_p | X_b^e) &= \sum_{d=1}^{K} P(\tilde{\varphi}_d, \varphi_p | X_b^e) \\ &= \sum_{d=1}^{K} P(\tilde{\varphi}_d | X_b^e) P(\varphi_p | \tilde{\varphi}_d, X_b^e) \end{aligned} \quad \text{(EQ 67)}$$

En faisant les hypothèses :

- il n'y a qu'un seul $\tilde{\varphi}_d$ pour un X_b^e donné :

$$\sum_{d=1}^{K} P(\tilde{\varphi}_d, \varphi_p | X_b^e) = P(\tilde{\varphi}_d, \varphi_p | X_b^e) ;$$

- la probabilité de confusion peut être estimée par un probabilité indépendante de la séquence acoustique :

$$P(\varphi_p | \tilde{\varphi}_d, X_b^e) = P(\varphi_p | \tilde{\varphi}_d) ;$$

on trouve finalement :

$$P(\varphi_p | X_b^e) = P(\tilde{\varphi}_d | X_b^e) P(\varphi_p | \tilde{\varphi}_d) . \quad \text{(EQ 68)}$$

La probabilité $P(\varphi_p | \tilde{\varphi}_d)$ associée à la confusion entre le phonème réellement prononcé φ_p et le phonème détecté $\tilde{\varphi}_d$ est extraite d'un matrice de confusion préalablement calculée, à l'aide de la base de données d'entraînement. Les éléments de cette matrice sont estimés par dénombrement des confusions obtenues lors de l'évaluation des probabilités locales d'émissions.

$$\begin{aligned} P(\varphi_p | \tilde{\varphi}_d) &= \sum_{x \in X} P(\varphi_p, x | \tilde{\varphi}_d) \\ &= \sum_{x \in X} P(x | \tilde{\varphi}_d) P(\varphi_p | \tilde{\varphi}_d, x) \\ &= \sum_{x \in X} P(x | \tilde{\varphi}_d) P(\varphi_p | x) \end{aligned} \quad \text{(EQ 69)}$$

où le premier facteur du membre de droite correspond aux probabilités d'émission et le second est estimé suivant la segmentation de la base d'entraînement.

Il faut remarquer que les probabilités de confusion sont estimées à l'aide des vecteurs acoustiques pris séparément, alors qu'elles sont utilisées pour des séquences de vecteurs.

Deuxième méthode

La deuxième solution consiste à utiliser cette matrice de confusion, non pas sur le treillis, mais sur la séquence de phonèmes que l'on recherche.

Si l'on considère la transcription phonétique $\phi = \{\varphi_1, \ldots, \varphi_N\}$ représentant la prononciation correcte du mot clé recherché, il suffit d'appliquer la même méthode en acceptant la substitution du phonème d'origine φ_n par les q phonèmes avec lequel il se confond le plus souvent, $\left\{\tilde{\varphi}_n^1, \ldots, \tilde{\varphi}_n^q\right\}$. Précisons que le phonème d'origine φ_n est inclus dans cet ensemble de phonèmes. Le coefficient q est considéré comme un paramètre permettant l'accélération du processus au détriment de l'évaluation de toutes les hypothèses.

En agissant de la sorte, on obtient q^N séquences à rechercher. Pour réduire cette valeur, nous ne retenons que les variantes ne présentant qu'une seule substitution. Cette hypothèse forte est cependant nécessaire pour réduire le nombre de séquences à une valeur raisonnable, $G = qN$. Notons ces séquences choisies par :

$$\tilde{\phi}_g = \{\tilde{\varphi}_{g,1}, \ldots, \tilde{\varphi}_{g,N}\},$$

où $g = 1, \ldots, G$.

Chaque séquence ainsi générée, se voit directement associée une probabilité de confusion, $P(\phi|\tilde{\phi}_g)$, directement estimée par la matrice de confusion calculée comme précédemment :

$$\begin{aligned} P(\phi|\tilde{\phi}_g) &= P(\varphi_1, \ldots, \varphi_N|\tilde{\varphi}_{g,1}, \ldots, \tilde{\varphi}_{g,N_g}) \\ &= \prod_{n=1}^{N} P(\varphi_n|\tilde{\varphi}_{g,n}) \end{aligned} \quad \textbf{(EQ 70)}$$

où nous avons effectué l'hypothèse suivante :

- la probabilité d'avoir prononcé un phonème φ_n, ne dépend que du phonème détecté au même instant.

Cette dernière méthode fut choisie car elle présente l'avantage de conserver une grande vitesse de recherche, sans imposer une augmentation de la taille du treillis.

4.2.5 Algorithme de recherche dans le treillis d'hypothèses

Pour une des séquences phonétiques $\tilde{\phi}_g = \{\tilde{\varphi}_1, \ldots, \tilde{\varphi}_N\}$ donnée, nous désirons chercher dans le treillis d'hypothèses, $L = \{h_1, \ldots, h_M\}$, la séquence d'hypothèses, $H_g = \{h_{l_1}, \ldots, h_{l_N}\}$, qui maximise la probabilité :

$$P(H_g) = \prod_{i \in [l_1, \ldots, l_N]} P(h_i),$$

et telle que les transitions entre hypothèses vérifient les contraintes citées au paragraphe 4.2.4.1, à savoir :

$$\text{soit } d(h_{l_n}) < d(h_{l_{n+1}}) < f(h_{l_n}) \text{ et } f(h_{l_n}) < f(h_{l_{n+1}}), \quad \textbf{(EQ 71)}$$

$$\text{soit } f(h_{l_n}) < d(h_{l_{n+1}}) < f(h_{l_n}) + C^{st}. \quad \textbf{(EQ 72)}$$

La constante C^{st} représentant le saut possible entre deux hypothèses est fixé arbitrairement à 4 trames.

La recherche de la séquence optimale est basée sur un processus récursif.

4.2.5.1 Initialisation

On recherche dans le treillis L chaque hypothèse $h_{l_1}(\tilde{\varphi}_1, P, s, t)$, $l_1 \in [1, \ldots, (M - N + 1)]$ correspondant à une occurrence du phonème $\tilde{\varphi}_1$.

Pour chaque hypothèse trouvée, nous pouvons initialiser la séquence d'hypothèses sélectionnées par :

$$H_g^1 = \{h_{l_1}\},$$

et la probabilité associée à cette séquence par :

$$P(H_g^1) = P(h_{l_1}).$$

Ensuite, nous passons au processus itératif qui consiste à rechercher les hypothèses successives. Soit $k = 2$ l'indice de la récurrence, représentant le numéro suivant du phonème recherché.

4.2.5.2 Récurrence

Etant donnée la dernière hypothèse contenue dans H_g^{k-1}, soit $h_{l_{k-1}}(\tilde{\varphi}_{k-1}, P, s, t)$, nous cherchons les hypothèses suivantes, $h_{l_k}(\tilde{\varphi}_k, P', s', t')$ correspondant à une occurrence du phonème $\tilde{\varphi}_k$, et telles qu'elles vérifient les conditions de transition **(EQ 71)** ou **(EQ 72)**.

Pour chaque hypothèse h_{l_k} trouvée, nous construisons :

$$H_g^k = \{H_g^{k-1}, h_{l_k}\} ,$$

et calculons :

$$P(H_g^k) = P(H_g^{k-1})P(h_i)$$

Si le nombre d'hypothèses sélectionnées est inférieur à la taille de la séquence recherchée, $k < N$, nous pouvons, passer à la récurrence suivante en incrémentant k d'une unité.
Si par contre, nous avons le nombre voulu d'hypothèses, $k = N$, nous avons détecté une occurrence possible du mot clé. La probabilité associée à cette occurrence est contenue dans $P(H_g^N)$, et la séquence d'hypothèses l'ayant générée est contenue dans H_g^N. Il suffit de conserver ces informations si l'on désire conserver toutes les occurrences possibles.
Par contre, si l'on désire obtenir l'occurrence la plus probable, il suffit de vérifier si $P(H_g^N)$ est la probabilité maximale rencontrée jusqu'à l'étape N. Dans ce cas, nous sommes en présence de la séquence la plus probable depuis le début du treillis et il nous suffit de conserver cette séquence H_g^N au détriment de toute autre trouvée préalablement.

Dès que chaque hypothèse h_{l_k} a été traitée ou si aucune hypothèse n'a pu être trouvée, nous revenons à l'étape $k-1$, où nous réétudions la prolongation d'une séquence d'hypothèses préalablement trouvée.

4.2.5.3 Conclusion

En effectuant ce traitement sur tout le treillis, nous trouvons la séquence d'hypothèses offrant la plus grande probabilité de contenir le mot clé prononcé suivant la transcription phonétique $\tilde{\phi}_g$:

$$P(\tilde{\phi}_g|L) = \max_L [P(H_g^N)] .$$

En utilisant le même algorithme pour chaque transcription de $\tilde{\Phi}$, nous pouvons obtenir la probabilité maximale d'occurrence du mot clé et la séquence d'hypothèses associée :

$$P(\phi|L) = \max_g [P(\tilde{\phi}_g|L)P(\tilde{\phi}_g)] \qquad \textbf{(EQ 73)}$$

4.2.6 Résultats

4.2.6.1 Conditions expérimentales

Le signal de parole est préalablement analysé sur des fenêtres de Hamming d'une durée de 32 msec, décalées de 12 msec. Chaque trame de parole est composée des 16 coefficients cepstraux extraits de chaque segment.

Les tests nécessaires à l'évaluation de cette méthode ont tous été effectués sur la base de données DARPA TIMIT corpus 1990. Cette base de données est composée de 6300 phrases, 10 prononcées par chacun des 630 locuteurs répartis dans 8 catégories représentant la région associée à leur dialecte. Les textes lus sont composés de 2 phrases prononcées par chacun des locuteurs (SA1 et SA2), 450 phrases lues par au moins 7 personnes différentes (SX1,..., SX450) et 1890 phrases énoncées par une seule personne (SI1,..., SI1890). Le tableau (9) reprend les caractéristiques extraites du corpus.

Type de Phrases	Nombre de phrases	Nombre de locuteurs	Total	Phrases par locuteur
SA	2	630	1260	2
SX	450	7	3150	5
SI	1890	1	1890	3
Total	2342		6300	10

TABLEAU 9. **Contenu de la base TIMIT.**

Le corpus est séparé en une partie entraînement et une partie test. La partie test est composée de 168 locuteurs et est sélectionnée de manière à séparer les phrases de type SX pour qu'elles apparaissent soit lors de l'entraînement, soit lors du test, mais en aucun cas dans les deux. La partie de test contient 120 phrases différentes de type SX, chacune prononcée par 7 locuteurs différents.

Pour l'évaluation du système, nous nous focaliserons sur cette partie du corpus. Ces phrases permettent en effet de tester les résultats, indépendamment de l'accent et de la prononciation des locuteurs.

Parmi les 120 phrases tests différentes du type SX, nous avons choisi aléatoirement 20 phrases dans lesquelles nous avons extrait 20 mots clés. Ces mots clés sont repris dans le tableau (10).

N^0	Mots clés	N^0	Mots clés	N^0	Mots clés	N^0	Mots clés
sx113	muscular	sx53	vocabulary	sx290	informative	sx101	decorate
sx10	grades	sx133	pizzerias	sx109	ankle	sx199	exposure
sx110	problems	sx95	alligators	sx373	superb	sx99	society
sx103	ambulance	sx100	proceeding	sx8	silly	sx102	kidnappers
sx137	tradition	sx20	overalls	sx14	thursday	sx280	mirage

TABLEAU 10. **Liste des mots clés.**

Ces 20 mots clés, étant prononcés 7 fois dans la base de test, représentent un ensemble de 140 phrases. Ces 140 phrases sont prononcées par 99 personnes différentes (les 5 phrases de type SX prononcées par une même personne peuvent contenir plusieurs mots clés différents). Nous utiliserons ces 140 phrases pour tester notre système.

4.2.6.2 Procédure

Cette évaluation ayant pour but de vérifier l'efficacité du système dans l'optique précise de l'indexation d'une bande vidéo, nous avons cherché à accorder les conditions d'évaluation à l'utilisation qu'un documentaliste pourrait faire d'un tel outil. Ayant précisé un mot clé, il attend de cet outil qu'il repère les occurrences de ce mot clé puis les présente dans l'ordre décroissant de probabilité d'occurrence. L'efficacité d'un tel système peut être facilement évaluée par la mesure de la *précision*, définie à la section 3.1.5.

Pour chacune des 140 phrases nous avons préalablement généré leur treillis respectif, L_i, où $i = 1, \ldots, 140$.

Pour chaque mot clé, représenté par sa transcription phonétique standard, $\phi_k = \{\varphi_1, \ldots, \varphi_N\}$, où $k = 1, \ldots, 20$, nous estimons par l'utilisation de la matrice de confusion ses $G = qN$ séquences phonétiques plausibles $\tilde{\phi}_{k,g} = \{\tilde{\varphi}_1, \ldots, \tilde{\varphi}_N\}$, où q, le nombre de phonèmes pouvant être confondus avec le phonème de la séquence standard est fixé à 6. Nous montrons dans le tableau (11) les variations de la transcription phonétique du mot clé ”ankle” étant donnée sa transcription phonétique standard (“an ng k el”).

$P(\phi_p \mid \tilde{\phi}_{g,d})$	$\tilde{\phi}_{k,g}$	$P(\phi_p \mid \tilde{\phi}_{g,d})$	$\tilde{\phi}_{k,g}$
0.317479	ae ng k el	0.029433	ae ng g el
0.049545	eh ng k el	0.026612	ae ng jh el
0.039948	ay ng k el	0.025654	ae ng p el
0.031503	ey ng k el	0.025385	ae ng t el
0.030670	aw ng k el	0.024024	ae ng ch el
0.025591	ah ng k el	0.021621	ae ng f el
0.022212	ih ng k el	0.020800	ae ng hh el
0.020035	aa ng k el	0.016697	ae ng sh el
0.018969	nx ng k el	0.016457	ae ng d el
0.016155	uh ng k el	0.084083	ae ng k w
0.069514	ae eng k el	0.064687	ae ng k l
0.040384	ae en k el	0.038203	ae ng k ao
0.034847	ae n k el	0.023504	ae ng k ow
0.028616	ae m k el	0.017287	ae ng k oy
0.026252	ae em k el	0.016638	ae ng k ax
0.019293	ae bcl k el	0.010931	ae ng k ah
0.016389	ae iy k el	0.010619	ae ng k uw
0.015277	ae gcl k el	0.009116	ae ng k aa
0.013048	ae y k el		

TABLEAU 11. **Transcriptions déduites.**

A l'aide de (EQ 73), nous extrayons pour tous les treillis, la probabilité d'occurrence de ce mot clé dans chaque phrase, $P(\phi_k|L_i)$.

Ensuite, pour le mot clé donné, nous trions ces probabilités d'occurrence en ordre décroissant. Pour le mot clé "ankle" nous obtenons ainsi, dans le cas d'une distribution gaussienne, les positions du tableau (12).

Tri	$P(\phi_k\|L_i)$	n^0	SX	Détecté	Précision
1	1.1720070e-05	79	dr3/mcsh0/sx109	XXXX	1
2	1.1551630e-05	78	dr2/mpgl0/sx109	XXXX	1
3	8.6649680e-06	82	dr4/fmaf0/sx109	XXXX	1
4	8.4308670e-06	80	dr4/fadg0/sx109	XXXX	1
5	8.2102490e-06	24	dr3/mrtk0/sx103		4/5
6	5.9324970e-06	25	dr4/mlll0/sx103		4/6
7	5.0159970e-06	26	dr5/mlih0/sx103		4/7
8	5.0074000e-06	23	dr3/mjjg0/sx103		4/8
9	4.8340710e-06	83	dr7/ftlh0/sx109	XXXX	5/9
10	4.3559180e-06	28	dr8/mdaw1/sx103		5/10
11	3.9620370e-06	60	dr5/fawf0/sx100		5/11
12	3.1516090e-06	52	dr2/mmdb1/sx95		5/12
13	3.0826830e-06	33	dr7/fcau0/sx137		5/13
14	2.5319700e-06	54	dr4/frng0/sx95		5/14
15	2.4716240e-06	50	dr2/fjwb0/sx95		5/15
16	2.2559370e-06	81	dr4/flkd0/sx109	XXXX	6/17
17	2.0042810e-06	84	dr8/mpam0/sx109	XXXX	7/17
18	1.5200610e-06	85	dr2/mdld0/sx373		
19	1.2387740e-06	115	dr4/fadg0/sx199		
...	...	...	...		
140	6.0205e-10	76	dr6/mcmj0/sx104		

TABLEAU 12. **Phrases triées.**

Nous obtenons pour ce mot clé une précision moyenne de 0,76.

Du point de vue de l'utilisateur, si nous réduisons le dernier tableau sous la forme suivante,

Mot clé	1 occ.	2 occ.	3 occ.	4 occ.	5 occ.	6 occ.	7occ.
ankle	1	2	3	4	9	16	17

TABLEAU 13. **Nombre de phrases à parcourir.**

nous voyons directement le nombre de phrases qu'il est nécessaire d'écouter pour observer N occurrences du mot clé recherché. Sans traitement préalable, ce nombre vaudrait en moyenne 20*N si l'on considère les phrases recherchées équiprobablement réparties.

4.2.6.3 Sensibilité aux probabilités d'émission

Nous montrons dans cette partie, les résultats obtenus en fonction des méthodes utilisées pour l'estimation des probabilités d'émission : distribution monogaussienne, multigaussienne et neuronales. Comme le montre le tableau (14), chaque modélisation utilise un nombre différent de paramètres. Une démarche rigoureuse exigerait de comparer des systèmes de complexités équivalentes. Cependant cette démarche s'avère difficile, voir impossible à réaliser en pratique (chaque modélisation possède un nombre optimal de paramètres et l'ajout de paramètres supplémentaires n'apporte généralement que peu d'amélioration). Nous nous contentrons donc ici d'indiquer ces complexités.

Pour la modélisation monogaussienne, nous avons 61 modèles phonétiques chacun contenant un vecteur moyen et un vecteur variance, chacun de taille 16 : $16 \times 2 \times 61 = 1952$ paramètres.

Pour la modélisation multigaussienne, nous avons en moyenne 16 multigaussiennes par phonème ce qui donne : $16 \times (16 \times 2 + 1) \times 61 = 32208$ paramètres, en comptant les pondérations entre gaussiennes d'un même phonème.

Pour les réseaux de neurones, nous avons 16 neurones en couche d'entrée, 200 en couche cachée, 61 en couche de sortie, ce qui donne : $(16 + 1) \times 200 + (200 + 1) \times 61 = 15661$ paramètres, en comptant les biais.

Notons également que n'utilisant pas de modèle markovien, nous n'avons pas de paramètres représentant les probabilités de transitions entre états.

modèles	**paramètres**
Monogaussienne	1952
Multigaussienne	31232
Réseau de neurones	15661

TABLEAU 14. **Nombre de paramètres dans les différentes méthodes.**

Nous énonçons tout d'abord, pour la probabilité monogaussienne, les résultats en terme de positions d'occurrence des mots clés et de leurs moyennes. Nous explicitons également les résultats en terme de précision et précision moyenne.

Par la suite, nous appliquerons un schéma identique pour les méthodes multigaussienne et neuronale. Finalement, nous reprendrons dans un tableau comparatif les mesures obtenues par les différentes méthodes et en tirerons les conclusions.

Dans le cas de l'utilisation de probabilités *monogaussiennes*, les résultats obtenus pour tous les mots clés sont repris tableau (15).

Mots clés	1 occ.	2 occ.	3 occ.	4 occ.	5 occ.	6 occ.	7 occ.
muscular	1	2	3	4	5	7	19
grades	1	3	7	35	38	46	50
problems	1	2	3	4	19	31	93
ambulance	1	2	3	4	8	13	21
tradition	1	3	4	6	7	20	54
vocabulary	1	2	3	6	15	19	34
pizzerias	1	2	3	4	9	11	17
alligators	4	8	10	17	24	52	70
proceeding	3	4	8	13	49	57	107
overalls	1	2	7	10	11	39	47
informative	1	4	7	9	11	36	71
ankle	1	2	3	4	9	16	17
superb	6	13	37	48	103	124	126
silly	3	4	8	9	10	15	32
thursday	1	2	5	16	21	23	27
decorate	1	2	5	17	32	64	94
exposure	1	3	9	14	15	29	53
society	6	19	34	37	49	54	94
kidnappers	6	8	13	27	45	69	84
moyenne	2.1	4.45	8.75	14.4	24.3	40.6	60.5

TABLEAU 15. **Position pour la distribution gaussienne.**

En terme de précision cela donne,

Mots clés	1 occ.	2 occ.	3 occ.	4 occ.	5 occ.	6 occ.	7 occ.	Moy.
muscular	1.00	1.00	1.00	1.00	1.00	0.85	0.36	0.88
grades	1.00	0.66	0.42	0.11	0.13	0.13	0.14	0.373
problems	1.00	1.00	1.00	1.00	0.26	0.19	0.07	0.647
ambulance	1.00	1.00	1.00	1.00	0.62	0.46	0.33	0.774
tradition	1.00	0.66	0.75	0.66	0.71	0.30	0.12	0.604
vocabulary	1.00	1.00	1.00	0.66	0.33	0.31	0.20	0.646
pizzerias	1.00	1.00	1.00	1.00	0.55	0.54	0.41	0.787
alligators	0.25	0.25	0.30	0.23	0.20	0.11	0.10	0.208
proceeding	0.33	0.50	0.37	0.30	0.10	0.10	0.06	0.255
overalls	1.00	1.00	0.42	0.40	0.45	0.15	0.14	0.513
informative	1.00	0.50	0.42	0.44	0.45	0.16	0.09	0.441
ankle	1.00	1.00	1.00	1.00	0.55	0.37	0.41	0.763
superb	0.16	0.15	0.08	0.08	0.04	0.04	0.05	0.09
silly	0.33	0.50	0.37	0.44	0.50	0.40	0.21	0.396
thursday	1.00	1.00	0.60	0.25	0.23	0.26	0.25	0.515
decorate	1.00	1.00	0.60	0.23	0.15	0.09	0.07	0.451
exposure	1.00	0.66	0.33	0.28	0.33	0.20	0.13	0.423
society	0.16	0.10	0.08	0.10	0.10	0.11	0.07	0.108
kidnappers	0.16	0.25	0.23	0.14	0.11	0.08	0.08	0.154
mirage	1.00	1.00	1.00	1.00	0.83	0.06	0.07	0.711
moyenne	0.77	0.71	0.6	0.52	0.38	0.25	0.17	0.48

TABLEAU 16. **Précision et précision moyenne pour la distribution gaussienne.**

Pour la distribution *multigaussienne*, nous avons successivement les positions :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
muscular	1	2	3	4	5	7	9
grades	1	8	11	13	30	35	48
problems	1	2	4	5	11	25	118
ambulance	1	2	3	6	11	24	33
tradition	1	2	7	11	49	71	84
vocabulary	1	2	3	4	5	11	47
pizzerias	1	2	3	4	11	16	46
alligators	2	4	10	27	42	43	110
proceeding	1	6	15	18	24	41	45
overalls	3	6	12	16	21	22	23
informative	1	2	3	4	6	21	96
ankle	1	2	4	5	7	15	32
superb	11	14	45	53	61	69	78
silly	2	3	5	12	13	26	34
thursday	1	2	5	6	9	15	41
decorate	1	4	6	8	9	25	85
exposure	1	2	3	5	13	37	39
society	8	17	19	24	39	43	58
kidnappers	1	2	4	8	12	29	72
mirage	2	3	4	5	6	96	132
moyenne	2.10	4.35	8.45	11.9	19.2	33.55	61.5

TABLEAU 17. **Position pour la distribution multigaussienne.**

et la précision :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	moy
muscular	1.00	1.00	1.00	1.00	1.00	0.86	0.78	0.94
grades	1.00	0.25	0.27	0.31	0.17	0.17	0.15	0.33
problems	1.00	1.00	0.75	0.80	0.45	0.24	0.06	0.61
ambulance	1.00	1.00	1.00	0.67	0.45	0.25	0.21	0.65
tradition	1.00	1.00	0.43	0.36	0.10	0.08	0.08	0.43
vocabulary	1.00	1.00	1.00	1.00	1.00	0.55	0.15	0.81
pizzerias	1.00	1.00	1.00	1.00	0.45	0.38	0.15	0.71
alligators	0.50	0.50	0.30	0.15	0.12	0.14	0.06	0.25
proceeding	1.00	0.33	0.20	0.22	0.21	0.15	0.16	0.32
overalls	0.33	0.33	0.25	0.25	0.24	0.27	0.30	0.28
informative	1.00	1.00	1.00	1.00	0.83	0.29	0.07	0.74
ankle	1.00	1.00	0.75	0.80	0.71	0.40	0.22	0.69
superb	0.09	0.14	0.07	0.08	0.08	0.09	0.09	0.09
silly	0.50	0.67	0.60	0.33	0.38	0.23	0.21	0.42
thursday	1.00	1.00	0.60	0.67	0.56	0.40	0.17	0.63
decorate	1.00	0.50	0.50	0.50	0.56	0.24	0.08	0.48
exposure	1.00	1.00	1.00	0.80	0.38	0.16	0.18	0.65
society	0.12	0.12	0.16	0.17	0.13	0.14	0.12	0.14
kidnappers	1.00	1.00	0.75	0.50	0.42	0.21	0.10	0.57
mirage	0.50	0.67	0.75	0.80	0.83	0.06	0.05	0.52
Moyenne	0.80	0.73	0.62	0.57	0.45	0.26	0.17	0.52

TABLEAU 18. **Précision et précision moyenne pour la distribution multigaussienne.**

Pour les *réseaux de neurones*, nous avons en terme de position :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
muscular	1	2	3	4	5	7	13
grades	2	3	4	6	7	12	66
problems	1	2	3	4	6	48	49
ambulance	1	2	3	6	7	92	120
tradition	1	3	4	6	7	28	65
vocabulary	1	2	3	4	6	14	36
pizzerias	1	2	3	5	6	11	110
alligators	1	4	9	30	31	72	75
proceeding	1	5	10	15	57	66	126
overalls	1	2	3	5	7	9	15
informative	1	7	8	15	20	21	52
ankle	1	4	5	6	7	19	44
superb	1	7	16	68	78	123	127
silly	1	4	7	8	10	14	15
thursday	1	2	5	17	24	32	33
decorate	1	3	6	7	11	16	74
exposure	1	2	5	12	17	33	58
society	6	8	15	19	20	34	66
kidnappers	3	4	11	12	19	47	49
mirage	1	2	3	4	5	99	104
moyenne	1.40	3.50	6.30	12.65	17.50	39.85	64.85

TABLEAU 19. **Position lors de l'utilisation du réseau de neurones.**

et en terme de précision :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	moy
muscular	1.00	1.00	1.00	1.00	1.00	0.86	0.54	0.91
grades	0.50	0.67	0.75	0.67	0.71	0.50	0.11	0.56
problems	1.00	1.00	1.00	1.00	0.83	0.12	0.14	0.73
ambulance	1.00	1.00	1.00	0.67	0.71	0.07	0.06	0.64
tradition	1.00	0.67	0.75	0.67	0.71	0.21	0.11	0.59
vocabulary	1.00	1.00	1.00	1.00	0.83	0.43	0.19	0.78
pizzerias	1.00	1.00	1.00	0.80	0.83	0.55	0.06	0.75
alligators	1.00	0.50	0.33	0.13	0.16	0.08	0.09	0.33
proceeding	1.00	0.40	0.30	0.27	0.09	0.09	0.06	0.31
overalls	1.00	1.00	1.00	0.80	0.71	0.67	0.47	0.80
informative	1.00	0.29	0.38	0.27	0.25	0.29	0.13	0.37
ankle	1.00	0.50	0.60	0.67	0.71	0.32	0.16	0.57
superb	1.00	0.29	0.19	0.06	0.06	0.05	0.06	0.24
silly	1.00	0.50	0.43	0.50	0.50	0.43	0.47	0.55
thursday	1.00	1.00	0.60	0.24	0.21	0.19	0.21	0.49
decorate	1.00	0.67	0.50	0.57	0.45	0.38	0.09	0.52
exposure	1.00	1.00	0.60	0.33	0.29	0.18	0.12	0.50
society	0.17	0.25	0.20	0.21	0.25	0.18	0.11	0.19
kidnappers	0.33	0.50	0.27	0.33	0.26	0.13	0.14	0.28
mirage	1.00	1.00	1.00	1.00	1.00	0.06	0.07	0.73
moyenne	0.90	0.71	0.64	0.56	0.53	0.29	0.17	0.54

TABLEAU 20. **Précision lors de l'utilisation du réseau de neurones.**

En regroupant les moyennes pour souligner l'amélioration des résultats, nous obtenons :

Méthode	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ
Gaussienne	2.1	4.45	8.75	14.40	24.3	40.65	60.15
Multigaussienne	2.1	4.35	8.45	11.90	19.2	33.55	61.50
Réseau de neurones	1.4	3.50	6.30	12.65	17.5	39.85	64.85

TABLEAU 21. **Moyenne des positions pour les différentes approches.**

et pour les précisions moyennes :

Méthode	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
Gaussienne	0.77	0.71	0.60	0.52	0.38	0.25	0.17	0.48
Multigaussienne	0.80	0.73	0.62	0.57	0.45	0.26	0.17	0.52
Réseau de neurones	0.90	0.71	0.64	0.56	0.53	0.29	0.17	0.54

TABLEAU 22. **Moyenne des précisions pour les différentes approches.**

Au vu de ces résultats, nous pouvons apercevoir que la progression des résultats est faible par rapport à la complexité croissante des systèmes mis en oeuvre. Cependant, il convient de souligner que l'utilisateur final ne ressent aucunement cette complexité étant donné que l'apprentissage du système ainsi que l'extraction du treillis s'effectue dans la phase préparatoire.

Estimons le temps gagné par la personne utilisant cet outil d'indexation et regroupons ces résultats au tableau (23), suivant le nombre d'occurrences de mots clés qu'elle recherche.

Indexation	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
Sans	18	35.5	53	70.5	88	105.5	123	70
Avec	1.4	3.50	6.30	12.65	17.5	39.85	64.85	20.86
Economie	16.6	32	46.7	57.85	70.5	65.65	58.15	49.63
Pourcentage	92.22	90.14	88.11	82.06	80.11	62.23	47.28	77.45

TABLEAU 23. **Gain de temps de l'utilisateur en fonction du nombre de mots clés recherchés (approche neuronale).**

En effectuant le calcul pour les différentes méthodes, nous obtenons :

Méthode	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
Gaussienne	88.4	87.53	83.49	79.58	72.48	61.52	51.14	74.88
Multigaussienne	88.4	87.81	84.05	83.13	78.26	68.25	50.04	77.14
Réseau de neurones	92.22	90.14	88.11	82.06	80.11	62.23	47.28	77.45

TABLEAU 24. **Gain de temps pour les différentes approches.**

Sans que ces chiffres fournissent une analyse exhaustive, ils montrent néanmoins le confort d'utilisation que peut apporter un tel outil d'indexation.

En reprenant les positions moyennes du tableau (21), nous pouvons également estimé les valeurs de la courbe caractéristique par la méthode énoncée dans la section 3.1.7. Ces valeurs sont regroupées dans le tableau (25) ci-dessous :

Proba. Locales	**1/7**	**2/7**	**3/7**	**4/7**	**5/7**	**6/7**	**7/7**
Gaussienne	1.10	2.45	5.75	10.40	19.30	34.6	53.50
Multigaussiennes	1.10	2.35	5.45	7.90	14.20	27.55	54.50
Réseau de neurones	0.40	1.50	3.30	8.65	12.50	33.85	57.85

TABLEAU 25. **Estimation des points de la courbe caractéristique pour les différentes approches.**

4.2.6.4 Sensibilité à la taille du corpus

Lors de l'extraction de mots clés dans une base de données, il est évident que la fréquence d'apparition du mot clé a un effet sur les résultats énoncés en terme de position moyenne et précision moyenne. En effet, plus le corpus est grand, plus la probabilité d'obtenir des fausses alarmes est élevée.

Pour mesurer cet effet, nous avons effectué des tests équivalents mais avec un nombre de phrases égal à 140, 800 puis 1095. Les résultats ont été obtenus à partir d'une modélisation multi-gaussienne et avec un algorithme légèrement moins optimal que celui utilisé pour les comparaisons entre les différentes méthodes. C'est pour cette raison que l'on ne retrouve pas de résultats identiques lors de l'utilisation de 140 phrases.

Les positions moyennes sont reprises dans le tableau ci-dessous,

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
140	2.36	4.26	7.79	12.78	21.21	31.47	68.32
800	8.52	15.42	30.37	51.32	92.47	136.6	309.4
1095	12.2	21.9	42.5	73.2	135.4	202.5	455.6

TABLEAU 26. **Moyenne des positions suivant la taille du corpus.**

alors que les précisions moyennes sont :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	Moy
140	0.75	0.74	0.63	0.56	0.46	0.35	0.17	0.49
800	0.60	0.49	0.28	0.21	0.18	0.12	0.04	0.25
1095	0.58	0.44	0.25	0.18	0.15	0.09	0.03	0.22

TABLEAU 27. **Précision moyenne suivant la taille du corpus.**

Ces valeurs montrent, la détérioration des résultats observés en fonction de la taille du corpus. Cependant, si nous reprenons la mesure du gain de temps de l'utilisateur, on trouve :

Indexation	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
140	92.22	90.14	88.11	82.06	80.11	62.23	47.28	77.45
800	91.47	92.20	89.87	87.17	81.54	77.28	55.87	82.21
1095	91.25	92.07	89.72	86.68	80.27	75.45	52.55	81.14

TABLEAU 28. **Gain de temps de l'utilisateur en fonction de la fréquence d'apparition des mots clés recherchés.**

On peut ainsi constater que le gain en temps apporté par cet outil reste relativement constant par rapport à la fréquence d'occurrence des mots clés.

4.3 Indexation par maximum de vraisemblance

4.3.1 Principe de la méthode

La deuxième approche étudiée pour la création d'un treillis phonétique, est basée sur une approche plus classique utilisant les modèles de Markov.

L'estimation des probabilités d'émission s'effectuera d'une part par modélisation paramétrique, et d'autre part par une approche neuronale.

Nous conservons le même schéma de présentation que pour la première approche. Après, la présentation du modèle de langage utilisé, à la section 4.3.2, la méthode utilisée pour la génération du treillis est décrite à la section 4.3.4. Nous explicitons dans la section 4.3.5 l'algorithme de recherche du mot clé dans le treillis et mettons en avant les différences engendrées par les spécificités de cette approche vis-à-vis de la première. Finalement, nous présentons, dans la section 4.3.6, les résultats obtenus pour évaluer cette méthode.

4.3.2 Modèle de langage

Le modèle de langage est donc basé sur une chaîne de Markov composée par des sous-modèles décrivant chacun des phonèmes $\varphi \in \phi$, connectés de manière à générer n'importe quelle séquence phonétique possible, en accord avec une grammaire de type bigramme, comme le montre la figure 43.

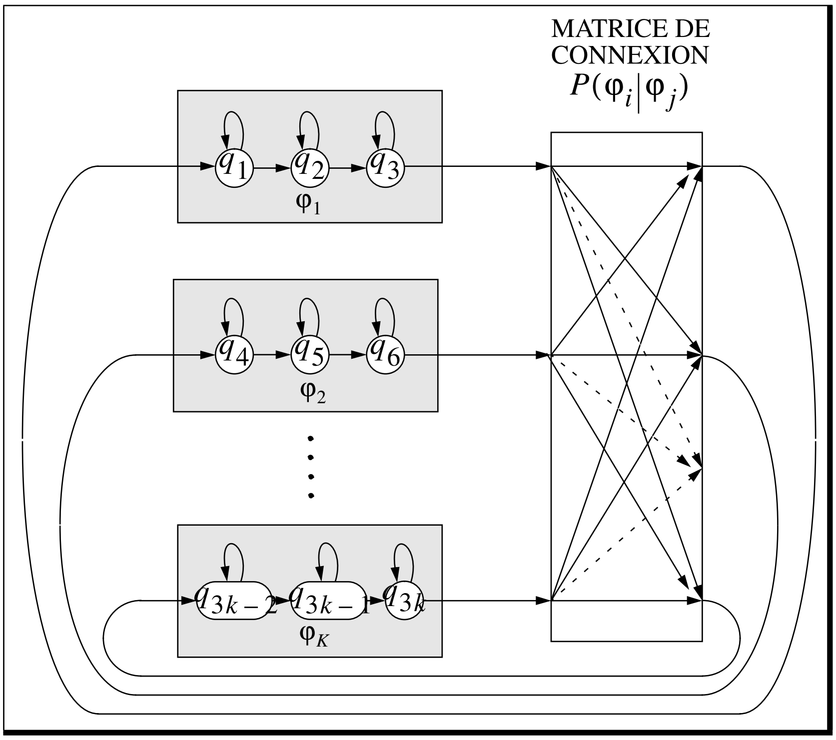

FIGURE 43. **Modèle de langage.**

Les probabilités de transition entre phonèmes sont invariantes dans le temps et estimées par dénombrement des transitions phonétiques observées dans la base de données utilisée lors de l'entraînement. En notant $N(\varphi_i, \varphi_j)$ le nombre de transitions observées entre le phonème φ_i et le phonème φ_j, nous estimons la probabilité de transition par :

$$P(\varphi_j|\varphi_i) = \frac{N(\varphi_i, \varphi_j)}{\sum_v N(\varphi_i, \varphi_v)}.$$

4.3.3 Modèle acoustique

Chaque phonème $\varphi_k \in \phi \ ,\forall k = 1, \ldots, K$, est modélisé par une chaîne de Markov cachée à trois états. Ces états sont notés q_{3k-2}, q_{3k-1} et q_{3k}. A chaque état q est associée une probabilité d'émission $P(x_t|q)$.

Deux méthodes sont utilisées pour l'estimation de cette probabilité. La première consiste à utiliser une modélisation gaussienne, et la seconde utilise une approche neuronale.

Rappelons succinctement les hypothèses sous-jacentes à l'utilisation de distributions monogaussiennes :

- utilisation de densités de probabilité en lieu et place de probabilités ;
- matrice de covariance supposée diagonale.

L'entraînement du modèle gaussien s'effectue par la méthode classique, rappelée à la section 2.5.1, utilisant itérativement l'algorithme de Viterbi pour effectuer une segmentation puis estimer les paramètres.

Pour la modélisation par réseau de neurones, rappelons qu'elle nous fournit une probabilité a posteriori $P(q|x)$, qui est liée par la loi de Bayes aux vraisemblances de la modélisation gaussienne $P(x|q)$. Son entraînement s'effectue suivant la méthode développée à la section 2.2.3.

4.3.4 Génération du treillis

4.3.4.1 Probabilité associée à un segment

En reprenant la même notation qu'au chapitre précédent, la génération du treillis consiste donc à rechercher les hypothèses $h(\varphi, P, b, e)$, où P représente la probabilité $P(X_b^e|\varphi)$ que la séquence de vecteurs acoustiques $X_b^e = \{x_b, \ldots, x_e\}$ soit émise lors de la prononciation du phonème φ.

La probabilité que cette séquence de vecteurs acoustiques X_b^e soit associée à un chemin spécifique dans le modèle de langage, $\gamma_b^e = \left\{q_\gamma^b, q_\gamma^{b+1}, \ldots, q_\gamma^e\right\}$ s'écrit :

$$P(X_b^e | \gamma_b^e) = \prod_{t=b}^{e} P(x_t | q_\gamma^t) P(q_\gamma^{t+1} | q_\gamma^t)$$

Chaque séquence d'états empruntée par un chemin dans le modèle peut également être vue comme une séquence de sous-chemins parmi les modèles phonétiques :

$$\gamma = \{\gamma_{\varphi_1}, \ldots, \gamma_{\varphi_v}, \ldots, \gamma_{\varphi_V}\},$$

où V phonèmes sont supposés être émis, et

$$\gamma_{\varphi_v} = \left\{ q_{\varphi_v}^{b_v}, \ldots, q_{\varphi_v}^{e_v} \right\},$$

où $q_{\varphi_v} = q_{3v-2}$ ou q_{3v-1} ou q_{3v} selon le chemin et les contraintes classiques de début et de fin de phonème. De plus b_v représente l'instant où le chemin γ entre dans le sous-modèle associé au phonème φ_v et e_v l'instant où il en sort. Nous avons trivialement $b_1 = b$, $_V =$, $q_{\varphi_v}^{b_v} = q_{3v-2}$ et $q_{\varphi_v}^{e_v} = q_{3v}$.

Ainsi, la probabilité associée au chemin peut s'écrire :

$$\boxed{P(X_b^e | \gamma_b^e) = \prod_{v=1}^{V} P\left(X_{b_v}^{e_v} \middle| \varphi_v\right) P(\varphi_{v+1} | \varphi_v)} \quad \textbf{(EQ 74)}$$

où :

$$\boxed{P\left(X_{b_v}^{e_v} \middle| \varphi_v\right) = \prod_{t=b_v}^{e_v} P\left(x_t \middle| q_{\varphi_v}^t\right) P\left(q_{\varphi_v}^{t+1} \middle| q_{\varphi_v}^t\right)} \quad \textbf{(EQ 75)}$$

4.3.4.2 Progression avant

Lors de la progression avant de l'algorithme de Viterbi, les probabilités associées aux meilleurs chemins finissant sur chaque état sont calculées à chaque instant de façon récurrente.

Comme il est montré dans la section 2.5.3, il n'est pas nécessaire de conserver toute l'information concernant les probabilités cumulées estimées lors de la progression avant, pour pouvoir effectuer la progression arrière nécessaire à la segmentation en phonèmes.

Lors de l'utilisation de l'algorithme de Viterbi nous ne devons conserver, à chaque instant t, que l'indice du phonème $\varphi_{best}(t)$, sur lequel se termine le meilleur chemin, et le pointa d'entrée de ce phonème $t_b(t)$.

Cependant, pour générer le treillis, comme pour une approche N-Best (section 2.5.4), nous devons conserver plus d'information. Comme le montre la figure 44, à chaque instant t, nous devons dans notre cas, conserver les N indices correspondant aux N meilleurs chemins :

$$\varphi_{best,\,1}(t),\ \varphi_{best,\,2}(t),\ \ldots,\ \varphi_{best,\,N}(t),$$

et le point d'entrée du dernier phonème contenu dans le chemin :

$$t_b(1,t),\ t_b(2,t),\ \ldots,\ t_b(N,t).$$

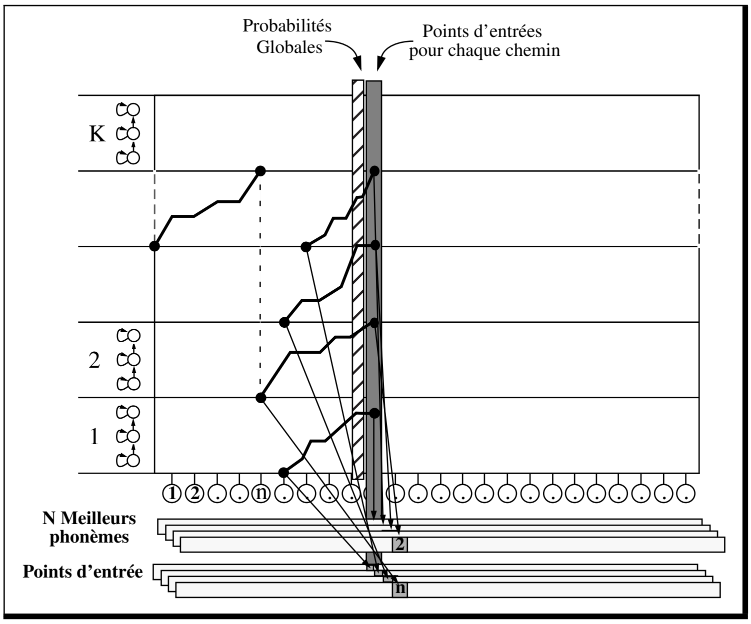

FIGURE 44. **Espace mémoire pour la génération du treillis.**

A partir de la probabilité associée à chaque meilleur chemin $P(X_1^t \mid q_{3\varphi}^t, M)$, nous pouvons extraire à l'aide de (EQ 74) la probabilité associée aux derniers phonèmes du chemin :

$$P\left(X_{t_b}^t \middle| \varphi_{best,\,n}\right),$$

En utilisant la loi de Bayes, on obtient facilement :

$$P\left(\varphi_{best,\,n}\middle|X^t_{t_b(n,\,t)}\right) = \frac{P\left(X^t_{t_b(n,\,t)}\middle|\varphi_{best,\,n}\right)}{P\left(X^t_{t_b(n,\,t)}\right)}P(\varphi_{best,\,n})\,.$$

En faisant les hypothèses successives :

- l'indépendance entre vecteurs acoustiques :

$$P\left(X^t_{t_b(n,\,t)}\right) = \prod_{u\,=\,t_b(n,\,t)}^{t} P(x_u)\,;$$

- la probabilité d'avoir un vecteur acoustique x_t est constante pour tout t :

$$P\left(X^t_{t_b(n,\,t)}\right) = \prod_{u\,=\,t_b(n,\,t)}^{t} Cst \;=\; Cst^{t-t_b(n,\,t)+1}\,,$$

nous obtenons finalement :

$$P\left(\varphi_{best,\,n}\middle|X^t_{t_b(n,\,t)}\right) = P\left(X^t_{t_b(n,\,t)}\middle|\varphi_{best,\,n}\right)\frac{P(\varphi_{best,\,n})}{Cst^{t-t_b(n,\,t)+1}}\,, \qquad \textbf{(EQ 76)}$$

que nous noterons $P(n, t)$ par souci de clarté.

Nous obtenons donc également à chaque instant, t, les probabilités associées aux N phonèmes les plus probables :

$$P(1, t), P(2, t), \ldots, P(N, t)\,.$$

Nous pouvons déjà considérer ces associations $(\varphi_{best,\,n}(t), P(n, t), t_b(n, t), t)$, comme étant des hypothèses à part entière.

La progression arrière est utilisée pour sélectionner les hypothèses qui constitueront le treillis.

4.3.4.3 Progression arrière

La détection des N meilleurs chemins lors de la progression arrière n'est pas suffisante. En effet, ces N meilleurs chemins correspondent généralement à une même séquence phonétique ayant des segmentations légèrement différentes. Nous obtiendrions donc un treillis ayant une pauvre diversité en phonèmes tout en ayant une multitude de segmentations, comme le schématise la figure ci-dessous.

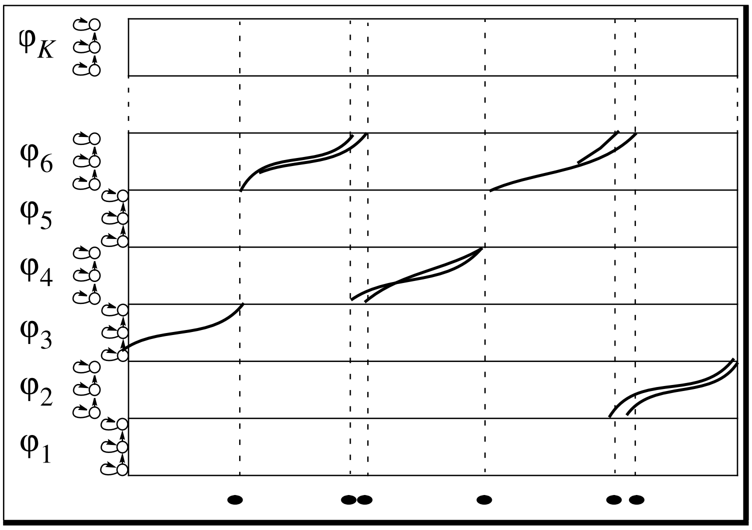

FIGURE 45. **N meilleurs chemins.**

Si par contre, nous tenons compte, lors de la progression arrière, de toutes les hypothèses alternatives lors de saut entre phonèmes, nous observons une augmentation exponentielle des hypothèses sélectionnées :

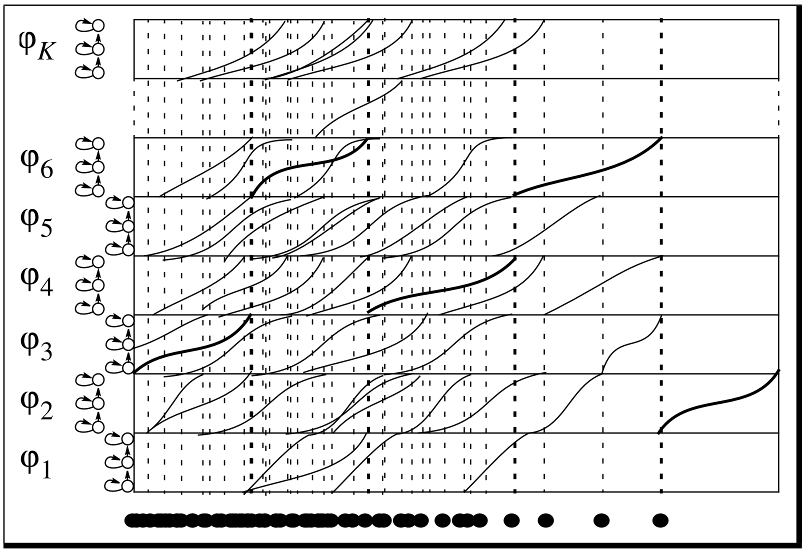

FIGURE 46. **Progression arrière et sélection globale.**

Nous devons donc trouver un compromis entre ces deux extrêmes.

La méthode choisie est basée sur la sélection préalable des transitions entre phonèmes que l'on inscrira dans le treillis, puis de la sélection des hypothèses proprement dite. Nous pouvons séparer cette méthode en 3 étapes.

La première étape consiste à sélectionner les points de transition entre phonèmes, générés par le meilleur chemin (algorithme de Viterbi).

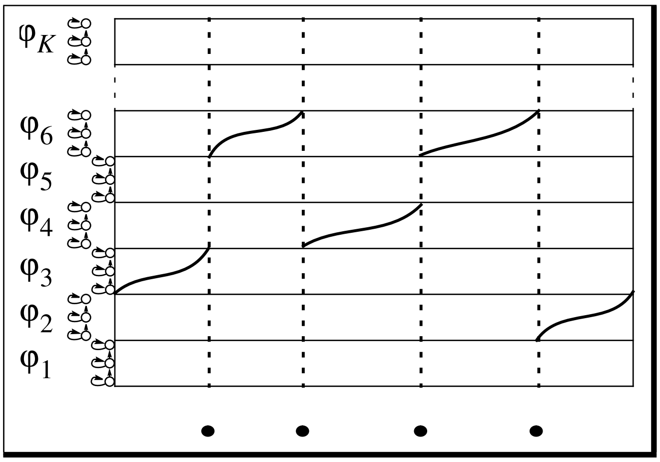

FIGURE 47. **Première sélection des noeuds.**

Nous appelons *noeud*, tout endroit de transition sélectionné pour le treillis.

La deuxième étape consiste, à partir de chaque noeud sélectionné, à considérer les N meilleures hypothèses finissant en ces points et à conserver les points de départ de chaque hypothèse en tant que nouveaux noeuds.

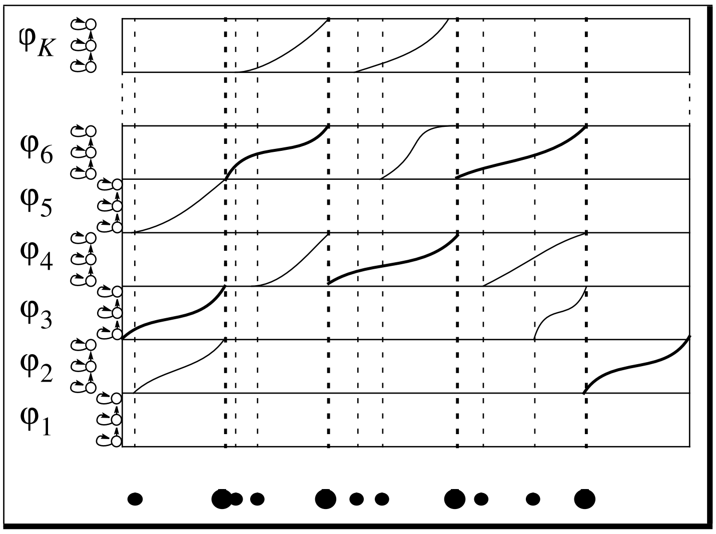

FIGURE 48. **Deuxième sélection des noeuds.**

La dernière étape consiste, à partir de chaque noeud, à conserver les N meilleures hypothèses finissant sur ce noeud. Cependant, pour celles finissant sur les noeuds générés lors de la deuxième étape, leur point de départ ne correspond à aucun noeud. Pour conserver un treillis homogène, nous modifions le point de départ de ces hypothèses de manière à les mettre en correspondance avec le noeud existant le plus proche (soit vers le futur, soit vers le passé).

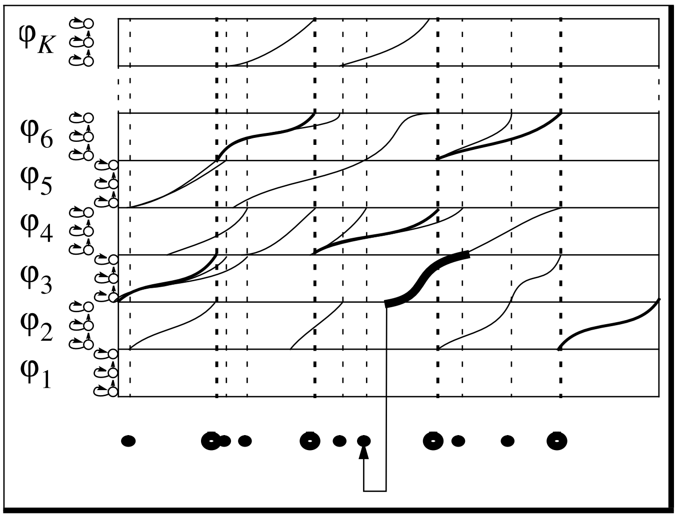

FIGURE 49. **Mise en correspondance des hypothèses.**

Cette dernière opération implique une modification de la probabilité associée à cette hypothèse.

En reprenant l'équation **(EQ 75)** :

$$P\left(X_{b_v}^{e_v} \middle| \varphi_v\right) = \prod_{t = b_v}^{e_v} P\left(x_t \middle| q_{\varphi_v}^t\right) P\left(q_{\varphi_v}^{t+1} \middle| q_{\varphi_v}^t\right),$$

on remarque qu'en première approximation, si l'on considère $P\left(x_t \middle| q_{\varphi_v}^t\right) = a$ et $P\left(q_{\varphi_v}^{t+1} \middle| q_{\varphi_v}^t\right) = b$, où a et b sont constants, on obtient les deux relations :

$$P\left(X_{t_b(n,t)}^t \middle| \varphi_v\right) = (ab)^{t - t_b(n,t) + 1} \text{ et } P\left(X_{t_b'(n,t)}^t \middle| \varphi_v\right) = (ab)^{t - t_b'(n,t) + 1}.$$

on en tire aisément :

$$P\left(X_{t_b'(n,t)}^t \middle| \varphi_v\right) = P\left(X_{t_b(n,t)}^t \middle| \varphi_v\right)^{\frac{t - t_b'(n,t) + 1}{t - t_b(n,t) + 1}}. \quad \text{(EQ 77)}$$

et injectant ce résultat dans **(EQ 76)**, on obtient :

$$P\left(\varphi_{best,\,n} \middle| X^{t}_{t_b'(n,\,t)}\right) = P\left(X^{t}_{t_b(n,\,t)} \middle| \varphi_{best,\,n}\right)^{\frac{t - t_b'(n,\,t) + 1}{t - t_b(n,\,t) + 1}} \frac{P(\varphi_{best,\,n})}{Cst^{t - t_b'(n,\,t) + 1}} \qquad \textbf{(EQ 78)}$$

Par ailleurs, de **(EQ 76)**, on peut extraire :

$$P\left(X^{t}_{t_b(n,\,t)} \middle| \varphi_{best,\,n}\right) = P\left(\varphi_{best,\,n} \middle| X^{t}_{t_b(n,\,t)}\right) \frac{Cst^{t - t_b(n,\,t) + 1}}{P(\varphi_{best,\,n})} ,$$

qui une fois introduit dans **(EQ 78)** donne la relation utilisée pour ajuster les bornes des hypothèses :

$$\boxed{P\left(\varphi_{best,\,n} \middle| X^{t}_{t_b'(n,\,t)}\right) = P\left(\varphi_{best,\,n} \middle| X^{t}_{t_b(n,\,t)}\right)^{\frac{t - t_b'(n,\,t) + 1}{t - t_b(n,\,t) + 1}} P(\varphi_{best,\,n})^{1 - \frac{t - t_b'(n,\,t) + 1}{t - t_b(n,\,t) + 1}}} \qquad \textbf{(EQ 79)}$$

Une fois ces modifications apportées, nous réalisons aisément que les hypothèses ainsi regroupées forment des *groupes d'hypothèses. A*u sein de chaque groupe, toutes les hypothèses ont les mêmes noeuds de départ et d'arrivée.

Nous noterons :

$$H^{e}_{b} = \{h_1(\varphi, P, b, e), \ldots, h_{N(b,\,e)}(\varphi, P, b, e)\} , \qquad \textbf{(EQ 80)}$$

le regroupement d'hypothèses, où $N(b, e)$ correspond au nombre d'hypothèses de ce groupe.

4.3.5 Recherche dans le treillis

Etant donnée la structure du treillis, les transitions entre phonèmes sont triviales car elles coïncident toujours avec le début d'autres hypothèses (par l'entremise des noeuds). Il n'est plus nécessaire, comme lors de l'étiquetage des trames, de considérer les sauts temporels entre phonèmes. La recherche du mot clé s'en trouve donc accélérée.

4.3.5.1 Matrice de confusion

Les hypothèses ayant été obtenues par une méthode basée sur une approche N-Best, on peut considérer que la prononciation phonétique propre au locuteur est partiellement prise en compte. Par exemple, si la personne prononce le mot “chaise” en prononçant le deuxième phonème entre le “é” et le “è”, l’algorithme utilisé génèrera les hypothèses “é” et “è”. Cependant, en cas de prononciation totalement erronée (“prononciation parfaite d’un “é”), le treillis ne contiendra pas l’hypothèse “è” et la recherche n’aboutira pas.

Pour éviter cette éventualité, ainsi que pour prendre en compte la qualité du système de décodage phonétique, nous utilisons, comme dans le cas de l’étiquetage de trames, une matrice de confusion. Cette matrice prend en compte les similarités entre phonèmes, permettant ainsi plus de flexibilité pour la représentation phonétique. Elle peut, dans une certaine mesure, masquer les problèmes de mauvais étiquetage phonétique dans la base de données d’entraînement et également réduire les problèmes de mauvaises prononciations.

La méthode de détermination de la matrice de confusion est quasi identique à celle utilisée lors de l’étiquetage de trames. Nous nous basons sur une segmentation phonétique de référence obtenue en utilisant l’algorithme de Viterbi à partir de la transcription phonétique estimée par un expert. A l’aide de cette segmentation, nous pouvons estimer les probabilités de confusion entre phonèmes, en fonction de la probabilité d’émission des différents états.

En modifiant légèrement **(EQ 69)** pour tenir compte des états, nous trouvons :

$$\begin{aligned} P(\varphi_p|q_i) &= \sum_{x \in X} P(\varphi_p, x|q_i) \\ &= \sum_{x \in X} P(x|q_i)P(\varphi_p|q_i, x) \\ &= \sum_{x \in X} P(x|q_i)P(\varphi_p|x) \end{aligned} \quad \textbf{(EQ 81)}$$

Par ailleurs, nous avons :

$$\begin{aligned} P(\tilde{\varphi}_d|\varphi_p) &= P(q_{3d-2} \vee q_{3d-1} \vee q_{3d}|\varphi_p) \\ &= P(q_{3d-2}|\varphi_p) + P(q_{3d-1}|\varphi_p) + P(q_{3d}|\varphi_p) \\ &= \frac{P(\varphi_p|q_{3d-2})P(q_{3d-2}) + P(\varphi_p|q_{3d-1})P(q_{3d-1}) + P(\varphi_p|q_{3d})P(q_{3d})}{P(\varphi_p)} \end{aligned}$$

qui, en utilisant **(EQ 81)**, fournit la matrice de confusion :

$$P(\varphi_p|\tilde{\varphi}_d) = \sum_{x \in X}\left(\sum_{i=0}^{2} P(x|q_{3d-i})P(q_{3d-i})\right)\frac{P(\varphi_p|x)}{P(\varphi_p)}, \quad \textbf{(EQ 82)}$$

où $P(\varphi_p)$ et $P(q_{3d-i})$ peuvent être estimées par dénombrement, $P(x|q_{3d-i})$ sont les probabilités d’émission et $P(\varphi_p|x)$ vaut 0 ou 1 suivant la segmentation de référence.

4.3.5.2 Algorithme de recherche

La méthode utilisée pour la recherche d'un mot clé consiste à modifier le treillis en fonction du mot clé recherché. Cette modification peut aisément s'effectuer simultanément à la recherche proprement dite.

Modification du treillis

Cette modification utilise une approche identique à la première méthode d'utilisation de la matrice de confusion lors de l'étiquetage de trames, exposée à la section 4.2.4.2.

Soit le mot clé représenté par sa séquence phonétique $\phi = \{\varphi_1, \ldots, \varphi_p, \ldots, \varphi_N\}$. Pour chaque groupe d'hypothèses, H_b^e, nous pouvons estimer la probabilité d'avoir prononcé un des phonèmes de ϕ : φ_p, $p = 1, \ldots, N$, connaissant les hypothèses associées à H_b^e : $h(\tilde{\varphi}_d, P_d, b, e)$, $d = 1, \ldots, N(b, e)$, $P_d = P(\tilde{\varphi}_d | X_b^e)$.

En reprenant l'équation **(EQ 67)**, nous avons :

$$\begin{aligned} P(\varphi_p | X_b^e) &= \sum_{d=1}^{N(b,e)} P(\tilde{\varphi}_d, \varphi_p | X_b^e) \\ &= \sum_{d=1}^{N(b,e)} P(\tilde{\varphi}_d | X_b^e) P(\varphi_p | \tilde{\varphi}_d, X_b^e) \end{aligned}$$

En faisant l'hypothèse suivante :

- la probabilité de confusion peut être estimée par une probabilité indépendante de la séquence acoustique :

$$P(\varphi_p | \tilde{\varphi}_d, X_b^e) = P(\varphi_p | \varphi_d),$$

on trouve finalement :

$$P(\varphi_p | X_b^e) = \sum_{d=1}^{N(b,e)} P(\tilde{\varphi}_d | X_b^e) P(\varphi_p | \tilde{\varphi}_d) \qquad , \forall p = 1, \ldots, N. \tag{EQ 83}$$

Ces nouvelles probabilités peuvent générer de nouvelles hypothèses, qui, rangées en groupe d'hypothèses, forment un nouveau treillis, dépendant du mot clé prononcé. Soit L, ce treillis.

Algorithme

Nous cherchons dans ce nouveau treillis L la meilleure séquence d'hypothèses $\mathcal{H} = \{h_{l_1}, \ldots, h_{l_N}\}$ telle qu'elle maximise la probabilité :

$$P(\mathcal{H}) = \prod_{i \in [l_1, \ldots, l_N]} P(h_i),$$

où $l_n \in [1, M]$ et telle que les hypothèses soient contenues dans des groupes d'hypothèses contigus :

$$\text{Si } h_{l_i} \in H_{b_i}^{e_i} \text{ et } h_{l_{i+1}} \in H_{b_{i+1}}^{e_{i+1}}, \text{ alors } e_i = b_{i+1} \quad \forall i = 1, \ldots, N-1.$$

Cette recherche de la séquence optimale est basée sur un processus récursif identique à celui utilisé lors de l'étiquetage des trames.

Notons cependant que la détection d'hypothèses successives est nettement plus rapide que dans la méthode précédente, car il suffit de chercher le phonème suivant dans les groupes d'hypothèses contigus à celui où le dernier phonème a été détecté.

Par cet algorithme nous détectons donc les endroits où la séquence phonétique représentant le mot clé apparaît. Il fournit également une probabilité associée à chacune de ces détections.

4.3.6 Résultats

L'évaluation de cette méthode a été conduite de manière identique à celle basée sur l'étiquetage de trames, dans le but de pouvoir comparer les deux méthodes.

Nous présentons successivement les résultats obtenus lors de l'utilisation de l'approche *gaussienne* puis lors de l'approche *neuronale*.

Comme le montre le tableau (29), le nombre de paramètres utilisé dans cette approche markovienne est légèrement suppérieur à ceux utilisé dans l'approche précédente car il faut maintenant tenir compte des probabilités de transitions.

Pour la modélisation gaussienne, nous avons trois états par phonèmes, ce qui nous donne : $3 \times 16 \times 2 \times 61 + 61 \times 61 = 9577$ paramètres.

Pour la modélisation par réseaux de neurones, nous avons : $(16 + 1) \times 200 + (200 + 1) \times 61 + 61 \times 61 = 19382$ paramètres.

modèles	**paramètres**
Monogaussienne	9577
Réseau de neurones	19382

TABLEAU 29. **Nombre de paramètres dans les différentes méthodes.**

Ces résultats sont, comme précédemment, exprimés en terme de position d'occurrence ainsi qu'en terme de précision.

Lors de l'utilisation de densité de probabilité gaussienne, les résultats obtenus en terme de position d'occurrence sont repris ci-dessous :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
muscular	1	2	3	4	5	7	12
grades	1	2	3	4	6	7	8
problems	1	2	3	4	5	6	13
ambulance	1	4	5	9	18	22	100
tradition	1	3	4	7	8	9	88
vocabulary	1	8	19	71	72	73	74
pizzerias	1	2	3	4	5	8	20
alligators	1	2	3	5	7	11	12
proceeding	1	2	3	4	5	6	7
overalls	1	2	3	5	6	14	53
informative	1	2	5	46	49	126	127
ankle	2	3	6	8	34	78	105
superb	1	2	5	28	30	45	121
silly	1	2	3	4	7	13	14
thursday	1	2	3	7	18	29	70
decorate	1	4	5	7	9	10	12
exposure	1	2	3	4	5	8	9
society	1	2	3	4	5	18	23
kidnappers	1	2	7	18	49	135	136
mirage	4	41	51	115	124	136	139
moyenne	1.2	4.55	7.00	17.9	23.35	38.05	57.15

TABLEAU 30. **Position dans l'approche gaussienne.**

Nous pouvons aisément en extraire les résultats en terme de précision :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	Moy
muscular	1.00	1.00	1.00	1.00	1.00	0.86	0.58	0.92
grades	1.00	1.00	1.00	1.00	0.83	0.86	0.88	0.93
problems	1.00	1.00	1.00	1.00	1.00	1.00	0.54	0.93
ambulance	1.00	0.50	0.60	0.44	0.28	0.27	0.07	0.45
tradition	1.00	0.67	0.75	0.57	0.62	0.67	0.08	0.62
vocabulary	1.00	0.25	0.16	0.06	0.07	0.08	0.09	0.24
pizzerias	1.00	1.00	1.00	1.00	1.00	0.75	0.35	0.87
alligators	1.00	1.00	1.00	0.80	0.71	0.55	0.58	0.80
proceeding	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
overalls	1.00	1.00	1.00	0.80	0.83	0.43	0.13	0.74
informative	1.00	1.00	0.60	0.09	0.10	0.05	0.06	0.41
ankle	0.50	0.67	0.50	0.50	0.15	0.08	0.07	0.35
superb	1.00	1.00	0.60	0.14	0.17	0.13	0.06	0.44
silly	1.00	1.00	1.00	1.00	0.71	0.46	0.50	0.81
thursday	1.00	1.00	1.00	0.57	0.28	0.21	0.10	0.59
decorate	1.00	0.50	0.60	0.57	0.56	0.60	0.58	0.63
exposure	1.00	1.00	1.00	1.00	1.00	0.75	0.78	0.93
society	1.00	1.00	1.00	1.00	1.00	0.33	0.30	0.80
kidnappers	1.00	1.00	0.43	0.22	0.10	0.04	0.05	0.41
mirage	0.25	0.05	0.06	0.03	0.04	0.04	0.05	0.07
moyenne	0.94	0.83	0.76	0.64	0.57	0.46	0.34	0.65

TABLEAU 31. **Précision dans l'approche gaussienne.**

En utilisant un réseau de neurones, nous avons obtenu les résultats suivants, exprimés en terme de position d'occurrence des mots clés :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
muscular	1	2	3	4	6	7	10
grades	1	2	3	4	5	7	8
problems	1	2	3	4	5	6	14
ambulance	1	4	6	9	17	19	52
tradition	1	2	4	7	14	15	68
vocabulary	1	3	17	35	69	71	75
pizzerias	1	2	3	4	5	6	15
alligators	1	2	5	6	9	10	13
proceeding	1	2	3	4	5	6	7
overalls	1	2	5	6	9	10	42
informative	1	2	6	32	65	75	98
ankle	2	3	4	19	56	72	86
superb	1	3	6	16	32	54	86
silly	1	2	3	4	7	16	32
thursday	1	2	3	6	17	28	75
decorate	1	3	6	7	8	9	27
exposure	1	2	3	4	5	6	7
society	1	2	3	4	15	20	56
kidnappers	1	2	8	15	42	56	75
mirage	2	7	47	56	75	87	98
moyenne	1.10	2.55	7.05	12.30	23.30	29.00	47.20

TABLEAU 32. **Position dans l'approche neuronale.**

Si nous exprimons ces résultats en terme de précision, nous obtenons :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	Moy
muscular	1.00	1.00	1.00	1.00	0.83	0.86	0.70	0.91
grades	1.00	1.00	1.00	1.00	1.00	0.86	0.88	0.96
problems	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.93
ambulance	1.00	0.50	0.50	0.44	0.29	0.32	0.13	0.46
tradition	1.00	1.00	0.75	0.57	0.36	0.40	0.10	0.60
vocabulary	1.00	0.67	0.18	0.11	0.07	0.08	0.09	0.32
pizzerias	1.00	1.00	1.00	1.00	1.00	1.00	0.47	0.92
alligators	1.00	1.00	0.60	0.67	0.56	0.60	0.54	0.71
proceeding	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
overalls	1.00	1.00	0.60	0.67	0.56	0.60	0.17	0.66
informative	1.00	1.00	0.50	0.12	0.08	0.08	0.07	0.41
ankle	0.50	0.67	0.75	0.21	0.09	0.08	0.08	0.34
superb	1.00	0.67	0.50	0.25	0.16	0.11	0.08	0.40
silly	1.00	1.00	1.00	1.00	0.71	0.38	0.22	0.76
thursday	1.00	1.00	1.00	0.67	0.29	0.21	0.09	0.61
decorate	1.00	0.67	0.50	0.57	0.62	0.67	0.26	0.61
exposure	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
society	1.00	1.00	1.00	1.00	0.33	0.30	0.12	0.68
kidnappers	1.00	1.00	0.38	0.27	0.12	0.11	0.09	0.42
mirage	0.50	0.29	0.06	0.07	0.07	0.07	0.07	0.16
moyenne	0.95	0.87	0.72	0.63	0.50	0.49	0.33	0.64

TABLEAU 33. **Précision dans l'approche neuronale.**

En regroupant les mesures moyennes pour les deux méthodes d'évaluation de probabilités d'émissions, on trouve en terme de position :

Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ
Gaussienne	1.2	4.55	7.00	17.93	23.35	38.05	57.15
Réseau de neurones	1.1	2.55	7.05	12.30	23.30	29.00	47.20

TABLEAU 34. **Moyenne des positions pour les différentes approches.**

En terme de précision :

Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy.
Gaussienne	0.94	0.83	0.76	0.64	0.57	0.46	0.34	0.65
Réseau de neurones	0.95	0.87	0.72	0.63	0.50	0.49	0.33	0.64

TABLEAU 35. **Moyenne des précision pour les différentes approches.**

En reprenant la mesure de l'estimation du gain de temps de l'utilisateur d'un tel outil d'indexation, nous obtenons :

Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy.
Gaussienne	93.37	87.25	86.79	74.62	73.56	63.99	53.57	76.16
Réseau de neurones	93.92	92.85	86.69	82.56	73.62	72.55	61.82	80.57

TABLEAU 36. **Moyenne des gains de temps pour les différentes approches.**

Ces résultats montrent une amélioration de la qualité du système d'indexation lors de l'utilisation des modèles neuronaux. Rappelons que cette amélioration peut s'expliquer par la flexibilité qu'offrent les réseaux de neurones quant aux formes des densités de probabilités.

4.4 Indexation par probabilités a posteriori

4.4.1 Principe de la méthode

La dernière approche étudiée se fonde sur la théorie développée par H.Bourlard, Y. Konig et N. Morgan, [BOU95]. Cette théorie est exposée à la section 2.6, "REMAP", où nous décrivons le modèle acoustique, le modèle de langage, ainsi que la méthode d'entraînement des réseaux de neurones. Nous expliquerons ici les méthodes utilisées pour générer le treillis (section 4.4.2) et effectuer la recherche dans ce dernier (section 4.4.3). Nous expliciterons ensuite les résultats engendrés par cette méthode (section 4.4.4).

4.4.2 Génération du treillis

Lors de la création du treillis, nous appliquons un schéma équivalent à celui utilisé pour l'approche précédente. Premièrement, nous effectuons la détection de la meilleure segmentation phonétique pour sélectionner les bornes d'hypothèses. Deuxièmement, nous utilisons les N meilleurs phonèmes pour détecter d'autres segmentations possibles et augmenter le nombre de bornes pour la génération de treillis. Finalement, pour chaque hypothèse phonétique générée, nous calculons la probabilité à lui associer.

4.4.2.1 Segmentation principale

Pour déterminer le chemin optimal, nous utilisons l'algorithme de Viterbi. Or celui-ci n'est pas applicable tel quel avec les réseaux de neurones entraînés suivant le critère de maximisation des probabilités a posteriori. Il est donc modifié comme suit.

De l'équation de récurrence de la progression avant de REMAP, **(EQ 52)** reproduite ci-dessous,

$$\alpha_i^t = \sum_{k=1}^{K} \alpha_k^{t-1} P(q_i^t \mid q_k^{t-1}, X_1^t, M) c_k^t,$$

nous pouvons aisément déduire une approche de type Viterbi prenant uniquement en compte le meilleur chemin :

$$\begin{aligned}
v_i^t &= P(\gamma_{max,i}^t, X_1^t \mid M) \\
&= \max_k P(q_i^t, \gamma_{max,k}^{t-1}, x_t, X_1^{t-1} \mid M) \\
&= \max_k P(q_i^t, x_t \mid \gamma_{max,k}^{t-1}, X_1^{t-1}, M) P(\gamma_{max,k}^{t-1}, X_1^{t-1} \mid M) \\
&= \max_k P(x_t \mid \gamma_{max,k}^{t-1}, X_1^{t-1}, M) P(q_i^t \mid \gamma_{max,k}^{t-1}, X_1^t, M) \nu_k^{t-1}
\end{aligned}$$

En utilisant les hypothèses successives :

- la chaîne de Markov est du premier ordre :

  $P(q_i^t \mid \gamma_{max,k}^{t-1}, X_1^{t-1}, M) = P(q_i^t \mid q_k^{t-1}, X_1^{t-1}, M)$,

  $P(x_t \mid \gamma_{max,k}^{t-1}, X_1^{t-1}, M) = P(x_t \mid q_k^{t-1}, X_1^{t-1}, M)$;

- la probabilité de transition conditionnelle ne dépend que des vecteurs acoustiques temporellement proches (fenêtre $[n-c, n+d]$) :

  $P(q_i^t \mid q_k^{t-1}, X_1^{t-1}, M) = P(q_i^t \mid q_k^{t-1}, X_{t-c}^{t+d}, M)$;

- la probabilité d'émettre x_t, connaissant X_1^{t-1}, et l'état précédent, q_k^{t-1}, peut soit être estimée par un simple algorithme de prédiction acoustique dépendant de l'état précédent, soit être supposée constante. Nous utiliserons cette dernière hypothèse et noterons c_k^t cette probabilité :

$$P(x_t \mid q_k^{t-1}, X_1^{t-1}, M) = c_k^t ; \qquad \textbf{(EQ 84)}$$

on obtient finalement :

$$\boxed{v_i^t = \max_k \nu_k^{t-1} P(q_i^t \mid q_k^{t-1}, X_{t-c}^{t+d}, M) c_k^t} \qquad \textbf{(EQ 85)}$$

Cette formule de récurrence peut être utilisée au même titre que la récurrence classique de Viterbi pour détecter la meilleure séquence phonétique.

Nous utiliserons cette relation pour extraire, tout comme dans l'approche utilisée à la section précédente, la première sélection de noeuds, ainsi que les premières hypothèses contenant les meilleurs phonèmes et leurs bornes respectives, (voir figure 47).

4.4.2.2 *Segmentation secondaire*

Il est également possible de conserver, par cette dernière équation légèrement modifiée, non plus le meilleur chemin, mais les N meilleurs chemins. Ces derniers nous permettent lors de la rétropropagation de générer, tout comme dans la section 4.3.4.3, "Progression arrière", un deuxième groupe de noeuds ainsi que de nouvelles hypothèses partant de ces nouveaux noeuds pour aboutir sur les noeuds de la première génération, (voir figure 48).

Une troisième génération d'hypothèses partant de noeuds existants est également utilisée pour rendre accessibles les nouveaux noeuds. En partant des N meilleurs chemins finissant sur les nouveaux noeuds, nous sélectionnons les derniers phonèmes prononcés sur ces chemins pour former les hypothèses de troisième génération. Les débuts de ces hypothèses sont fixés par la règle du plus proche voisin, comme commenté à la section 4.3.4.3, (voir figure 49).

4.4.2.3 *Probabilités associées aux hypothèses*

Connaissant les bornes d'entrées et de sorties des phonèmes associées aux hypothèses, nous pouvons aisément estimer la probabilité associée à chaque hypothèse.

En effet, pour une hypothèse donnée $h(\varphi_k, P, b, e)$ nous recherchons :

$$P\Big((\varphi_k)_b^e \Big| X, M\Big) = \sum_{i \neq k} \sum_{j \neq k} P(q_i^{b-1}, q_k^b, q_\varphi^{b+1}, \dots, q_k^e, q_j^{e+1} \Big| X, M) \qquad \text{(EQ 86)}$$

qui peut s'écrire de manière simplifiée :

$$P\Big((\varphi_k)_b^e \Big| X, M\Big) = P\Big(q_{\overline{k}}^{b-1}, q_k^b, q_\varphi^{b+1}, \dots, q_k^e, q_{\overline{k}}^{e+1} \Big| X, M\Big), \qquad \text{(EQ 87)}$$

où q_k représente l'état lié au phonème φ_k, et $q_{\overline{k}}$ représente n'importe quel état sauf q_k.

En développant cette dernière relation, nous avons :

$$P\Big((\varphi_k)_b^e\Big|X, M\Big) = P\Big(q_{\bar{k}}^{b-1}, q_k^b, q_k^{b+1}, \ldots, q_k^e, q_{\bar{k}}^{e+1}\Big|X, M\Big)$$

$$= P\Big(q_{\bar{k}}^{b-1}\Big|X, M\Big)P\Big(q_k^b, q_k^{b+1}, \ldots, q_k^e, q_{\bar{k}}^{e+1}\Big|q_{\bar{k}}^{b-1}, X, M\Big)$$

$$= P\Big(q_{\bar{k}}^{b-1}\Big|X, M\Big)P\Big(q_k^b\Big|X, q_{\bar{k}}^{b-1}, M\Big)P\Big(q_k^{b+1}, \ldots, q_k^e, q_{\bar{k}}^{e+1}\Big|q_{\bar{k}}^{b-1}, q_k^b, X, M\Big)$$

$$= \ldots$$

$$= P\Big(q_{\bar{k}}^{b-1}\Big|X, M\Big)P\Big(q_k^b\Big|X, q_{\bar{k}}^{b-1}, M\Big) \prod_{t=b+1}^{e} P(q_k^t|q_k^{t-1}, X, M)P\Big(q_{\bar{k}}^{e+1}\Big|q_k^e, X, M\Big)$$

où nous avons utilisé l'hypothèse suivante :

- la chaîne de Markov du premier ordre :

$$P(q_i^t|q_k^{t-1}, q_j^{t-2}, X_1^{t-1}, M) = P(q_i^t|q_k^{t-1}, X_1^{t-1}, M).$$

Nous pouvons dès à présent effectuer l'estimation, en tenant compte de :

$$P\Big(q_{\bar{k}}^{b-1}\Big|X, M\Big) = \sum_{l \neq k} P(q_l^{b-1}|X, M),$$

et en se référant aux équations **(EQ 59)** et **(EQ 61)** :

$$P(q_k^{t-1}|X, M, \Lambda) = \frac{\alpha_k^{t-1}\gamma_k^{t-1}}{\sum_{k=1}^{K} \alpha_k^{t-1}\gamma_k^{t-1}}$$

et

$$P(q_k^t|X, q_j^{t-1}, M, \Lambda) = \frac{c_j^t P(q_k^t|q_j^{t-1}, X_{t-c}^{t+d}, M)\gamma_j^t}{\gamma_j^{t-1}}.$$

Cependant, nous pouvons ajouter les hypothèses :

- la probabilité de transition conditionnelle ne dépend que des vecteurs acoustiques temporellement proches (fenêtre $[n-c, n+d]$) :

$P(q_i^t | q_k^{t-1}, X_1^{t-1}, M) = P(q_i^t | q_k^{t-1}, X_{t-c}^{t+d}, M)$;

- suivant la sélection des bornes de l'hypothèse $h(\varphi_k, P, b, e)$, on peut considérer $P(q_k^{b-1} | X, M)$ et $P(q_k^{e+1} | q_k^e, X, M)$ suffisamment petits pour être négligés, se qui conduit trivialement à :

  $P\left(q_{\bar{k}}^{b-1} \middle| X, M\right) = 1 - P(q_k^{b-1} | X, M) \cong 1$

  et $P\left(q_{\bar{k}}^{e+1} \middle| q_k^e, X, M\right) = 1 - P(q_k^{e-1} | X, q_k^e, M) \cong 1$.

Dès lors, la nouvelle estimation peut être donnée par :

$$\boxed{P\left((\varphi_k)_b^e \middle| X, M\right) = P(q_k^b | X_{b-c}^{b+d}, q_{\bar{k}}^{b-1}, M) \prod_{t=b+1}^{e} P(q_k^t | q_k^{t-1}, X_{t-c}^{t+d}, M)} \quad \text{(EQ 88)}$$

où le premier facteur peut être calculé par :

$$P(q_k^b | X_{b-c}^{b+d}, q_{\bar{k}}^{b-1}, M) = \sum_{i \neq k} P(q_k^b | X_{b-c}^{b+d}, q_i^{b-1}, M),$$

vu la définition exacte donnée par **(EQ 86)**.

Pour raison de simplicité, nous utiliserons cette dernière méthode d'évaluation.

4.4.3 Recherche dans le treillis

La méthode utilisée pour la construction du treillis est similaire à celle de l'approche par maximisation de la vraisemblance énoncée section 4.3.5.

4.4.3.1 Matrice de confusion

La matrice de confusion utilisé est estimé par :

$$
\begin{aligned}
P(\varphi_p|\tilde{\varphi}_d) &= \sum_{t=1}^{N}\sum_{k=1}^{K} P(\varphi_p^t, \varphi_k^{t-1}, X_{t-c}^{t+d}|\tilde{\varphi}_d^t) \quad ,\\
&= \sum_{t=1}^{N}\sum_{k=1}^{K} P(\varphi_k^{t-1}, X_{t-c}^{t+d}|\tilde{\varphi}_d^t)P(\varphi_p^t|\tilde{\varphi}_d^t, \varphi_k^{t-1}, X_{t-c}^{t+d})\\
&= \sum_{t=1}^{N}\sum_{k=1}^{K} \frac{P(\tilde{\varphi}_d^t|\varphi_k^{t-1}, X_{t-c}^{t+d})P(\varphi_k^{t-1}, X_{t-c}^{t+d})}{P(\tilde{\varphi}_d^t)}P(\varphi_p^t|\tilde{\varphi}_d^t, \varphi_k^{t-1}, X_{t-c}^{t+d})\\
&= \sum_{t=1}^{N}\sum_{k=1}^{K} \frac{P(\tilde{\varphi}_d^t|\varphi_k^{t-1}, X_{t-c}^{t+d})P(\varphi_k^{t-1}|X_{t-c}^{t+d})P(X_{t-c}^{t+d})}{P(\tilde{\varphi}_d^t)}P(\varphi_p^t|\tilde{\varphi}_d^t, \varphi_k^{t-1}, X_{t-c}^{t+d})
\end{aligned}
$$

où :

- la probabilité $P(\varphi_p^t|\tilde{\varphi}_d^t, \varphi_k^{t-1}, X_{t-c}^{t+d})$ peut être estimée par $P(\varphi_p^t|\varphi_k^{t-1}, X)$, l'objectif imposé lors de l'entraînement du réseau (cf. **(EQ 61)**) ;
- la probabilité $P(\varphi_k^{t-1}|X_{t-c}^{t+d})$ est le probabilité associée aux fréquences d'apparition, (cf. **(EQ 59)**) ;
- la probabilité $P(\tilde{\varphi}_d^t|\varphi_k^{t-1}, X_{t-c}^{t+d})$ est estimé à l'aide du réseau de neurones auquel on applique le couple $(\varphi_k^{t-1}, X_{t-c}^{t+d})$ en entrée ;
- la probabilité a priori $P(\tilde{\varphi}_d^t)$ est supposée constante et également estimée par le réseau de neurones ;
- la probabilité a priori $P(X_{t-c}^{t+d})$ est supposé constante sur la base d'entraînement, et égale à $\frac{1}{N_x}$.

Ainsi que le montre la figure 50 et la figure 51, la matrice de confusion est effectivement à diagonale dominante.

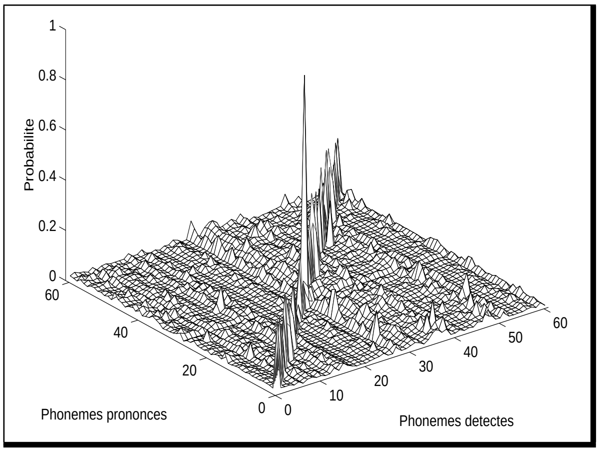

FIGURE 50. Matrice de confusion, vue en perspective.

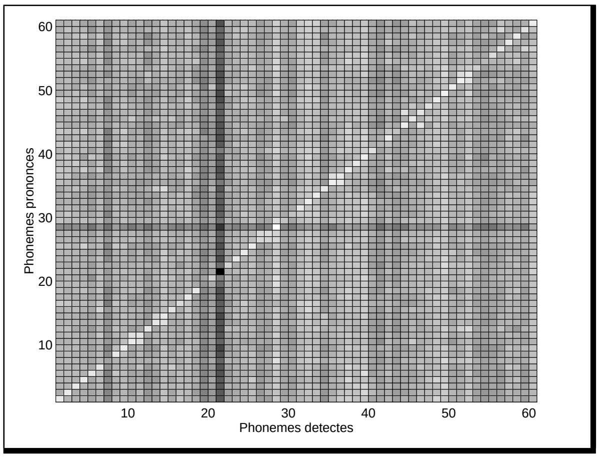

FIGURE 51. Matrice de confusion, vue en plan.

4.4.3.2 Modification du treillis

De la même manière que dans l'approche par maximisation de vraisemblance, les probabilités associées aux hypothèses sont réestimées en tenant compte de la matrice de confusion à l'aide de l'équation **(EQ 83)** :

$$P(\varphi_p | X_b^e) = \sum_{d=1}^{N(b,e)} P(\tilde{\varphi}_d | X_b^e) P(\varphi_p | \tilde{\varphi}_d) \qquad , \forall p = 1, \ldots, N.$$

4.4.3.3 Algorithme de recherche

L'algorithme de recherche est également identique à celui utilisé par l'approche précédente, section .

4.4.4 Résultats

L'évaluation de cette méthode à été conduite de manière identique aux deux autres méthodes (étiquetage de trames et maximum de vraisemblance).

Nous présentons successivement les résultats obtenus lors de l'utilisation de l'approche gaussienne puis lors de l'approche neuronale. Ces résultats sont, comme précédemment, exprimés en terme de position d'occurrence ainsi qu'en terme de précision.

Le réseau de neurones utilisé par cette approche est composé de 200 neurones en couche cachée et prend en entrée 9 vecteurs acoustiques consécutifs, chacun composé de 16 coefficients cepstraux.

L'utilisation des vecteurs acoustique voisin augmente le nombre de paramètres utilisés par cette méthode, qui comme le montre le tableau **(37)** monte à :

$(16 \times 9 + 1) \times 200 + (200 + 1) \times 61 + 61 \times 61 = 44982$ paramètres.

modèles	paramètres
Réseau de neurones	44982

TABLEAU 37. **Nombre de paramètres utilisés dans REMAP.**

Les résultats obtenus en terme de position d'occurrence sont repris dans le tableau **(38)** :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ
muscular	1	2	3	4	5	6	9
grades	1	2	4	5	6	7	12
problems	1	2	3	6	7	18	23
ambulance	1	2	6	9	16	45	62
tradition	2	3	4	8	23	30	45
vocabulary	1	4	15	24	37	68	85
pizzerias	1	2	3	4	5	6	10
alligators	1	2	6	7	14	35	58
proceeding	1	2	3	4	6	7	8
overalls	1	3	5	12	17	29	30
informative	1	2	3	7	15	22	51
ankle	1	2	5	6	11	18	43
superb	1	2	3	5	7	16	20
silly	1	3	5	13	15	23	31
thursday	1	2	5	6	9	11	17
decorate	1	3	4	6	7	9	11
exposure	1	2	3	4	5	6	7
society	1	2	3	6	12	15	24
kidnappers	1	2	3	5	9	14	25
mirage	2	3	4	13	33	40	53
moyenne	1.1	2.35	4.5	7.7	12.95	21.25	31.20

TABLEAU 38. **Position dans l'approche REMAP.**

En terme de précision, nous avons :

Mots clés	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	Moy
muscular	1.00	1.00	1.00	1.00	1.00	1.00	0.78	0.97
grades	1.00	1.00	0.75	0.80	0.83	0.86	0.58	0.83
problems	1.00	1.00	1.00	0.67	0.71	0.33	0.30	0.72
ambulance	1.00	1.00	0.50	0.44	0.31	0.13	0.11	0.50
tradition	0.50	0.67	0.75	0.50	0.22	0.20	0.16	0.43
vocabulary	1.00	0.50	0.20	0.17	0.14	0.09	0.08	0.31
pizzerias	1.00	1.00	1.00	1.00	1.00	1.00	0.70	0.96
alligators	1.00	1.00	0.50	0.57	0.36	0.17	0.12	0.53
proceeding	1.00	1.00	1.00	1.00	0.83	0.86	0.88	0.94
overalls	1.00	0.67	0.60	0.33	0.29	0.21	0.23	0.48
informative	1.00	1.00	1.00	0.57	0.33	0.27	0.14	0.61
ankle	1.00	1.00	0.60	0.67	0.45	0.33	0.16	0.60
superb	1.00	1.00	1.00	0.80	0.71	0.38	0.35	0.75
silly	1.00	0.67	0.60	0.31	0.33	0.26	0.23	0.48
thursday	1.00	1.00	0.60	0.67	0.56	0.55	0.41	0.69
decorate	1.00	0.67	0.75	0.67	0.71	0.67	0.64	0.73
exposure	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
society	1.00	1.00	1.00	0.67	0.42	0.40	0.29	0.68
kidnappers	1.00	1.00	1.00	0.80	0.56	0.43	0.28	0.72
mirage	0.50	0.67	0.75	0.31	0.15	0.15	0.13	00.38
moyenne	0.95	0.89	0.78	0.65	0.55	0.46	0.38	0.67

TABLEAU 39. **Précision dans l'approche REMAP.**

Si nous reprenons la mesure de l'estimation du gain de temps de l'utilisateur, nous obtenons :

1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7 occ	Moy
93.9	93.4	91.5	89.1	85.3	79.9	74.6	86.8

TABLEAU 40. **Moyenne du gain de temps dans l'approche REMAP.**

4.5 Comparaison entre les trois méthodes

4.5.1 Nombre de paramètres

Avant toute comparaison entre les méthodes, reprenons dans le tableau (41) le nombre de paramètre utilisés dans les différentes méthodes.

Méthodes	Proba. Locales	Paramètres
Etiquetage de trames	Gaussienne	1952
	Multigaussienne	31232
	Réseau de neurones	15661
HMM	Gaussienne	9577
	Réseau de neurones	19382
REMAP	Réseau de neurones	44982

TABLEAU 41. **Nombre de paramètres utilisés par les différentes approches.**

4.5.2 Mesures en terme de “position”

En regroupant les résultats obtenus par les différentes méthodes en terme de “position”, nous obtenons le tableau ci-dessous :

Méthodes	Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ
Etiquetage de trames	Gaussienne	2.1	4.45	8.75	14.40	24.3	40.65	60.15
	Multigaussienne	2.1	4.35	8.45	11.90	19.2	33.55	61.50
	Réseau de neurones	1.4	3.50	6.30	12.65	17.5	39.85	64.85
HMM	Gaussienne	1.2	4.55	7.00	17.93	23.35	38.05	57.15
	Réseau de neurones	1.1	2.55	7.05	12.30	23.30	29.00	47.20
REMAP	Réseau de neurones	1.1	2.35	4.5	7.7	12.95	21.25	31.20

TABLEAU 42. **Moyenne des positions pour les différentes approches.**

4.5.3 Mesures en termes de "précision"

En regroupant les résultats obtenus en terme de "précision", nous pouvons remplir le tableau ci-dessous :

Méthodes	Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy.
Etiquetage de trames	Gaussienne	0.77	0.71	0.60	0.52	0.38	0.25	0.17	0.48
	Multigaussienne	0.80	0.73	0.62	0.57	0.45	0.26	0.17	0.52
	Réseau de neurones	0.90	0.71	0.64	0.56	0.53	0.29	0.17	0.54
HMM	Gaussienne	0.94	0.83	0.76	0.64	0.57	0.46	0.34	0.65
	Réseau de neurones	0.95	0.87	0.72	0.63	0.50	0.49	0.33	0.64
REMAP	Réseau de neurones	0.95	0.89	0.78	0.65	0.55	0.46	0.38	0.67

TABLEAU 43. **Moyenne des précisions pour les différentes approches.**

4.5.4 Mesures en termes de "gain de temps"

En terme de "gain de temps", nous pouvons regrouper les résultats dans le tableau ci-dessous :

Méthodes	Proba. Locales	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy.
Etiquetage de trames	Gaussienne	88.4	87.53	83.49	79.58	72.48	61.52	51.14	74.88
	Multigaussiennes	88.4	87.81	84.05	83.13	78.26	68.25	50.04	77.14
	Réseau de neurones	92.2	90.14	88.11	82.06	80.11	62.23	47.28	77.45
HMM	Gaussienne	93.37	87.25	86.79	74.62	73.56	63.99	53.57	76.16
	Réseau de neurones	93.92	92.85	86.69	82.56	73.62	72.55	61.82	80.57
REMAP	Réseau de neurones	93.9	93.4	91.5	89.1	85.3	79.9	74.6	86.8

TABLEAU 44. **Moyenne des gains de temps pour les différentes approches.**

4.5.5 Mesures par les "courbes caractéristiques"

Pour estimer les résultats en fonction des courbes caractéristiques, nous utilisons la méthode exposée dans la section 3.1.7 qui nous permet d'obtenir certains points de la courbe caractéristique repris dans le tableau ci-dessous :

Méthodes	Proba. Locales	1/7	2/7	3/7	4/7	5/7	6/7	7/7
Etiquetage de trames	Gaussienne	1.10	2.45	5.75	10.40	19.30	34.6	53.50
	Multigaussiennes	1.10	2.35	5.45	7.90	14.20	27.55	54.50
	Réseau de neurones	0.40	1.50	3.30	8.65	12.50	33.85	57.85
HMM	Gaussienne	0.20	2.55	4.00	13.93	18.35	32.05	50.15
	Réseau de neurones	0.10	0.55	4.05	8.30	18.30	23.00	40.20
REMAP	Réseau de neurones	0.10	0.35	1.50	3.70	7.95	15.35	24.20

TABLEAU 45. **Points des courbes caractéristiques pour les différentes approches.**

En reportant ces points en fonction des probabilités de détection, nous obtenons la figure 52 :

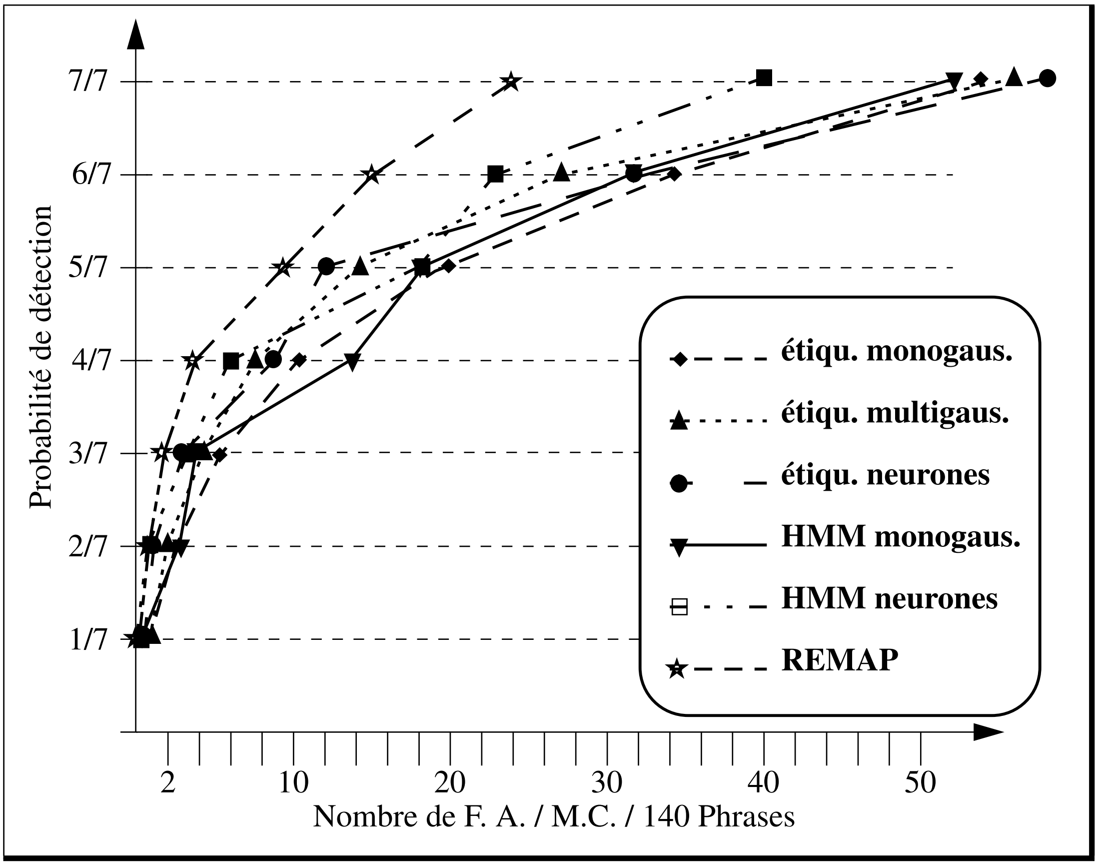

FIGURE 52. **Courbes caractéristiques pour les différentes approches.**

4.5.6 Commentaires

Ces résultats mettent en évidence l'efficacité des modèles markoviens vis-à-vis des méthodes basées sur l'étiquetage des trames. Ces différences sont principalement dues aux contraintes lexicales imposées par le modèle markovien et qui sont absentes du modèle basé sur l'étiquetage de trames.

De même, en effectuant la comparaison entre les modèles gaussiens et les modèles hybrides, nous pouvons remarquer, que l'utilisation du modèle hybride apportait une amélioration tant au niveau de la position que de la mesure du gain.

Quant au fonctionnement de REMAP, les bons résultats vis-à-vis des autres méthodes sont d'une part du au modèle de langage maintenant contenu dans le réseau plus flexible, mais également à l'utilisation des entrées contextuelles et aux plus grand nombre de paramètres impliqué dans le modèle.

Rappelons également les résultats obtenus lors de l'évaluation de la première méthode, concernant la sensibilité du système en fonction de la fréquence d'apparition des mots clés, tableau (46) :

Indexation	1 occ	2 occ	3 occ	4 occ	5 occ	6 occ	7occ	Moy
140	92.22	90.14	88.11	82.06	80.11	62.23	47.28	77.45
800	91.47	92.20	89.87	87.17	81.54	77.28	55.87	82.21
1095	91.25	92.07	89.72	86.68	80.27	75.45	52.55	81.14

TABLEAU 46. **Gain de temps de l'utilisateur en fonction de la fréquence d'apparition des mots clés recherchés.**

On peut ainsi constater que le gain en temps apporté par cet outil reste relativement constant par rapport à la fréquence d'occurrence des mots clés.

CHAPITRE 5

Conclusions & Perspectives

En reprenant les points importants de ce travail, nous présentons les conclusions générales issues de notre étude sur l'indexation par reconnaissance de mots clés.

Nous esquissons ensuite les lignes directrices des évolutions possibles des outils d'indexation développés dans cette thèse.

5.1 Conclusion

L'univers du multimédia est en pleine mutation. En effet, il y a à peine quelques années, le domaine d'indexation automatique par le contenu était inexploré. Non seulement parce que la quantité d'informations à trier était beaucoup moins importante et que la diversité de leur nature était réduite aux données textuelles, mais également parce que la puissance de calcul requise était nettement insuffisante. Récemment, nombre de recherches dans le domaine de l'indexation ont vu le jour. Le travail présenté ici en est un exemple.

Dans la première partie de ce travail, nous nous sommes appliqués à décrire les outils théoriques nécessaires à l'élaboration d'un outil d'indexation par détection de mots clés.
Apres un bref survol de la techniques choisie pour l'extraction des caractéristiques du signal acoustique, nous avons mis en évidence l'intérêt des réseaux de neurones en tant que classificateurs probabilistes et le rôle qu'ils pouvaient jouer dans le cadre de notre étude.
Nous avons ensuite décrit la théorie des chaînes de Markov et nous avons souligné les hypothèses nécessaires à leurs utilisations dans la reconnaissance de parole.
L'analyse de l'algorithme de Viterbi dans le cadre de l'indexation nous a amené à concevoir une méthode de rétropropagation automatique afin d'optimiser la recherche de la meilleure séquence phonétique dans des signaux acoustiques de longue durée.
Ensuite, nous avons exposé une méthode d'optimisation des modèles markoviens fondée sur l'estimation des probabilités a posteriori. Nous nous sommes également attachés à dégager son caractère discriminant et à décrire sa mise en oeuvre.

Dans la seconde partie du travail, nous nous sommes concentrés sur l'élaboration de systèmes de recherche de mots clés respectant les contraintes spécifiques à l'indexation que sont respectivement : l'indépendance du locuteur et du vocabulaire, mais également rapidité de recherche sur n'importe quel mot clé. Ces contraintes nous ont obligé à concevoir un système de détection dont le travail est séparé en deux parties. La première, pouvant être exécutée préalablement à toute requête, consiste à générer un treillis d'hypothèses phonétiques. La seconde partie consiste à rechercher le mot clé voulu dans ce treillis et est exécuté au moment même de la requête.
Lors de la description du premier système, basé sur l'étiquetage des vecteurs acoustiques, nous avons détaillé et mis en oeuvre les différents mécanismes nécessaires pour mener à bien cette tâche. Nous avons détaillé les différentes modélisations phonétiques, à savoir : la modélisation gaussienne, multigaussienne et neuronale. Lors de la génération du treillis d'hypothèses, nous nous avons montré la nécessité de lisser des probabilités phonétiques associées aux trames acoustiques en effectuant une intégration temporelle sur une durée proportionnelle à la durée moyenne des différents phonèmes considérés. Nous avons également montré la technique utilisée pour la détection des hypothèses phonétiques ainsi que les bornes temporelles repérant leurs début et fin. Nous avons par ailleurs montré que l'utilisation de différents seuils au niveau des probabilités d'occurrence du phonème permettait de détecter différentes bornes pour une même occurrence et ainsi conserver une certaine flexibilité quant aux transitions entre phonèmes. Nous avons par ailleurs montré les formules permettant d'estimer les probabilités à associer aux différentes hypothèses ainsi détectées. En étudiant le treillis ainsi formé, nous avons

analysé les différentes transitions possibles entre phonèmes, et déduit les règles nécessaires à la recherche du mot clé. Pour conserver une plus grande flexibilité du treillis, nous avons introduit le concept de matrice de confusion et avons discutée différentes manière de l'utiliser. Nous avons ensuite décrit en détail l'algorithme utilisé pour la recherche du mot clé dans le treillis.
Après avoir décrit les conditions expérimentales utilisées pour comparer le système d'indexation entre les différents modèles phonétiques, nous avons exprimé les résultats obtenus en terme de "position", "précision", "gain de temps" et "courbe caractéristique".
Nous avons par ailleurs analysé la sensibilité du système à la fréquence d'apparition des mots clés. Cette analyse nous a permis de constater que cet outil d'indexation conservait des propriétés intéressantes même dans le cas où la fréquence d'apparition des mots clés était faible.

La première évolution du système élaboré repose sur l'utilisation des modèles markoviens classique pour la modélisation du langage. Nous avons tout d'abord décrit le modèle de langage utilisé, puis nous nous sommes attardé à l'estimation de la probabilité qu'un segment acoustique donné ait été émis lors de la prononciation d'un phonème particulier. Nous avons ensuite décrit le principe utilisé pour élaborer le treillis. Ce principe, repose sur l'utilisation d'un algorithme de type "N-Best" pour tout d'abord détecter les noeuds du treillis, puis ensuite détecter les différentes hypothèses phonétiques reliant les noeuds du treillis. Lors de la description de l'algorithme de recherche dans le treillis, nous avons relevé les différences entre la première méthode pour l'estimation de la matrice de confusion ainsi que dans l'algorithme de recherche proprement dit. Nous avons ensuite donné les résultats obtenus par la modélisation phonétique de type monogaussienne et neuronale.

La dernière évolution proposée s'appuie sur l'utilisation simultanée des réseaux de neurones et de la théorie de maximisation des probabilités a posteriori nommée REMAP. Pour pouvoir appliquer cette nouvelle théorie à notre problème d'indexation, nous en avons déduit un algorithme équivalent à l'algorithme de Viterbi et son extension vers l'algorithme N-Best. Nous en avons également déduit une nouvelle formulation pour l'estimation des probabilités d'existence des différentes hypothèses générées ainsi que pour la création de la matrice de confusion. Nous avons ensuite présenté les résultats obtenus par cette nouvelle méthode.

Finalement, nous avons comparé les différentes évolutions présentées et mesuré les améliorations apportées par la modélisation de plus en plus complexe. Comme nous l'avons vu à la section 4.5, toute personne concernée par la gestion de documents multimédia pourrait aisément gagner 80% de son temps consacré au classement de ces documents, si elle utilisait un tel système d'aide à l'indexation.

5.2 Perspectives

Comme nous venons de le rappeler, le système d'indexation mis au point lors de ce travail est dès à présent utilisable car il peut offrir un gain de temps considérable à ses utilisateurs. Cependant, il peut encore évoluer et nous énonçons ici différentes idées susceptibles d'apporter une amélioration au système. La première partie des améliorations successives relève plutôt de considérations théoriques tandis que la seconde partie se penche spécialement sur des considérations pratiques.

5.2.1 Estimation du prédicteur acoustique de REMAP

Lors de l'étude de la maximisation des probabilités a posteriori (voir section 4.4.2.1), nous avons précisé que nous négligions l'influence du facteur relatif à la prédiction acoustique $c_k^t = P(x_t | q_k^{t-1}, X_1^{t-1}, M)$ dans les formules de récurrences :

$$\alpha_i^t = \sum_{k=1}^{K} \alpha_k^{t-1} P(q_i^t | q_k^{t-1}, X_1^t, M) c_k^t \text{ et } \gamma_i^t = c_i^{t+1} \sum_{k=1}^{K} P(q_k^{t+1} | q_i^t, X_{t-c}^{t+d}, M) \gamma_k^{t+1} .$$

Cette hypothèse est forte et il serait utile d'analyser plus en profondeur les implications de cette hypothèse. Il serait sans doute utile de considérer l'utilisation d'un simple prédicteur acoustique associé à chaque état du modèle pour estimer cette probabilité.

5.2.2 Mise en parallèle de Baum-Welch et REMAP

Une autre amélioration envisageable peut être suggérée par les observations suivantes.

Nous avons vus que la différence entre l'algorithme de REMAP et l'algorithme de Baum-Welch tenait essentiellement dans la séparation des variables lors du développement de la probabilité associée à la contribution locale :

$$P(q_i^t, x_t | q_k^{t-1}, X_1^{t-1}, M) .$$

Si nous mettons en parallèle les équations de “récurrence avant” de l’algorithme REMAP :

$$P(q_i^t, X_1^t \middle| M) = \sum_{k=1}^{K} \underbrace{P(x_t \middle| q_k^{t-1}, X_1^{t-1}, M)}_{\text{Réseau de Neurones}} \underbrace{P(q_i^t \middle| q_k^{t-1}, X_1^{t}, M)}_{\text{Processus markovien}} P(q_k^{t-1}, X_1^{t-1} \middle| M)$$

et de l’algorithme de Baum-Welch :

$$P(q_i^t, X_1^t \middle| M) = \sum_{k=1}^{K} \underbrace{P(q_i^t \middle| q_k^{t-1}, X_1^{t-1}, M)}_{\text{Réseau de Neurones}} \underbrace{P(x_t \middle| q_i^t, q_k^{t-1}, X_1^{t-1}, M)}_{\text{Prédicteur acoustique}} P(q_k^{t-1}, X_1^{t-1} \middle| M),$$

nous remarquons que la première estime la probabilité d’émission du vecteur acoustique par une approche neuronale tandis que la seconde utilise les réseaux de neurones pour estimer la probabilité de visite de l’état suivant. Nous pourrions envisager l’utilisation simultanée de deux réseaux de neurones, l’un utilisé pour estimer $P(q_i^t | q_k^{t-1}, X_1^{t-1}, M)$ et l’autre pour estimer $P(x_t | q_i^t, q_k^{t-1}, X_1^{t-1}, M)$. Nous combinerions peut-être ainsi l’avantage des deux approches tout en évitant leurs hypothèses : l’indépendance des vecteurs acoustiques pour l’estimation des probabilités de visite dans l’approche Baum-Welch, et l’utilisation de prédiction acoustique simple dans l’approche REMAP.

5.2.3 Représentation phonétique multiple des mots clés

Lors de notre étude, nous avons pu mesurer l’importance de la représentation phonétique des mots clés.
Pour pallier les erreurs d’estimation du système inhérentes à la variation de la prononciation, nous avons utilisé une matrice de confusion. Cependant, bien que plus restrictif, la simple intégration d’un dictionnaire de prononciation des mots clés rendrait le système nettement moins sensible à ces perturbations.

5.2.4 Robustesse aux bruits

Lors d’applications réelles, les signaux de parole sont enregistrés dans des conditions fort variables. Les enregistrements effectués dans les studios ou sur les plateaux de télévisions sont généralement de très bonne qualité technique tandis que les enregistrements extérieurs et les reportages souffrent souvent de bruit extérieurs tels que les passages de véhicules à proximité, d’autres personnes discutant dans le voisinage des micros, etc.

En outre, il est fréquent de superposer de la musique (jingle, musique d'introduction, musique de fond) ou du bruitage volontaire (sonnerie de téléphone, crissement de pneu, etc) aux dialogues.

Dans le cas d'enregistrements bruités, l'extraction des caractéristiques de la parole est nettement plus compliquée que dans le cas non bruité. Pour conserver un taux de reconnaissance correcte, on est souvent ammené à traiter le signal au préalable. D'une part on peut filtrer le signal pour supprimer le maximum de bruit ambiant, par exemple en estimant la représentation spectrale du bruit dans les silence pour la soustraire au spect're de parole. D'autre part, il peut être également intéressant d'utiliser des coéfficients acoustiques connus pour être plus robuste aux bruits, tels que les P.L.P., les Rasta P.L.P., Delta énergie, Delta- Delta Cepstre,etc. Malgré toutes les techniques existantes, l'élimination de la musique ainsi que le traitement de paroles concomittantes restent des problèmes non résolus.

5.2.5 Détection du signal de parole

Un autre problème tout aussi important est la détection de parole. Il est probable qu' une analyse simple du signal acoustique permetrai de supprimer d'office la majeur partie du signal sonore ne présentant pas de contenu vocal. Cette analyse permettrai de réduire la quantité de signal sonore sur laquelle le reconnaisseur de parole devrait travailler. Cependant, peu d'études ont été effectuée dans le domaine.

5.2.6 Association à d'autres outils d'indexation

Comme nous l'avons signalé au début de cette thèse, cet outil d'indexation quoique viable en tant que tel, gagnerait beaucoup à être intégré dans un système global reprenant l'ensemble des outils d'indexation : Reconnaissance de locuteur, reconnaissance de caractères, suivit d'objet, détecteur de visage, etc.

Ces dernières sections donnerons, nous l'espérons, des directions possible à un travail futur qui permettrait d'améliorer les résultats obtenus jusqu'à présent.

'Eh bien indexez maintenant...

ANNEXE A

Algorithme de Baum-Welch : Preuve de convergence

Dans cette annexe sont reproduites la preuve de convergence, vers un maximum local de la vraisemblance et l'estimation des paramètres conduisant à cette optimisation.

Elle est reproduite ici pour facilité le lecteur dans la comparaison des preuves de convergences des algorithmes de Baum-Welch et REMAP.

A.1 Plan

Nous reproduisons ici la preuve de convergence, vers un maximum local de :

$$\max_{\Lambda} \prod_{i} P(X_{\mathcal{E},\, i} | M_{\mathcal{E},\, i}, \Lambda) ,$$

et l'estimation des paramètres conduisant à cette optimisation.

En considérant un modèle, M comme la concaténation de tous les modèles $M_{\mathcal{E},\, i}$ et la séquence acoustique X correspondant à la concaténation de toutes les séquences acoustiques $X_{\mathcal{E},\, i}$, le problème se résume à

$$\max_{\Lambda} P(X | M, \Lambda) \qquad \textbf{(EQ 89)}$$

La preuve de convergence est basée sur l'utilisation de la fonction auxiliaire de Baum, définie par :

$$F(\Lambda', \Lambda) = \sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \mathcal{log}(P(X, \gamma | M, \Lambda)) \qquad \textbf{(EQ 90)}$$

où γ est un chemin dans M pouvant être associé à la séquence X.

La première partie de la démonstration consiste à prouver :

$$F(\Lambda', \Lambda) \geq F(\Lambda', \Lambda') \Rightarrow P(X | M, \Lambda) \geq P(X | M, \Lambda') \qquad \textbf{(EQ 91)}$$

La seconde partie consiste à calculer les paramètres $\lambda_i \in \Lambda$ maximisant $F(\Lambda', \Lambda$

A.2 Première Partie

Pour prouver (EQ 91), il est nécessaire d'utiliser l'inégalité de Jensen sur la fonction concave $x\,log(\alpha x)$. Montrons rapidement la concavité de cette fonction et énonçons cette inégalité.

A.2.1 Concavité de "x log(ax)"

Montrons que la dérivée seconde est toujours positive, $\forall x > 0$.

$$\frac{d}{dx} x\,log(\alpha x) = log(\alpha x) + 1$$

$$\frac{d^2}{dx^2} x\,log(\alpha x) = \frac{1}{x}$$

A.2.2 Inégalité de Jensen

Enoncé.

Soit la fonction $g : \Omega \in \Re^M \to \Re$ convexe

Soit X une variable aléatoire telle que $E[\|X\|] < \infty$ et $P(X \in \Omega) = 1$

Alors,

$$g(E[X]) \geq E[g(X)].$$

Cette relation est stricte si g est strictement convexe et que X n'est pas concentrée en un point.

Le même raisonnement peut être tenu pour g concave, et dans ce cas, on a :

$$g(E[X]) \leq E[g(X)]$$

Preuve.

Le développement de $g(x)$ en série de Taylor autour du point $\mu = E[X]$ peut s'écrire :

$$g(x) = g(\mu) + g'(\mu)(x-\mu) + \frac{g''(\xi)(x-\xi)^2}{2}$$

où ξ est un point quelconque entre x et μ.

Etant donné que $g''(\xi) \geq 0$, on trouve

$$g(x) \geq g(\mu) + g'(\mu)(x - \mu)$$

Et ainsi :

$$g(X) \geq g(\mu) + g'(\mu)(X - \mu)$$

En prenant les espérances, on trouve :

$$E[g(X)] \geq g(\mu) + g'(\mu)E[(X - \mu)] \;=\; g(\mu),$$

ce qui montre l'inégalité.

A.2.3 Utilité de la fonction auxiliaire

Montrons que si un nouvel ensemble de paramètres, Λ, entraîne un accroissement de la fonction auxiliaire, alors la probabilité conditionnelle s'en trouvera également augmentée.

$$F(\Lambda', \Lambda) \geq F(\Lambda', \Lambda') \Rightarrow P(X|M, \Lambda) \geq P(X|M, \Lambda')$$

Dans le cas d'un tel ensemble, Λ, on a :

$$\begin{aligned} 0 &\geq F(\Lambda', \Lambda') - F(\Lambda', \Lambda) \\ &\geq \sum_{\gamma \in \Gamma} P(X, \gamma|M, \Lambda') \log\left(\frac{P(X, \gamma|M, \Lambda')}{P(X, \gamma|M, \Lambda)}\right) \end{aligned}$$

Or, la probabilité sur un chemin donné est inférieure ou égale à celle sur tous les chemins :

$$P(X, \gamma|M, \Lambda) \leq P(X|M, \Lambda), \forall(\gamma \in \Gamma).$$

On peut donc minorer chaque produit :

$$\log\left(\frac{P(X, \gamma|M, \Lambda')}{P(X, \gamma|M, \Lambda)}\right) \geq \log\left(\frac{P(X, \gamma|M, \Lambda')}{P(X|M, \Lambda)}\right)$$

et sachant que $P(X, \gamma|M, \Lambda') \geq 0$, la somme est donc également minorée.

Donc

$$0 \geq \sum_{\gamma \in \Gamma} P(X, \gamma|M, \Lambda') \log\left(\frac{P(X, \gamma|M, \Lambda')}{P(X|M, \Lambda)}\right).$$

En posant $\alpha = \dfrac{1}{P(X|M, \Lambda)}$, et en utilisant l'inégalité de Jensen appliquée à la fonction concave $x \log \alpha x$, on obtient :

$$0 \geq \left(\sum_{\gamma \in \Gamma} P(X, \gamma_i | M, \Lambda') \right) \log \left(\frac{\sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda')}{P(X|M, \Lambda)} \right)$$

$$\geq P(X|M, \Lambda') \log \left(\frac{P(X|M, \Lambda')}{P(X|M, \Lambda)} \right)$$

Or, $P(X|M, \Lambda') \geq 0$, et ainsi :

$$0 \geq \log \left(\frac{P(X|M, \Lambda')}{P(X|M, \Lambda)} \right).$$

Compte tenu du domaine positif de $\log x$, on a :

$$1 \geq \frac{P(X|M, \Lambda')}{P(X|M, \Lambda)}$$

Et finalement :

$$P(X|M, \Lambda) \geq P(X|M, \Lambda')$$

A.3 *Deuxième partie*

La seconde partie consiste à déterminer les nouveaux paramètres $\lambda_i \in \Lambda$ qui maximisent $F(\Lambda', \Lambda)$.

A.3.1 Définition des paramètres

Les paramètres λ_i sont de deux types : ceux définissant les probabilités d'émission et ceux définissant les probabilités de transition.

Pour les probabilités de transitions, le modèle markovien permet de les définir directement, et on les notera :

$$\begin{aligned} \lambda_{tr(i,k)} &= P(q_k^t | q_i^{t-1}) \qquad \forall t \\ &= P(q_k | \bar{q_i}) \end{aligned}$$

avec les contraintes suivantes :

$$\sum_{k=1}^{K} \lambda_{tr(i,k)} = 1 \qquad \forall i = 1, \ldots, K. \tag{EQ 92}$$

Pour les probabilités d'émission, on considérera ici chaque état associé à une densité de probabilité gaussienne d'ordre N où la matrice de covariance est supposée diagonale :

$$p(x|q_k) = \prod_{n=1}^{N} \frac{1}{\sigma_{k,n}\sqrt{2\pi}} e^{-\frac{(x_n - \mu_{k,n})^2}{2\sigma_{k,n}^2}}. \tag{EQ 93}$$

Dès lors, les paramètres des probabilités d'émission seront les $\mu_{k,n}$ et $\sigma_{k,n}$, $\forall n = 1, \ldots, N$ et $\forall k = 1, \ldots, K$.

A.3.2 Optimisation

On utilise les multiplicateurs de Lagrange, l_ι, pour insérer les contraintes (EQ 92) dans la maximisation de $F(\Lambda', \Lambda)$ et la nouvelle relation à maximiser est définie par :

$$F°(\Lambda', \Lambda) = F(\Lambda', \Lambda) + \sum_{i=1}^{K} l_i\left(1 - \sum_{k=1}^{K} \lambda_{tr(i,k)}\right). \qquad \textbf{(EQ 94)}$$

Cette maximisation impose l'annulation de la dérivée première en fonction de tous les paramètres :

$$\frac{\partial}{\partial \lambda_{tr(i,j)}} F°(\Lambda', \Lambda) = \frac{\partial}{\partial \lambda_{tr(i,j)}} F(\Lambda', \Lambda) + l_i = 0 \qquad ,\forall \lambda_{tr(i,j)} \in \Lambda \qquad \textbf{(EQ 95)}$$

$$\frac{\partial}{\partial \mu_{k,n}} F°(\Lambda', \Lambda) = \frac{\partial}{\partial \mu_{k,n}} F(\Lambda', \Lambda) = 0 \qquad ,\forall \mu_{k,n} \in \Lambda \qquad \textbf{(EQ 96)}$$

$$\frac{\partial}{\partial \sigma_{k,n}} F°(\Lambda', \Lambda) = \frac{\partial}{\partial \sigma_{k,n}} F(\Lambda', \Lambda) = 0 \qquad ,\forall \sigma_{k,n} \in \Lambda \qquad \textbf{(EQ 97)}$$

A.3.3 Estimation des paramètres de transition

Pour les probabilités de transition, on a successivement :

$$\begin{aligned} \frac{\partial}{\partial \lambda_{tr(i,k)}} F(\Lambda', \Lambda) &= \frac{\partial}{\partial \lambda_{tr(i,k)}}\left(\sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \mathit{log}(P(X, \gamma | M, \Lambda))\right) \\ &= \sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \frac{\partial}{\partial \lambda_{tr(i,k)}} \mathit{log}(P(X, \gamma | M, \Lambda)) \end{aligned} \qquad \textbf{(EQ 98)}$$

Si l'on note $\lambda^t_{tr(i,k)} = P(q^t_k \middle| q^{t-1}_i)$, on a :

$$\frac{\partial}{\partial \lambda_{tr(i,k)}} = \sum_{t=1}^{T} \frac{\partial}{\partial \lambda^t_{tr(i,k)}} \qquad \textbf{(EQ 99)}$$

Or, si on note par $\gamma^t_{i,k} = \left\{\gamma^{1,t-2}, q^{t-1}_i, q^t_j, \gamma^{t,T}\right\}$, tout chemin passant par q^{t-1}_i et q^t_j,

la probabilité associée à ce chemin peut se décomposer en :

$$\begin{aligned} P(X, \gamma^t_{i,k} \middle| M, \Lambda) &= P(X^{t-2}_1, \gamma^{1,t-2} \middle| M, \Lambda) P(x_{t-1} \middle| q^{t-1}_i) P(q^t_k \middle| q^{t-1}_i) P(X^T_t, \gamma^{t,T} \middle| M, \Lambda) \ . \\ &= P(X^{t-2}_1, \gamma^{1,t-2} \middle| M, \Lambda) P(x_{t-1} \middle| q^{t-1}_i) \lambda^t_{tr(i,k)} P(X^T_t, \gamma^{t,T} \middle| M, \Lambda) \end{aligned}$$

Le logarithme de celle-ci se décompose donc en :

$$
\begin{aligned}
\log P(X, \gamma_{i,k}^{t} \mid M, \Lambda) = {} & \log P(X_1^{t-2}, \gamma^{1,t-2} \mid M, \Lambda) \\
& + \log P(x_{t-1} \mid q_i^{t-1}) \\
& + \log \lambda_{tr(i,k)}^{t} \\
& + \log P(X_t^T, \gamma^{t,T} \mid M, \Lambda)
\end{aligned}
$$

Sa dérivée s'exprime directement par :

$$
\begin{aligned}
\frac{\partial}{\partial \lambda_{tr(i,k)}^{t}} \log P(X, \gamma_{i,k}^{t} \mid M, \Lambda) &= \frac{\partial}{\partial \lambda_{tr(i,k)}^{t}} \log \lambda_{tr(i,k)}^{t} \\
&= \frac{1}{\lambda_{tr(i,k)}^{t}} \\
&= \frac{1}{\lambda_{tr(i,k)}}
\end{aligned}
$$

Pour tous les chemins, on peut écrire;

$$
\frac{\partial}{\partial \lambda_{tr(i,k)}^{t}} \log P(X, \gamma \mid M, \Lambda) = \frac{1}{\lambda_{tr(i,k)}} \delta_{\gamma \in \gamma_{i,k}^{t}},
$$

$$
\text{où} \left(\begin{aligned} \delta_{\gamma \in \gamma_{i,k}^{t}} &= 1 \ \text{ si } \gamma \in \gamma_{i,k}^{t} \\ &= 0 \qquad \text{sinon} \end{aligned} \right.
$$

En injectant cette dernière équation et **(EQ 99)** dans **(EQ 98)**, on obtient :

$$
\begin{aligned}
\frac{\partial}{\partial \lambda_{tr(i,k)}} F(\Lambda', \Lambda) &= \sum_{\gamma \in \Gamma} P(X, \gamma \mid M, \Lambda') \sum_{t=1}^{T} \frac{1}{\lambda_{tr(i,k)}} \delta_{\gamma \in \gamma_{i,k}^{t}} \\
&= \frac{\sum_{t=1}^{T} \sum_{\gamma \in \Gamma} P(X, \gamma \mid M, \Lambda') \delta_{\gamma \in \gamma_{i,k}^{t}}}{\lambda_{tr(i,k)}} \\
&= \frac{\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^{t} \mid M, \Lambda')}{\lambda_{tr(i,k)}}
\end{aligned} \qquad \textbf{(EQ 100)}
$$

Finalement, **(EQ 95)** devient donc :

$$\frac{\partial}{\partial\lambda_{tr(i,j)}}F^{\circ}(\Lambda',\Lambda) = \frac{\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t \middle| M, \Lambda')}{\lambda_{tr(i,k)}} + l_i = 0 \qquad ,\forall\lambda_{tr(i,j)} \in \Lambda \qquad \text{(EQ 101)}$$

$\lambda_{tr(i,k)}$ peut donc s'exprimer en fonction du lagrangien et vaut :

$$\lambda_{tr(i,k)} = \frac{\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t \middle| M, \Lambda')}{l_i} \qquad ,\forall\lambda_{tr(i,j)} \in \Lambda \qquad \text{(EQ 102)}$$

En sommant sur k pour exprimer les contraintes, on a :

$$\sum_{k=1}^{K} \lambda_{tr(i,k)} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t \middle| M, \Lambda')}{l_i} = 1 .$$

Ce qui détermine l_i, et le fixe à :

$$l_i = \sum_{k=1}^{K}\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t \middle| M, \Lambda') = \sum_{t=1}^{T} P(X, q_i^{t-1} \middle| M, \Lambda') \qquad \text{(EQ 103)}$$

En injectant **(EQ 103)** dans **(EQ 102)**, on a finalement :

$$\boxed{\lambda_{tr(i,k)} = \frac{\sum_{t=1}^{T} P(X, q_i^{t-1}, q_j^t \middle| M, \Lambda')}{\sum_{t=1}^{T} P(X, q_i^{t-1} \middle| M, \Lambda')} \qquad ,\forall\lambda_{tr(i,j)} \in \Lambda} \qquad \text{(EQ 104)}$$

A.3.4 Estimation des moyennes

En partant de **(EQ 96)**,on a directement :

$$\frac{\partial}{\partial \mu_{k,n}} F(\Lambda', \Lambda) = \sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \frac{\partial}{\partial \mu_{k,n}} \log(P(X, \gamma | M, \Lambda)) = 0 \quad \text{(EQ 105)}$$

Or, la dérivée partielle $\frac{\partial}{\partial \mu_{k,n}}$ peut être décomposée pour tenir compte du temps :

$$\frac{\partial}{\partial \mu_{k,n}} = \sum_{t=1}^{T} \frac{\partial}{\partial \mu_{k,n}^{t}}, \quad \text{(EQ 106)}$$

où $\mu_{k,n}^{t}$, signifie $\mu_{k,n}$ employé en t.

En notant $\gamma_k^t = \left\{ \gamma^{1,t-1}, q_k^t, \gamma^{t+1,T} \right\} \in \Gamma_k^t$ tous les chemins, passant par q_k^t,

où le dernier état de $\gamma^{1,t-1}$ sera noté q_γ^{t-1}, et le premier état de $\gamma^{t+1,T}$ sera noté q_γ^{t+1},

le logarithme de $P(X, \gamma_k^t | M, \Lambda)$ peut être développé en :

$$\begin{aligned}
\log P(X, \gamma_k^t | M, \Lambda) = {} & \log P(X_1^{t-1}, \gamma^{1,t-1} | M, \Lambda) \\
& + \log P(q_k^t | q_\gamma^{t-1}) \\
& + \log p(x_t | q_k^t) \\
& + \log P(q_\gamma^{t+1} | q_k^t) \\
& + \log P(X_{t+1}^{T}, \gamma^{t+1,T} | q_k^t, M, \Lambda)
\end{aligned}$$

et sa dérivée conduit directement à :

$$\frac{\partial}{\partial \mu_{k,n}^{t}} \log P(X, \gamma_k^t | M, \Lambda) = \frac{\partial}{\partial \mu_{k,n}^{t}} \log p(x_t | q_k^t)$$

en se basant sur l'équation **(EQ 93)**, on trouve directement :

$$\frac{\partial}{\partial \mu_{k,n}^{t}} \log P(X, \gamma_k^t | M, \Lambda) = \frac{(x_n^t - \mu_{k,n})}{\sigma_{k,n}^2}$$

Pour les autres chemins ne passant pas par q_k^t, la dérivée est forcément nulle, et on peut écrire en toute généralité :

$$\frac{\partial}{\partial \mu_{k,n}^t} \log P(X, \gamma_k^t \middle| M, \Lambda) = \frac{(x_n^t - \mu_{k,n})}{\sigma_{k,n}^2} \delta_{\gamma \in \Gamma_k^t}$$

Et en injectant cette dernière équation dans **(EQ 105)**, en tenant compte de **(EQ 106)**, on a :

$$\begin{aligned} 0 &= \sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \sum_{t=1}^{T} \frac{(x_n^t - \mu_{k,n})}{\sigma_{k,n}^2} \delta_{\gamma \in \Gamma_k^t} \\ &= \sum_{t=1}^{T} \sum_{\gamma \in \Gamma} P(X, \gamma | M, \Lambda') \frac{(x_n^t - \mu_{k,n})}{\sigma_{k,n}^2} \delta_{\gamma \in \Gamma_k^t} \\ &= \sum_{t=1}^{T} P(X, q_k^t \middle| M, \Lambda')(x_n^t - \mu_{k,n}) \end{aligned}$$

On en déduit aisément :

$$\boxed{\mu_{k,n} = \frac{\sum_{t=1}^{T} x_n^t P(X, q_k^t | M, \Lambda')}{\sum_{t=1}^{T} P(X, q_k^t | M, \Lambda')}} \quad \textbf{(EQ 107)}$$

A.3.5 Estimation des variances

Un schéma identique, mène à

$$\boxed{\sigma_{k,n} = \frac{\sum_{t=1}^{T} (x_n^t - \mu_{k,n})^2 P(X, q_k^t | M, \Lambda')}{\sum_{t=1}^{T} P(X, q_k^t | M, \Lambda')}} \quad \textbf{(EQ 108)}$$

Si l'on remarque que :

$$\frac{\partial}{\partial \sigma_{k,n}^{t}} log P(X, \gamma_{k}^{t} \mid M, \Lambda) = \frac{-1}{\sigma_{k,n}^{t}} + \frac{(x_{n}^{t} - \mu_{k,n})^{2}}{\sigma_{k,n}^{2}}$$

ANNEXE B

Algorithme "REMAP" : Preuve de convergence

Dans cette annexe sont repris la preuve de convergence vers un maximum local de la probabilité globale a posteriori et l'estimation des paramètres conduisant à cette optimisation.

Cette preuve de convergence est issue des travaux de H. Bourlard, Y. Konig et N. Morgan, [BOU95].

Elle est reproduite ici pour facilité le lecteur dans la comparaison des preuves de convergences des algorithmes de Baum-Welch et REMAP.

B.1 Plan

Nous reproduisons ici la preuve de convergence vers un maximum local de :

$$\max_{\Lambda}\prod_{i} P(M_{\mathcal{E},\, i} | X_{\mathcal{E},\, i}, \Lambda),$$

et l'estimation des paramètres conduisant à cette optimisation.

En considérant un modèle M comme la concaténation de tous les modèles $M_{\mathcal{E},\, i}$ et la séquence acoustique X correspondant à la concaténation de toutes les séquences acoustiques $X_{\mathcal{E},\, i}$, le problème se résume directement à

$$\max_{\Lambda} P(M|X, \Lambda). \qquad \textbf{(EQ 109)}$$

Le schéma utilisé est identique à celui de l'annexe A.

La preuve de convergence est basée sur l'utilisation de la fonction auxiliaire définie par :

$$F(\Lambda', \Lambda) = \frac{1}{P(M|X, \Lambda')} \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \mathcal{log}(P(\gamma, M|X, \Lambda)), \qquad \textbf{(EQ 110)}$$

où Γ peut être réduit à l'ensemble des chemins de M pouvant être associés à la séquence X.

La première partie de la démonstration consiste à prouver :

$$F(\Lambda', \Lambda) \geq F(\Lambda', \Lambda') \Rightarrow P(M|X, \Lambda) \geq P(M|X, \Lambda') \qquad \textbf{(EQ 111)}$$

La seconde partie consiste à montrer que la nouvelle estimation de $P(q_i^t | x^t, q_k^{t-1}, M, \Lambda)$ par :

$$P(q_i^t | x^t, q_k^{t-1}, M, \Lambda) = P(q_i^t | X, q_k^{t-1}, M, \Lambda') \qquad \textbf{(EQ 112)}$$

maximise $F(\Lambda', \Lambda)$.

La troisième partie montre que le réseau de neurones peut être utilisé pour estimer correctement les probabilités de transition conditionnelles.

B.2 Première Partie

B.2.1 Utilité de la fonction auxiliaire

Montrons que si un nouvel ensemble de paramètres Λ entraîne un accroissement de la fonction auxiliaire, alors la probabilité a posteriori s'en trouvera également augmentée.

$$F(\Lambda', \Lambda) \geq F(\Lambda', \Lambda') \Rightarrow P(M|X, \Lambda) \geq P(M|X, \Lambda'),$$

$$F(\Lambda', \Lambda) = \frac{1}{P(M|X, \Lambda')} \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \log(P(\gamma, M|X, \Lambda)). \quad \textbf{(EQ 113)}$$

Dans le cas d'un tel ensemble Λ on a :

$$\begin{aligned} 0 &\geq F(\Lambda', \Lambda') - F(\Lambda', \Lambda) \\ &\geq \frac{1}{P(M|X, \Lambda')} \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \log\left(\frac{P(\gamma, M|X, \Lambda')}{P(\gamma, M|X, \Lambda)}\right). \end{aligned}$$

La probabilité $P(M|X, \Lambda')$ étant positive, on a directement :

$$0 \geq \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \log\left(\frac{P(\gamma, M|X, \Lambda')}{P(\gamma, M|X, \Lambda)}\right). \quad \textbf{(EQ 114)}$$

Or, la probabilité sur un chemin donné est inférieure ou égale à celle sur tous les chemins :

$$P(\gamma, M|X, \Lambda) \leq \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda) \leq P(M|X, \Lambda) \qquad , \forall \gamma \in \Gamma.$$

On peut donc minorer chaque produit :

$$\log\left(\frac{P(\gamma, M|X, \Lambda')}{P(\gamma, M|X, \Lambda)}\right) \geq \log\left(\frac{P(\gamma, M|X, \Lambda')}{P(M|X, \Lambda)}\right). .$$

Sachant que $P(\gamma, M|X, \Lambda') \geq 0$, la somme de l'équation **(EQ 114)** est également minorée, et on a :

$$0 \geq \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \log\left(\frac{P(\gamma, M|X, \Lambda')}{P(M|X, \Lambda)}\right).$$

En posant $\alpha = \dfrac{1}{P(M|X, \Lambda)}$, et en utilisant l'inégalité de Jensen appliquée à la fonction concave $x \log \alpha x$, (cf. A.2.1 et A.2.2.) on obtient :

$$0 \geq \left(\sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda')\right) \log\left(\frac{\sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda')}{P(M|X, \Lambda)}\right)$$

$$\geq P(M|X, \Lambda') \log\left(\frac{P(M|X, \Lambda')}{P(M|X, \Lambda)}\right).$$

Or, $P(M|X, \Lambda') \geq 0$, et on a ainsi :

$$0 \geq \log\left(\frac{P(M|X, \Lambda')}{P(M|X, \Lambda)}\right).$$

Compte tenu du domaine positif de $\log x$, on a également :

$$1 \geq \frac{P(M|X, \Lambda')}{P(M|X, \Lambda)},$$

et finalement :

$$\boxed{P(M|X, \Lambda) \geq P(M|X, \Lambda')}$$

B.3 Deuxième partie

La seconde partie consiste à déterminer les nouveaux paramètres $\lambda_i \in \Lambda$ qui maximisent $F(\Lambda', \Lambda)$ et contribue ainsi à l'augmentation de la probabilité a posteriori globale, $P(M|X, \Lambda)$.

B.3.1 Définition des paramètres

Les seuls paramètres du modèle sont ici les K^2T probabilités de transition conditionnelles $P(q_k^t | x^t, q_i^{t-1}, M, \Lambda)$ que l'on notera, par esprit de concision :

$$\lambda_{i,k}^t = P(q_k^t | x^t, q_i^{t-1}, M, \Lambda).$$

Ces paramètres sont reliés par KT contraintes :

$$\sum_{k=1}^{K} \lambda_{i,k}^t = 1 \qquad \forall i = 1, \ldots, K; t = 1, \ldots, T. \tag{EQ 115}$$

B.3.2 Optimisation

On utilise les multiplicateurs de Lagrange $l_{i,t}$ pour insérer les contraintes (EQ 115) dans la maximisation de $F(\Lambda', \Lambda)$. La nouvelle relation à maximiser est donc définie par :

$$F^\circ(\Lambda', \Lambda) = F(\Lambda', \Lambda) + \sum_{i=1}^{K} \sum_{t=1}^{N} l_{i,t}\left(1 - \sum_{k=1}^{K} \lambda_{i,k}^t\right). \tag{EQ 116}$$

Cette maximisation impose l'annulation de la dérivée première de (EQ 116) en fonction de tous les paramètres :

$$\frac{\partial}{\partial \lambda_{i,k}^t} F^\circ(\Lambda', \Lambda) = \frac{\partial}{\partial \lambda_{i,k}^t} F(\Lambda', \Lambda) - l_{i,t} = 0 \qquad , \forall \lambda_{i,k}^t \in \Lambda. \tag{EQ 117}$$

En explicitant $F(\Lambda', \Lambda)$ à l'aide de (EQ 113), on a :

$$\frac{\partial}{\partial\lambda_{i,k}^{t}}F(\Lambda',\Lambda) = \frac{\partial}{\partial\lambda_{i,k}^{t}}\left(\frac{1}{P(M|X,\Lambda')}\sum_{\gamma\in\Gamma}P(\gamma,M|X,\Lambda')\log(P(\gamma,M|X,\Lambda))\right) \qquad \textbf{(EQ 118)}$$
$$= \frac{1}{P(M|X,\Lambda')}\sum_{\gamma\in\Gamma}P(\gamma,M|X,\Lambda')\frac{\partial}{\partial\lambda_{i,k}^{t}}\log(P(\gamma,M|X,\Lambda)).$$

Or, si on note par $\gamma_{i,k}^{t} = \left\{\gamma^{1,t-2}, q_i^{t-1}, q_k^{t}, \gamma^{t,T}\right\}$, tout chemin passant par l'état q_i au temps $t-1$ et q_k au temps t, la probabilité associée à ce chemin peut se décomposer en :

$$P(\gamma_{i,k}^{t}, M\middle|X,\Lambda) = P(M|\gamma_{i,k}^{t},X,\Lambda)P(\gamma_{i,k}^{t}\middle|X,M,\Lambda) \qquad \textbf{(EQ 119)}$$
$$= P(M|\gamma_{i,k}^{t},X,\Lambda)P(\gamma^{1,t-1}\middle|X_1^{t-1},\Lambda)P(q_k^{t}\middle|x^{t},q_i^{t-1},\Lambda)P(\gamma^{t+1,T}\middle|X_{t+1}^{T},\Lambda)$$
$$= P(M|\gamma_{i,k}^{t},X,\Lambda)P(\gamma^{1,t-1}\middle|X_1^{t-1},\Lambda)\lambda_{i,k}^{t}P(\gamma^{t+1,T}\middle|X_{t+1}^{T},\Lambda).$$

Le logarithme de celle-ci se décompose donc en quatre termes :

$$\log P(\gamma_{i,k}^{t}, M\middle|X,\Lambda) = \log P(M|\gamma_{i,k}^{t},X,\Lambda) + \log P(\gamma^{1,t-1}\middle|X_1^{t-1},\Lambda) + \log\lambda_{i,k}^{t} + \log P(\gamma^{t+1,T}\middle|X_{t+1}^{T},\Lambda) \quad .$$

La dérivée partielle pour ce chemin s'écrit donc :

$$\frac{\partial}{\partial\lambda_{i,k}^{t}}\log P(\gamma_{i,k}^{t}, M\middle|X,\Lambda) = \frac{\partial}{\partial\lambda_{i,k}^{t}}\log\lambda_{i,k}^{t} = \frac{1}{\lambda_{i,k}^{t}}$$

Pour tous les chemins γ on peut écrire :

$$\frac{\partial}{\partial\lambda_{i,k}^{t}}\log P(\gamma,M|X,\Lambda) = \frac{1}{\lambda_{i,k}^{t}}\delta_{\gamma\in\gamma_{i,k}^{t}},$$

où $\delta_{\gamma\in\gamma_{i,k}^{t}} = \begin{cases}1 \text{ si } \gamma\in\gamma_{i,k}^{t} \\ 0 \text{ sinon}\end{cases}.$

En injectant cette dernière équation dans **(EQ 118)**, on obtient :

$$\frac{\partial}{\partial \lambda_{i,k}^{t}} F(\Lambda', \Lambda) = \frac{1}{P(M|X,\Lambda')} \sum_{\gamma \in \Gamma} P(\gamma, M|X, \Lambda') \frac{1}{\lambda_{i,k}^{t}} \delta_{\gamma \in \gamma_{i,k}^{t}}$$
$$= \frac{\sum_{\gamma \in \Gamma} P(\gamma, q_i^{t-1}, q_k^t, M \Big| X, \Lambda')}{\lambda_{i,k}^{t} P(M|X,\Lambda')}$$
$$= \frac{P(M, q_i^{t-1}, q_k^t \Big| X, \Lambda')}{\lambda_{i,k}^{t} P(M|X,\Lambda')}.$$

En insérant ce résultat dans l'équation **(EQ 117)**, on a :

$$\frac{P(M, q_i^{t-1}, q_k^t \Big| X, \Lambda')}{\lambda_{i,k}^{t} P(M|X,\Lambda')} - l_{i,t} = 0 \qquad ,\forall \lambda_{i,k}^{t} \in \Lambda.$$

Mettant $\lambda_{i,k}^{t}$ en évidence, on trouve :

$$\lambda_{i,k}^{t} = \frac{P(M, q_i^{t-1}, q_k^t \Big| X, \Lambda')}{l_{i,t} P(M|X,\Lambda')} \qquad ,\forall \lambda_{i,k}^{t} \in \Lambda. \tag{EQ 120}$$

En sommant sur k pour exprimer les contraintes, on a :

$$\sum_{k=1}^{K} \lambda_{i,k}^{t} = \frac{\sum_{k=1}^{K} P(M, q_i^{t-1}, q_k^t \Big| X, \Lambda')}{l_{i,t} P(M|X,\Lambda')} = \frac{P(M, q_i^{t-1} \Big| X, \Lambda')}{l_{i,t} P(M|X,\Lambda')} = \frac{P(q_i^{t-1} \Big| M, X, \Lambda')}{l_{i,t}} = 1$$

Et le lagrangien s'écrit donc :

$$l_{i,t} = P(q_i^{t-1} \Big| M, X, \Lambda'). \tag{EQ 121}$$

En injectant cette dernière dans l'équation **(EQ 120)**, on trouve :

$$
\begin{aligned}
\lambda_{i,k}^{t} &= \frac{P(M, q_i^{t-1}, q_k^t \middle| X, \Lambda')}{P(q_i^{t-1} \middle| M, X, \Lambda')P(M|X, \Lambda')} \\
&= \frac{P(M|X, \Lambda')P(q_i^{t-1}, q_k^t \middle| M, X, \Lambda')}{P(q_i^{t-1} \middle| M, X, \Lambda')P(M|X, \Lambda')} \\
&= \frac{P(q_k^t \middle| M, X, \Lambda')P(q_k^t \middle| q_i^{t-1}, X, M, \Lambda')}{P(q_i^{t-1} \middle| M, X, \Lambda')}.
\end{aligned}
$$

et finalement :

$$
\boxed{\lambda_{i,k}^{t} = P(q_k^t | q_i^{t-1}, X, M, \Lambda') \qquad ,\forall \lambda_{i,k}^{t} \in \Lambda}
\tag{EQ 122}
$$

Ces paramètres assurent donc la majoration de $P(M|X, \Lambda)$ par rapport à $P(M|X, \Lambda')$.

B.4 Troisième Partie

Le deuxième théorème nous a montré que la nouvelle estimation des probabilités conditionnelles $P(q_i^t | x^t, q_k^{t-1}, \Lambda)$ par :

$$P(q_i^t | x^t, q_k^{t-1}, \Lambda) = P(q_i^t | X, q_k^{t-1}, M, \Lambda') \quad \textbf{(EQ 123)}$$

maximise $F(\Lambda', \Lambda)$ et par conséquent assure une augmentation de la probabilité a posteriori :

$$P(M|X, \Lambda) \geq P(M|X, \Lambda').$$

Dans cette partie, nous montrons qu'un réseau de neurones peut être efficacement utilisé pour estimer les probabilités de transition conditionnelles.

Nous montrons également que son entraînement par une critère d'erreur approprié conduit à l'augmentation des fonctions axillaire, et dès lors à l'augmentation de la probabilité a posteriori globale.

B.4.1 Principe général

Notons $g(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda)$ la sortie du réseau de neurones obtenue quand on applique en entrée le couple $\left\{X_{t-c}^{t+d}, q_k^{t-1}\right\}$, alors que ce réseau est régi par l'ensemble des paramètres Λ.

Lors de l'apprentissage de ce couple, nous appliquons à la sortie les objectifs suivants :

$$O_{t,k} = P(q_i^t | X, q_k^{t-1}, M, \Lambda').$$

Idéalement, nous désirerions obtenir après convergence du réseau de neurones l'égalité entre la sortie du réseau et son objectif :

$$g(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda) = P(q_i^t | x^t, q_k^{t-1}, \Lambda) = O_{t,k} = P(q_i^t | X, q_k^{t-1}, M, \Lambda').$$

En effet, dans ce cas la relation **(EQ 123)** est vérifiée et nous assurons ainsi la maximisation de la fonction axillaire et l'augmentation des probabilités a posteriori globale. Par la suite, nous pourrions utiliser ce nouveau réseau de neurones pour estimer de nouveaux objectif et engager ainsi un processus itératif convergeant vers un maximum local de la probabilité a posteriori.

Cependant, cette égalité est respectée uniquement dans le cas où le réseau converge vers son minimum global et apprend exactement chaque paire (entrée, sortie). Cet objectif peut ne pas être atteint si l'algorithme d'apprentissage reste bloqué dans un minimum local ou si la configuration du réseau de neurones ne laisse pas assez de degrés de liberté pour apprendre toutes ces paires. Par ailleurs, si le réseau de neurones contient suffisamment de paramètres pour retenir toutes ces paires, il y aura risque de surentraînement et donc de mauvaise généralisation. Il faudra dès lors appliquer des méthodes de validation croisée pour arrêter l'algorithme avant de perdre ce pouvoir de généralisation.

Nous allons montrer, par le biais de la fonction auxiliaire, que l'algorithme d'apprentissage assure cependant, une augmentation directe des probabilités a posteriori. Pour ce faire, décrivons tout d'abord le critère d'erreur appliqué pour l'apprentissage et ensuite montrons cette propriété.

B.4.2 Critère d'erreur

Nous utiliserons ici un calcul d'erreur basé sur une approche entropique.

En toute généralité, pour une ensemble de N triplets $[e, s, o]$ comprenant l'entrée, la sortie et l'objectif, nous pouvons choisir comme erreur à minimiser :

$$E = \sum_{t=1}^{N} P(e_t) o_t \log \frac{o_t}{s_t},$$

où $P(e)$ représente la probabilité d'avoir l'échantillon d'entrée, e.

Dans notre cas, nous désirons appliqué en entrée le couple $\left\{ X_{t-c}^{t+d}, q_k^{t-1} \right\}$. Leur probabilité d'occurrence n'est pas constante et peut être estimée par :

$$\begin{aligned} P(q_k^{t-1}, X_{t-c}^{t+d} \mid X, M, \Lambda') &= P(X_{t-c}^{t+d} \mid X, M, \Lambda') P(q_k^{t-1} \mid X, X_{t-c}^{t+d}, M, \Lambda') \\ &= \frac{P(q_k^{t-1} \mid X, M, \Lambda')}{N}, \end{aligned} \qquad \text{(EQ 124)}$$

si l'on considère chaque vecteur acoustique équiprobable.

Compte tenu de cette probabilité d'occurrence des couples d'entrée, l'erreur totale peut maintenant être exprimée par :

$$E(\Lambda, \Lambda') = \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K} P(q_k^{t-1} | X, M, \Lambda')$$

$$\sum_{l=1}^{K} P(q_l^t | X, q_k^{t-1}, M, \Lambda') \log \frac{P(q_l^t | X, q_k^{t-1}, M, \Lambda')}{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda)}. \quad \textbf{(EQ 125)}$$

où Λ sont les paramètres modifiés au cours de l'apprentissage du réseau de neurones et Λ' les paramètres utilisés pour le calcul des objectifs et des probabilités d'occurrence des couple $\left\{X_{t-c}^{t+d}, q_k^{t-1}\right\}$.

L'ensemble des paramètres Λ' est supposé constant lors de l'entraînement du réseau de neurones.

B.4.3 Convergence

Nous montrons ici que la modification des paramètres Λ en vue de la minimisation de $E(\Lambda, \Lambda')$ entraîne une augmentation de la fonction auxiliaire $F(\Lambda, \Lambda')$, pour un Λ' fixé.

En notant Λ^o et Λ^n les paramètres avant et après les modifications effectuées pour l'entraînement du réseau de neurones, on cherche donc à montrer l'implication suivante :

$$E(\Lambda^n, \Lambda') \le E(\Lambda^o, \Lambda') \Rightarrow F(\Lambda', \Lambda^n) \ge F(\Lambda', \Lambda^o). \quad \textbf{(EQ 126)}$$

En partant des erreurs **(EQ 125)**, nous avons :

$$E(\Lambda^o, \Lambda') - E(\Lambda^n, \Lambda') \qquad \textbf{(EQ 127)}$$

$$\begin{aligned}
&= \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K} P(q_k^{t-1} \big| X, M, \Lambda')\sum_{l=1}^{K} P(q_l^t \big| X, q_k^{t-1}, M, \Lambda')\, log\frac{P(q_l^t \big| X, q_k^{t-1}, M, \Lambda')}{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^o)}\\
&- \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K} P(q_k^{t-1} \big| X, M, \Lambda')\sum_{l=1}^{K} P(q_l^t \big| X, q_k^{t-1}, M, \Lambda')\, log\frac{P(q_l^t \big| X, q_k^{t-1}, M, \Lambda')}{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^n)}\\
&= \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K} P(q_k^{t-1} \big| X, M, \Lambda')\sum_{l=1}^{K} P(q_l^t \big| X, q_k^{t-1}, M, \Lambda')\, log\frac{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^n)}{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^o)}\\
&= \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K} P(q_l^t, q_k^{t-1} \big| X, M, \Lambda')\, log\frac{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^n)}{g_l(X_{t-c}^{t+d}, q_k^{t-1}, \Lambda^o)}.
\end{aligned}$$

En partant de la définition de la fonction auxiliaire, **(EQ 110)**, nous avons, par contre :

$$F(\Lambda', \Lambda^n) - F(\Lambda', \Lambda^o) \qquad \textbf{(EQ 128)}$$

$$\begin{aligned}
&= \frac{1}{P(M|X, \Lambda')}\sum_{\gamma \in \Gamma} P(\gamma|X, M, \Lambda')\, log(P(\gamma|X, M, \Lambda^n))\\
&- \frac{1}{P(M|X, \Lambda')}\sum_{\gamma \in \Gamma} P(\gamma|X, M, \Lambda')\, log(P(\gamma|X, M, \Lambda^o))\\
&= \frac{1}{P(M|X, \Lambda')}\sum_{\gamma \in \Gamma} P(\gamma|X, M, \Lambda')\, log\frac{P(\gamma|X, M, \Lambda^n)}{P(\gamma|X, M, \Lambda^o)},
\end{aligned}$$

où Γ est l'ensemble des chemins possibles dans M.

En reprenant l'équation **(EQ 31)**, nous avons les égalités suivantes :

$$P(\gamma|X, M, \Lambda^n) = \prod_{t=1}^{N} P\left(q_{\gamma_t}^t \middle| q_{\gamma_{t-1}}^{t-1}, X_{t-c}^{t+d}, M, \Lambda^n\right) = \prod_{t=1}^{N} g_{\gamma_t}\left(X_{t-c}^{t+d}, q_{\gamma_{t-1}}^{t-1}, M, \Lambda^n\right)$$

et

$$P(\gamma|X, M, \Lambda^o) = \prod_{t=1}^{N} g_{\gamma_t}\left(X_{t-c}^{t+d}, q_{\gamma_{t-1}}^{t-1}, M, \Lambda^o\right).$$

On peut donc écrire **(EQ 128)** sous la forme :

$$F(\Lambda', \Lambda^n) - F(\Lambda', \Lambda^o) = \frac{1}{P(M|X, \Lambda')} \sum_{\gamma \in \Gamma} P(\gamma|X, M, \Lambda') \sum_{n=1}^{N} log \frac{g_{\gamma_t}\left(X_{t-c}^{t+d}, q_{\gamma_{t-1}}^{t-1}, \Lambda^n\right)}{g_{\gamma_t}\left(X_{t-c}^{t+d}, q_{\gamma_{t-1}}^{t-1}, \Lambda^o\right)}.$$

En insérant une somme sur tous les q_k^t et les q_l^{t-1}, on a successivement :

$$\sum_{\gamma \in \Gamma} P(\gamma|X, M, \Lambda') = \sum_{k=1}^{K} \sum_{l=1}^{K} \sum_{\gamma \in \Gamma} P(\gamma, q_k^t, q_l^{t-1} \big| X, M, \Lambda'),$$

et la somme sur γ peut se réduire à tous les chemins passant par les états q_k^t et q_l^{t-1}. Nous pouvons donc y remplacer toutes références à l'indice γ_t par k et γ_{t-1} par l.

La différence entre les deux fonctions auxiliaires est donc égale à :

$$\begin{aligned} & F(\Lambda', \Lambda^n) - F(\Lambda', \Lambda^o) \\ & = \frac{1}{P(M|X, \Lambda')} \sum_{k=1}^{K} \sum_{l=1}^{K} \sum_{\gamma \in \Gamma} P(\gamma, q_k^t, q_l^{t-1} \big| X, M, \Lambda') \sum_{n=1}^{N} log \frac{g_k(X_{t-c}^{t+d}, q_l^{t-1}, \Lambda^n)}{g_k(X_{t-c}^{t+d}, q_l^{t-1}, \Lambda^o)} \\ & = \frac{1}{P(M|X, \Lambda')} \sum_{k=1}^{K} \sum_{l=1}^{K} P(q_k^t, q_l^{t-1} \big| X, M, \Lambda') \sum_{n=1}^{N} log \frac{g_k(X_{t-c}^{t+d}, q_l^{t-1}, \Lambda^n)}{g_k(X_{t-c}^{t+d}, q_l^{t-1}, \Lambda^o)}. \end{aligned}$$

En comparant avec **(EQ 127)**, on trouve directement :

$$F(\Lambda', \Lambda^n) - F(\Lambda', \Lambda^o) = \frac{N}{P(M|X, \Lambda')} E(\Lambda^o) - E(\Lambda^n).$$

Pour un Λ' fixé, on a donc trivialement :

$$\boxed{E(\Lambda^n) \le E(\Lambda^o) \Rightarrow F(\Lambda', \Lambda^n) \ge F(\Lambda', \Lambda^o)} \qquad \textbf{(EQ 129)}$$

ANNEXE C

Classification probabiliste par réseaux de Neurones

Le rôle de cette annexe n'est pas de montrer le principe d'apprentissage des réseaux de neurones ni de démontrer la convergence de ceux-ci, mais plutôt de montrer sous quelles conditions ils peuvent être utilisés pour estimer les probabilités d'émission nécessaires dans une approche markovienne de reconnaissance.

Nous dérivons ici cette remarquable qualité d'estimation. Nous montrons qu'elle peut être déduite, en utilisant un critère d'entraînement basés sur l'erreur quadratique ou sur l'erreur entropique, mais aussi pour un espace d'entrée discret ou continu. De plus, nous montrons que les vecteurs d'apprentissage peuvent être construits de différentes manières tout en conduisant toujours à une estimation de ces probabilités.

C.1 Dans le cas discret, avec quantification

On suppose, dans ce cas, que les vecteurs acoustiques sont préalablement quantifiés avant d'être introduits dans le réseaux de neurones pour être classifiés. Montrons dans ce cas, qu'en appliquant à la sortie du réseau un vecteur d'apprentissage approprié et en utilisant un critère de minimisation déterminé, que l'on obtient, après apprentissage, une estimation de la probabilité a posteriori que le vecteur appartienne à une classe fixée.

C.1.1 Notations

$\mathcal{X} = \{x_1, \ldots, x_{N_X}\}$ L'ensemble des vecteurs acoustiques de la base de données d'entraînement, où $x_n \in \Re^{N_a}$.

$\mathcal{C} = \{c_1, \ldots, c_{N_C}\}$ L'ensemble des N_c *catégories* discrétisant l'espace d'entrée du réseau de neurones.

$\mathcal{Y} = \{y_1, \ldots, y_{N_C}\}$ L' ensemble des *vecteurs modèles* représentant chacun une catégorie.

$\mathcal{B} = \mathcal{F}(\mathcal{X}) \rightarrow \mathcal{C}$ La fonction associant chaque élément x_n à une et une seule catégorie c_i : $\mathcal{B}(n) = i$ où $n \in \{1, \ldots, N_{\mathcal{X}}\}$ et $i \in \{1, \ldots, N_{\mathcal{C}}\}$. Cette fonction est généralement une minimisation de distance telle que celle d'Euclide :

$$\mathcal{B}(n) = \underset{i}{argmin} \sum_{j=1}^{N_a} (x_{n,j} - y_{i,j})^2$$

$b(i) \in \Re^{N_e}$ Le *vecteur d'entrée*, représentant de façon unique la catégorie c_i et appliqué à l'entrée du réseau de neurones. Le vecteur d'entrée associé à x_n est donc $b(\mathcal{B}(n))$, qui sera noté $b(n)$ par abus de notation.

$\mathcal{Q} = \{q_1, \ldots, q_{N_Q}\}$ L'ensemble des *classes* dans lesquelles on veut répartir ces différents vecteurs.

$\mathcal{D} = \mathcal{F}(\mathcal{X}) \rightarrow \mathcal{Q}$ La fonction reliant chaque élément x_n à une et une seule classe q_k. $\mathcal{D}(n) = k$ où $n \in \{1, \ldots, N_{\mathcal{X}}\}$ et $k \in \{1, \ldots, N_{\mathcal{Q}}\}$. Cette fonction est supposée connue pour des vecteurs d'entraînement et inconnue pour des vecteurs de test.

$d(n) \in \Re^{N_Q}$ Le *vecteur d'apprentissage* appliqué à la sortie du réseau lorsque le vecteur $b(n)$ est appliqué à l'entrée.

$\Theta = \{\theta_1, \ldots, \theta_{N_\Theta}\}$ L'ensemble des paramètres modifiables du réseau (poids, biais,...).

$g(i, \Theta)$ Le *vecteur de sortie*, que l'on obtient quand on applique le vecteur d'entrée $b(i)$ à l'entrée du réseau de neurones et que l'on effectue une propagation avant. Le vecteur de sortie associé à x_n est donc $g(\mathcal{B}(n), \theta)$, qui sera noté $g(n, \Theta)$ par abus de notation.

C.1.2 Démonstration

Le réseau de neurones sera, d'après les notations, constitué de N_e entrées et de N_Q sorties.

L'erreur utilisée pour la rétro-propagation est définie comme une métrique entre le vecteur d'apprentissage $d(n)$ et le vecteur de sortie $g(n, \Theta)$.

Dans un premier temps, prenons l'erreur quadratique comme référence :

$$e(n) = \frac{1}{2}\sum_{l=1}^{N_Q}(g_l(n, \Theta) - d_l(n))^2 \quad \textbf{(EQ 130)}$$

L'erreur globale, pour tous les vecteurs acoustiques de la base de données, s'écrit :

$$E = \frac{1}{2}\sum_{n=1}^{N_X}\sum_{l=1}^{N_Q}(g_l(n, \Theta) - d_l(n))^2 \quad \textbf{(EQ 131)}$$

Il est bon de noter qu'à chaque x_n sont associées une et une seule catégorie c_i, par le biais de $\mathcal{B}$, et une et une seule classe q_k, par le biais de $\mathcal{D}$. Mais à une catégorie c_i donnée sont associés plusieurs x_n et donc probablement plusieurs classes différentes.

L'ensemble $\mathcal{X}$ peut être réparti en une union de sous-ensembles disjoints $\mathcal{X} = \bigcup_{k=1}^{N_Q}\bigcup_{i=1}^{N_C}\mathcal{X}_{ik}$, où chaque sous-ensemble $\mathcal{X}_{ik}$ contient l'ensemble des vecteurs acoustiques x_n associés à la catégorie c_i et à la classe q_k :

$$\mathcal{X}_{ik} = \{x_n | \mathcal{B}(n) = i \wedge \mathcal{D}(n) = k\}.$$

Pour chaque x_n d'un même sous-ensemble $\mathcal{X}_{ik}$, nous avons $g_l(n, \Theta) = g_l(i, \Theta)$.

En imposant $d(n)$ identique pour un même $\mathcal{X}_{ik}$, on peut noter $d(n) = d(i, k)$ et ainsi l'erreur $e(n) = \frac{1}{2}\sum_{l=1}^{N_Q}(g_l(i, \Theta) - d_l(i, k))^2 = e(i, k)$ est identique pour chaque x_n d'un même sous-ensemble $\mathcal{X}_{ik}$.

Si nous notons n_{ik} le nombre d'éléments dans l'ensemble $\mathcal{X}_{ik}$, nous pouvons sommer l'erreur totale sur les différents vecteurs acoustiques regroupés en sous-ensembles. On a ainsi :

$$E = \frac{1}{2}\sum_{i=1}^{N_C}\sum_{k=1}^{N_Q}\sum_{l=1}^{N_Q} n_{ik}(g_l(i, \Theta) - d_l(i, k))^2. \qquad \textbf{(EQ 132)}$$

Déterminons la valeur des $g_l(i, \Theta)$ qui minimisent E. Ces valeurs respectent donc la relation :

$$\frac{\partial E}{\partial g_l(i, \Theta)} = 0 \qquad , \forall(l, i) \qquad \textbf{(EQ 133)}$$

En injectant **(EQ 132)**, on obtient successivement :

$$\frac{\partial\left[\frac{1}{2}\sum_{s=1}^{N_C}\sum_{v=1}^{N_Q}\sum_{u=1}^{N_Q} n_{sv}(g_u(s, \Theta) - d_u(s, v))^2\right]}{\partial g_l(i, \Theta)} = 0 \qquad , \forall(l, i)$$

$$\frac{1}{2}\sum_{s=1}^{N_C}\sum_{v=1}^{N_Q}\sum_{u=1}^{N_Q} n_{sv}\frac{\partial(g_u(s, \Theta) - d_u(s, v))^2}{\partial g_l(i, \Theta)} = 0 \qquad , \forall(l, i).$$

Le minimum est donc atteint pour un ensemble de $g_l^{opt}(i, \Theta)$ vérifiant :

$$\sum_{v}^{N_Q} n_{iv}(g_l^{opt}(i, \Theta) - d_l(i, v)) = 0 \qquad \forall i = 1, \ldots, N_C ; \forall l = 1, \ldots, N_Q \qquad \textbf{(EQ 134)}$$

Il ne faut cependant pas perdre de vue que l'obtention d'un minimum global n'est pas assuré. En effet, cela dépend du degré de liberté du réseau qui doit être suffisamment grand (neurones en suffisance) pour permettre une recherche exhaustive dans l'espace des variables, mais dépend aussi de l'espace d'entrée qui doit être séparable. De plus l'algorithme mis en oeuvre doit être suffisamment robuste pour ne pas s'arrêter dans des minima locaux.

Supposons ces contraintes vérifiées. De **(EQ 134)**, on obtient directement :

$$\boxed{g_l^{opt}(i, \Theta) = \frac{\sum_v^{N_Q} n_{iv} d_l(i, v)}{\sum_v^{N_Q} n_{iv}}}. \qquad \textbf{(EQ 135)}$$

Montrons que $d_l(i, v)$ peut prendre différentes valeurs tout en permettant à $g_l^{opt}(i, \Theta)$ de conserver une relation directe avec la probabilité a posteriori.

C.1.3 Vecteur d'apprentissage indépendant de la catégorie

Soit $d_l(i, v)$, indépendant de la catégorie c_i.

Nous pouvons donc écrire $d_l(i, v) = d_l(v)$ et choisir $d_l(v)$ tel que

$$d_l(v) = A\delta_{lv} + B, \; -B \neq A. \qquad \textbf{(EQ 136)}$$

où $\begin{cases} \delta_{lv} = 1 \text{ si } l = v \\ \quad\;\; = 0 \text{ sinon} \end{cases}$

Dans ce cas, nous avons :

$$g_l^{opt}(i, \Theta) = \frac{\sum_v^{N_Q} n_{iv}(A\delta_{lv} + B)}{\sum_v^{N_Q} n_{iv}} = \frac{\sum_v^{N_Q} n_{iv}(A\delta_{lv} + B)}{\sum_v^{N_Q} n_{iv}} = A\frac{n_{il}}{\sum_v^{N_Q} n_{iv}} + B.$$

Par application de la théorie du dénombrement, on trouve directement :

$$g_l^{opt}(i,\Theta) = A\tilde{P}(q_l|c_i) + B. \quad \textbf{(EQ 137)}$$

Si nous choisissons $A = 1$ et $B = 0$, nous avons directement :

$$g_l^{opt}(i,\Theta) = \frac{\sum_v^{N_Q} n_{iv}\delta_{lv}}{\sum_v^{N_Q} n_{iv}} = \frac{n_{il}}{\sum_v^{N_Q} n_{iv}} = \tilde{P}(q_l|c_i). \quad \textbf{(EQ 138)}$$

C.1.4 Vecteur d'apprentissage indépendant de la classe

Soit $d_l(i,v)$, dépendant uniquement de la classe c_i.

Nous avons donc $d_l(i,v) = d_l(i)$.

Dans ce cas, nous pouvons choisir :

$$d_l(i) = \frac{n_{il}}{\sum_v^{N_Q} n_{iv}} = \tilde{P}(q_l|c_i) \quad \textbf{(EQ 139)}$$

$$g_l^{opt}(i,\Theta) = \frac{\sum_v^{N_Q} n_{iv}n_{il}}{\left(\sum_v^{N_Q} n_{iv}\right)^2} = \frac{n_{il}\sum_v^{N_Q} n_{iv}}{\left(\sum_v^{N_Q} n_{iv}\right)^2} = \frac{n_{il}}{\sum_v^{N_Q} n_{iv}} = \tilde{P}(q_l|c_i) \quad \textbf{(EQ 140)}$$

L'intérêt de cette représentation réside dans l'annulation de l'erreur lors de la convergence du système. En effet, en remplaçant **(EQ 139)** dans **(EQ 132)**, on obtient

$$E = \sum_{i=1}^{N_C}\sum_{k=1}^{N_Q}\sum_{l=1}^{N_Q} n_{ik}(g_l(i,\Theta) - \tilde{P}(q_l|c_i))^2. \quad \textbf{(EQ 141)}$$

Et lorsque $g_l(i,\Theta)$ tend vers son optimum, E s'annule.

C.1.5 Autres métriques pour le calcul de l'erreur

En se basant sur le critère d'entropie relative définie par [HERTZ92], on peut réécrire **(EQ 131)** par :

$$E = \frac{1}{2}\sum_{n=1}^{N_X}\sum_{l=1}^{N_Q}\left((1+d_l(n))\,\ln\left(\frac{1+d_l(n)}{1+g_l(n,\Theta)}\right)+(1-d_l(n))\,\ln\left(\frac{1-d_l(n)}{1-g_l(n,\Theta)}\right)\right) \qquad \text{(EQ 142)}$$

Qui se dérive de façon identique pour obtenir le même critère d'optimisation.

En effet, **(EQ 133)** devient :

$$\sum_{v}^{N_Q} n_{iv}\frac{(g_l(i,\Theta)-d_l(i,v))}{(1-g_l(i,\Theta))^2} = 0 \qquad ,\forall(l,i) \qquad \text{(EQ 143)}$$

et **(EQ 134)** reste identique.

Des résultats identiques peuvent être tirés de l'équation d'entropie relative définie par [BOUR94] :

$$E = \sum_{n=1}^{N_X}\sum_{l=1}^{N_Q}\left(d_l(n)\,\ln\left(\frac{d_l(n)}{g_l(n,\Theta)}\right)+(1-d_l(n))\,\ln\left(\frac{1-d_l(n)}{1-g_l(n,\Theta)}\right)\right). \qquad \text{(EQ 144)}$$

Dans ce cas **(EQ 133)** devient :

$$\sum_{v}^{N_Q} n_{iv}\frac{(g_l(i,\Theta)-d_l(i,v))}{(1-g_l(i,\Theta))g_l(i,\Theta)} = 0 \qquad ,\forall(l,i)\,, \qquad \text{(EQ 145)}$$

qui conduit également à **(EQ 134)**.

C.2 Dans le cas discret, sans quantification

Montrons maintenant, comment l'on peut généraliser les résultats obtenus dans le chapitre précédent dans le cas où nous n'avons pas de quantification en entrée.

C.2.1 Notations

$\mathcal{X} = \{x_1, \ldots, x_{N_X}\}$	L'ensemble des vecteurs acoustiques de la base de données d'entraînement, où $x_n \in \Re^{N_a}$.
$\mathcal{Q} = \{q_1, \ldots, q_{N_Q}\}$	L'ensemble des *classes* dans lesquelles on veut répartir ces différents vecteurs.
$P(q_k\|x_n)$	La probabilité que le vecteur acoustique x_n appartienne à la classe q_k.
$d(n) \in \Re^{N_Q}$	Le *vecteur d'apprentissage* appliqué à la sortie du réseau lorsque le vecteur x_n est appliqué à l'entrée.
$d(n, k) \in \Re^{N_Q}$	Le *vecteur d'apprentissage* appliqué à la sortie du réseau lorsque le vecteur x_n est appliqué à l'entrée et que la classe q_k lui est associée.
$\Theta = \{\theta_1, \ldots, \theta_{N_\Theta}\}$	L'ensemble des paramètres modifiables du réseau (poids, biais,...).
$g(n, \Theta)$	Le *vecteur de sortie*, lorsque l'on applique le vecteur x_n à l'entrée du réseau.

C.2.2 Démonstration

Le réseau de neurones sera dans ce cas, constitué de N_a entrées et de N_Q sorties.

Comme dans le cas discret, prenons tout d'abord l'erreur quadratique comme mesure entre le vecteur d'apprentissage $d(k)$ et le vecteur de sortie $g(n, \Theta)$.

Pour un x_n donné, en faisant l'hypothèse que la classe q_k lui est associée, l'erreur est donc :

$$e(n, k) = \frac{1}{2}\sum_{l=1}^{N_Q}(g_l(n, \Theta) - d_l(n, k))^2 \quad \textbf{(EQ 146)}$$

En relâchant cette dernière hypothèse, nous avons donc :

$$e(n) = \frac{1}{2}\sum_{k=1}^{N_Q}\sum_{l=1}^{N_Q} P(q_k|x_n)(g_l(n,\Theta) - d_l(n,k))^2$$

L'erreur globale, pour tous les vecteurs acoustiques de la base de données, s'écrit :

$$E = \frac{1}{2}\sum_{n=1}^{N_x}\sum_{k=1}^{N_Q}\sum_{l=1}^{N_Q} P(q_k|x_n)(g_l(n,\Theta) - d_l(n,k))^2 \tag{EQ 147}$$

Déterminons la valeur des $g_l(n, \Theta)$ qui minimise (EQ 134) E. Ces valeurs respectent la relation :

$$\frac{\partial E}{\partial g_l(n,\Theta)} = 0 \qquad ,\forall(l,n) \tag{EQ 148}$$

En injectant **(EQ 147)**, on obtient successivement :

$$\frac{\partial E}{\partial g_l(n,\Theta)} = \frac{\partial\left[\frac{1}{2}\sum_{s=1}^{N_x}\sum_{v=1}^{N_Q}\sum_{u=1}^{N_Q} P(q_v|x_s)(g_u(s,\Theta) - d_u(s,v))^2\right]}{\partial g_l(n,\Theta)} = 0 \qquad ,\forall(l,n),$$

$$\frac{1}{2}\sum_{s=1}^{Nx}\sum_{v=1}^{N_Q}\sum_{u=1}^{N_Q} P(q_v|x_s)\frac{\partial(g_u(s,\Theta) - d_u(s,v))^2}{\partial g_l(n,\Theta)} = 0 \qquad ,\forall(l,n),$$

De cette dernière équation, on peut déduire que le minimum global est atteint pour un ensemble de $g_l^{opt}(i, \Theta)$ vérifiant :

$$\sum_{v=1}^{N_Q} P(q_v|x_n)(g_l^{opt}(n,\Theta) - d_l(n,v)) = 0 \qquad \forall\begin{cases} n = 1, \ldots, N_x \\ l = 1, \ldots, N_Q \end{cases}. \tag{EQ 149}$$

Les mêmes remarques que dans le cas discret doivent être notées. L'obtention d'un minimum global n'est pas assurée. En effet, cela dépend du degré de liberté du réseau qui doit être suffisamment grand (neurones en suffisance) pour permettre une recherche exhaustive dans l'espace des variables. De plus l'algorithme mis en oeuvre doit être suffisamment robuste pour ne pas s'arrêter dans des minima locaux.

Supposons ces contraintes vérifiées. De **(EQ 134)**, on obtient directement :

$$g_l^{opt}(n, \Theta) = \sum_{v=1}^{N_Q} P(q_v | x_n) d_l(n, v) . \qquad \textbf{(EQ 150)}$$

Montrons que $d_l(n, v)$ peut prendre différentes valeurs tout en permettant à $g_l^{opt}(n, \Theta)$ de conserver une relation directe avec la probabilité a posteriori.

C.2.3 Soit chaque x_n associé à une classe unique

Dans ce cas, $d_l(n, v)$ dépend uniquement de la classe q_v associée à x_n.

Nous avons donc $d_l(n, v) = d_l(v)$ et nous pouvons choisir $d_l(v)$ tel que :

$$d_l(v) = A\delta_{lv} + B , -B \neq A . \qquad \textbf{(EQ 151)}$$

Où $\delta_{lv} = 1$ si $l = v$ et $\delta_{lv} = 0$ sinon.

Dans ce cas, nous avons :

$$g_l^{opt}(n, \Theta) = \sum_v^{N_Q} P(q_v | x_n)(A\delta_{lv} + B) = AP(q_l | x_n) + \sum_v^{N_Q} P(q_v | x_n) B = AP(q_l | x_n) + B$$

$$g_l^{opt}(n, \Theta) = AP(q_l | x_n) + B \qquad \textbf{(EQ 152)}$$

Si nous choisissons $A = 1$ et $B = 0$, nous avons directement :

$$g_l^{opt}(n, \Theta) = P(q_l | x_n) . \qquad \textbf{(EQ 153)}$$

C.2.4 Soit chaque x_n associé statistiquement à chaque classe

Dans ce cas, nous pouvons choisir :

$$d_l(n, v) = P(q_l | x_n)$$

$$g_l^{opt}(n, \Theta) = \sum_v^{N_Q} P(q_v | x_n) P(q_l | x_n) = P(q_l | x_n) \qquad \textbf{(EQ 154)}$$

L'intérêt de cette représentation réside dans l'annulation de l'erreur lors de la convergence du système. En effet, en remplaçant **(EQ 139)** dans **(EQ 132)**, on obtient

$$E = \frac{1}{2}\sum_{n=1}^{N_X}\sum_{k=1}^{N_Q}\sum_{l=1}^{N_Q} P(q_k|x_n)(g_l(n,\Theta) - P(q_l|x_n))^2. \qquad \textbf{(EQ 155)}$$

Et lorsque $g_l(n, \Theta)$ tend vers son optimum, E s'annule.

C.2.5 Autres métriques pour le calcul de l'erreur

De façon identique au chapitre précédent, ces dernières conclusions peuvent être étendues à d'autres critères d'erreur.

C.3 Dans le cas continu

Montrons maintenant, comment l'on peut généraliser les résultats obtenus dans la dernière section si nous n'avons pas de quantification en entrée.

C.3.1 Notations

$p(x)$	La densité de probabilité associée au vecteur acoustique
$\mathcal{P} \subset \Re^{N_a}$	Le support de x où $p(x) \neq 0$. Nous pouvons supposer aisément ce support fermé.
$P(q_k \vert x)$	La probabilité qu'un x donné soit associé à la classe q_k. Nous avons : $\int_{\mathcal{P}} p(x)dx = 1$ et $\sum_{k=1}^{N_Q} P(q_k \vert x) = 1$
$d(x, k) \in \Re^{N_Q}$	Le *vecteur d'apprentissage* appliqué à la sortie du réseau lorsque le vecteur x est appliqué à l'entrée et que la classe q_k lui est associée.
$g(x, \Theta)$	Le *vecteur de sortie*, lorsque l'on applique le vecteur x à l'entrée du réseau.

C.3.2 Démonstration

Le réseau de neurones sera, comme dans le cas discret sans quantification, constitué de N_a entrées et de N_Q sorties.

Comme dans les deux cas précédents, prenons tout d'abord l'erreur quadratique comme mesure entre le vecteur d'apprentissage $d(x, k)$ et le vecteur de sortie $g(x, \Theta)$.

Pour un x donné, sachant qu'il appartient à la classe q_k, écrivons l'erreur utilisée pour la rétropropagation comme étant :

$$e(x, k, \Theta) = \frac{1}{2}\sum_{l=1}^{N_Q} (g_l(x, \Theta) - d_l(x, k))^2. \qquad \textbf{(EQ 156)}$$

En supposant que l'on ne connaisse pas exactement son appartenance, mais la probabilité associée à chaque classe, cette erreur doit être pondérée, et on modifie **(EQ 156)** par :

$$e(x, \Theta) = \frac{1}{2} \sum_{l=1}^{N_Q} \sum_{k=1}^{N_Q} P(q_k | x)(g_l(x, \Theta) - d_l(x, k))^2 . \tag{EQ 157}$$

L'erreur moyenne totale peut être écrite :

$$E(\Theta) = \int_{\mathcal{P}} p(x) e(x, \Theta) dx \tag{EQ 158}$$

Recherchons maintenant le minimum de cette valeur en fonction des paramètres Θ.

Pour ce faire, dérivons $E(\Theta)$ en fonction de

$$\frac{\partial E(\Theta)}{\partial \theta_r} = \frac{\partial \int_{\mathcal{P}} p(x) e(x, \Theta) dx}{\partial \theta_r} = \int_{\mathcal{P}} \frac{\partial (p(x) e(x, \Theta))}{\partial \theta_r} dx = \int_{\mathcal{P}} p(x) \frac{\partial e(x, \Theta)}{\partial \theta_r} dx \tag{EQ 159}$$

Le passage de la dérivée partielle sous le signe intégral étant permis sous la condition d'avoir $e(x, \Theta)$ continûment dérivable en Θ et $\mathcal{P}$ un support borné.

Or, comme précédemment,

$$\frac{\partial e(x, \Theta)}{\partial \theta_r} = \sum_{l=1}^{N_Q} \frac{\partial e(x, \Theta)}{\partial g_l(x, \Theta)} \frac{\partial g_l(x, \Theta)}{\partial \theta_r} = 0 \qquad \forall r = 1, \ldots, N_\Theta ; \forall x \subset \mathcal{P} \tag{EQ 160}$$

Où l'on suppose $e(x, \Theta)$ continûment dérivable sur l'espace des Θ et $g_l(x, \Theta)$ dérivable sur θ_r.

Le premier facteur ne prend en considération que les sorties du réseau, indépendamment de la topologie du réseau dont l'effet est reporté dans le deuxième facteur.

En injectant **(EQ 157)**, on obtient successivement :

$$\frac{\partial e(x, \Theta)}{\partial \theta_r} = \frac{1}{2} \sum_{l=1}^{N_Q} \frac{\partial \left(\sum_{s=1}^{N_Q} \sum_{v=1}^{N_Q} P(q_v | x)(g_s(x, \Theta) - d_s(x, v))^2 \right)}{\partial g_l(x, \Theta)} \frac{\partial g_l(x, \Theta)}{\partial \theta_r} = 0 \qquad \forall r ,$$

$$\frac{\partial e(x,\Theta)}{\partial\theta_r} = \sum_{l=1}^{N_Q}\left(\sum_{v=1}^{N_Q} P(q_v|x)(g_l(x,\Theta) - d_l(x,k))\right)\frac{\partial g_l(x,\Theta)}{\partial\theta_r} = 0 \qquad \forall r$$

De cette dernière équation, on peut déduire qu'un minimum est atteint pour un ensemble de $g_l^{opt}(i,\Theta)$ vérifiant :

$$\sum_{v=1}^{N_Q} P(q_v|x)(g_l^{opt}(x,\Theta) - d_l(x,v)) = 0 \qquad \forall x;\forall l = 1,\ldots,N_Q. \qquad \textbf{(EQ 161)}$$

Il ne faut cependant pas perdre de vue que l'obtention d'un minimum global n'est pas assurée. En effet, cela dépend du degré de liberté du réseau qui doit être suffisamment grand (neurones en suffisance) pour permettre une recherche exhaustive dans l'espace des variables. De plus l'algorithme mis en oeuvre doit être suffisamment robuste pour ne pas s'arrêter dans des minima locaux.

Supposons ces contraintes vérifiées. De **(EQ 134)**, on obtient directement :

$$g_l^{opt}(x,\Theta) = \sum_{v}^{N_Q} P(q_v|x)d_l(x,v). \qquad \textbf{(EQ 162)}$$

Montrons que $d_l(x,v)$ peut prendre différentes valeurs tout en permettant à $g_l^{opt}(x,\Theta)$ de conserver une relation directe avec la probabilité a posteriori.

C.3.3 Soit $\mathcal{D} = \mathcal{F}(\mathcal{X}) \to \mathcal{Q}$, une fonction connue reliant tout élément x à une et une seul classe

Dans ce cas, $d_l(x,v)$ dépendant uniquement de la classe q_v associée à x.

Nous avons donc $d_l(x,v) = d_l(v)$ et nous pouvons choisir $d_l(v)$ tel que :

$$d_l(v) = A\delta_{lv} + B, -B \neq A. \qquad \textbf{(EQ 163)}$$

Où $\delta_{lv} = 1$ si $l = v$ et $\delta_{lv} = 0$ sinon.

Dans ce cas, nous avons :

$$g_l^{opt}(x,\Theta) = \sum_{v}^{N_Q} P(q_v|x)(A\delta_{lv} + B) = AP(q_l|x) + \sum_{v}^{N_Q} P(q_v|x)B = AP(q_l|x) + B$$

$$g_l^{opt}(x, \Theta) = AP(q_l|x) + B \tag{EQ 164}$$

Si nous choisissons $A = 1$ et $B = 0$, nous avons directement :

$$g_l^{opt}(x, \Theta) = P(q_l|x). \tag{EQ 165}$$

C.3.4 Soit chaque x associé statistiquement à chaque classe

Dans ce cas, nous pouvons choisir :

$$d_l(x, v) = P(q_l|x) \tag{EQ 166}$$

$$g_l^{opt}(x, \Theta) = \sum_{v}^{N_Q} P(q_v|x)P(q_l|x) = P(q_l|x) \tag{EQ 167}$$

L'intérêt de cette représentation réside dans l'annulation de l'erreur lors de la convergence du système. En effet, en remplaçant (EQ 139) dans (EQ 157), on obtient

$$E(\Theta) = \int_{\mathcal{P}} p(x)\left(\frac{1}{2}\sum_{l=1}^{N_Q}\sum_{k=1}^{N_Q} P(q_k|x)(g_l(x, \Theta) - P(q_l|x))^2\right)dx. \tag{EQ 168}$$

Et lorsque $g_l(x, \Theta)$ tend vers son optimum, $E(\Theta)$ s'annule.

Bibliographie

ADJ96: Adjoudani, A., Benoîît, C., *On the integration of auditory and visual parameters in an HMM-based ASR.* In Speechreading by humans and machines, models, systems and applications, Springer-Verlag Berlin , Vol 150, pp. 461-472, 1996

AIG94: Aigrin, P., Joly, P., *The Automatic Real-Time Analysis of Film Editing and Transition Effects and its Applications.* Computers & Graphics. Vol. 18, No. 1, 1994

AKU92: Akutsu, A., Tonomura, Y., Hashimoto, H., Ohba, Y., *Video Indexing Using Motion Vectors Data.* Proc SPIE, Vol. 1818, pp. 1522-1530, 1992

ALVA93: Alvarez-Cercadillo,J., Hernandez-Gomez, L.A., *Grammar learning and word spotting using recurrent neural networks.* EUROSPEECH, pp. 1277-1280, 1993

ANG94: Angelini, B, Brugnara, F., Falavigna, D., Giuliani, D., Gretter, R., Omologo, M., *Speaker independent continuous speech recognition using an acoustic-phonetic italian corpus.* ICSLP, pp. 1391-1394, 1994

ARM93: Arman, F., Hsu, A., Chiu, M.-Y., *Feature Management for Large Video databases. Storage and Retrieval for Image and Video Databases.* part II of IS&T Symposium on Electronic Imaging Science and Technology, San José, Vol. 1908 February, 1993

BAHL75: Bahl, R.R., Jelinek, F., *Decoding for channels with insertions, deletions and substitutions with application to speech recognition.* Transs IEEE IT-21, pp. 404-411, 1975

BAHL84: Bahl, L.R., Das, S. K., DeSousa, P. V., *Some experiments with large vocabulary isolated word sentence recognition.* ICASSP, vol 2, n 26.5, 1984

BAHL86: Bahl, L. R., Brown, P. F., de Souza, P. V., Mercer, R. L., *Maximum Mutual information Estimation in Hidden Markov Model Parameters for Speech Recognition.* Proc. ICASSP, pp. 81-84, 1986

BAKER75: Baker, J. K., *The DRAGON system - an overview.* ICASSP,vol 23, no. 1, pp. 24-29, 1975

BAKIS76: Bakis, R., *Continuous speech recognition via centisecond acoustic states.* 91st Meeting of the Acoustical Society of America, 1976

BAUM66: Baum, L.E., Petrie, T, *Statistical inference for probabilistic functions of finite state markov chains.* Annals of Mathematical Statistics, vol 37, pp. 1554-1563, 1966

BAUM70: Baum, L. E., Petrie, T., Soules, G. et al., *A maximization technique in the statistical analysis of probabilistic functions of Markov chains.* Annals of Mathematical Statistics, vol. 41, pp. 164-171, 1970

BAUM72: Baum, L. E., Eagon, J. A., *An inequality with applications to statistical estimation for probabilistic functions of Markov processes.* Inequalities, vol. 3, pp. 1-8, 1972

BELL52: Bellman, R., *On the theory of dynamic programming.* Proc. of the National Academy f Sciences, vol. 38, pp. 716-719, 1952

BER87: Berthod, M., *Un nouvel algorithme d'approximation polygonale.* Technical Repport, INRIA, 1987

BIM93: Bimbot, F., Mathan, L., *Text-free speaker recognition using an arithmetic-harmonic sphericity measure.* EUROSPEECH, pp. 169-172, 1993

BOIT93: Boite, J-M., Bourlard, H., D'hoore, B., Haesen, M., *A New approach toward keyword spotting.* EUROSPEECH, pp. 1273-1276, 1993

BOSS88: Bossemeyer, R. W., Wilpon, J. G., Lee, C. H., Rabiner, L. R., *Automatic speech recognition of small vocabularies within the context of unconstraint input.* J. Acoust. Soc. Amer., suppl. 1., vol 84,November, 1988

BOUR85: Bourlard, H., Kamp, Y., Wellekens, C.J., *Speaker dependent connected speech recognition via phonemic Markov models.* ICASSP, pp. 254-257, Tampa, 1985

BOU85B: Bourlard, H., Kamp, Y., Ney, H., Wellekens, C. J., *Speaker-Dependent Connected Speech Recognition via Dynamic Programming and Statistical Methods.* Speech and Speaker Recognition, Karger, 1985

BOU86: Bourlard, H., Wellekens, C. J., *Connected Speech recognition by phonemic Semi-Markov Chains for State Occupancy Modelling.* Signal Processing III : Theories and Appli., Elsevier Sc. Publ., EUSIPCO pp. 511-514, 1986

BOU88: Bourlard, H., Wellekens, C., J., *Links between Markov Models and Multilayer Perceptrons.* IEEE conference on Neural Information Processing Systems, Denver, 1988

BOUR94: Bourlard, H., D'hoore, B. , Boite, J-M., *Optimizing recognition and rejection performance in wordspotting systems.* ICASSP, pp. I 373-I 376, 1994

BOUR94: Bourlard, H., Morgan, N., *Connectionist speech recognition, a hybrid approach.* Kluwer Academic Publishers, 1994.

BOU95: Bourlard, H., Konig, Y., Morgan, N., *REMAP: Recursive Estimation and Maximization of A posteriori Probabilities. Application to Transition-Based Connectionist Speech Recognition.* Internal Repport of ICSI, TR-94-064, August, 1995.

BROW96: Brown, M. G., Foote, J. T., Jones, G. J., Sparck Jones, K., Young, S. J., *Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval.* ACM Multimedia, pp. 307-316, 1996

CAN83: Canny, J. F., *Finding Edges and Lines in Images.* MIT, TR720, June, 1983

CAR95: Carey, M. J., Parris, E. S., *Topic spotting with task independent models.* Eurospeech, pp. 2133-2136, 1995

CHAN96: Chang E. I., Lippmann, R. P., *Improving wordspotting performance with artificially generated data.* ICASSP, pp. 526-529, 1996

CHER95: Cherfaoui, Mourad, *Indexation et consultation de documents vidéo.* Thèse de doctorat de l' Université de Rennes I, 1995

CHRIS77: Christiansen, R., Rushforth, C., *Detecting and locating key words in continuous speech using linear predictive coding.* ICASSP, vol 25, pp. 361-367, 1977

CHRIS77: Christiansen, R. W., Rushforth, C.K., *Detecting and locating key words in continuous speech using linear predictive coding.* Trans. on speech and signal proc, vol assp-25, N5, pp. 361-367, 1977

CLAR92: Clary, G. J., Hanse, J. H. L., *A novel speech rocognizer for keyword spotting.* ICSLP, pp. 13-16, 1992

DEL93: Deller, J. R., Proakis, J. G., Hansen, J. H., *Discrete-Time Processing Of Speech Signals.* Maxwell Macmillan International, 1993

ELM95: El Meliani, R., O'Shaughnessy, D., *Lexical fillers for task-independent-training based keyword spotting and detection of new words.* EUROSPEECH, pp. 2129-2133, 1995

FENG92: Feng, M-W., Mazo, B., *Continuous word spotting for applications in telecommunications.* ICSPL, 21-24, 1992

FIES97: Fiesler, E., Beale, R., *Handbook of Neural Computation.* IOP Publishing Ltd & Oxford university Press, New-York, 1997

FOOT95: Foote, J. T., Jones, G. J. F., Sparck Jones, K., Young, S. J., *Talker-independent keyword spotting for information retrieval.* Eurospeech, pp. 2145-2148, 1995

FUR89: Furui, S., *Digital Speech Processing, Synthesis, and Recognition.* Marcel Dekker INC, ISBN 0-8247-7965-7, 1989

GILL93: Gillick, L., Baker, J., Bridle, J., Hunt, M., Ito, Y., *Application of large vocabulary continuous speech recognition to topic and speaker identification using telephone speech.* ICASSP, pp. II 471-II 474, 1993

GISH90: Gish, H., Chow, Y-L, Rohlicek, J. R. *Probabilistic vector mapping of noisy speech parameters for HMM word spotting.* ICASSP, pp. 117-120, 1990

GISH92: Gish, H., Ng, K., Rohlicek, J. R., *Secondary processing using speech segments for an hmm word spotting system.* ICSPL, pp. 17-20, 1992

GISH93: Gish, H., Ng, K., 1993, *A segmental speech model with applications to word spotting.* ICASSP, pp. II447-II450, 1993

GOR90: Gorin, A. L., Levinson, S. E., Miller, L. G., Gertner, A. N., Ljolje, A.g, Goldman, E. R., *On adaptive aquisition of language.* ICASSP, pp. 601-604, 1990

HER85: Hermansky, H., Hanson, B. A., Wakita, H., *Low-dimensional representation of vowels based on all-pole modeling in the psychophysical domain.* Speech Comm., Vol. 4, pp. 181-187, 1985

HER90: Hermansky, H., *Perceptual Linear Predictive (PLP) Analysis of Speech.* Journal of the Acoust. Soc. Am., vol87, no. 4, 1990

HERTZ92: Hertz, J., Krogh, A., Palmer, R. G., *Introduction to the theory of neural computation,* Addisson-Wesley Publishing Company, 1992.

HIGG85: Higgins, A. L., Wohlford, R. E., *Keyword recognition using template concatenation.* ICASSP, pp. 1233-1236, 1985

HIL82: Hildreth, E. C. *The Measurement of Visual Motion.* The MIT press, 1982

HOFS92: Hofstetter, E. M., Rose, R. C., *Techniques for task independent word spotting in continuous speech messages.* ICASSP, pp. II 101-II 104, 1992

HUE95: Huet, C., *Choix et développement d'une méthode d'identification du locuteur adaptée à l'indexation de documents video.* Rapport de stage D.E.A. T.T.I. Université de Nice, Sophia-Antipolis, 1995

IMAM93: Imamura, A., Kitai, M., *An application of word spotting in a voice activeted service entry system.* EUROSPEECH, pp. 1061-1064, 1993

ITAK75: Itakura, F., *Line spectrum representation of linear predictor coefficients of speech signal.* Trans. Committee on Speech Research, Acoust. Soc. Jap., S75-34, 1975

JAME94: James, D.A., Young, S.J., *A Fast lattice-based approch to vocabulary independent word spotting.* ICASSP, pp. I 377-I 380, 1994

JEAN93: Jeanrenaud, P., Ng, K., Siu, M., Rohlicek, J.R., Gish, H., *Phonetic-based word spotter: various configurations and application to event spotting.* EUROSPEECH, pp. 1057-1060, 1993

JEL76: Jelinek, F., *Continuous Recognition by statistical methods.* Proc of IEEE, vol. 64, no. 4, pp. 532-555, 1976

JON95: Jones, G. J. F., Foote, J. T., Sparck Jones, K., Young, S. J., *Video mail retrieval: The effect of word spotting accuracy on precision.* ICASSP, pp. 309-312, 1995

JON96: Jones, G. J. F., Foote, J. T., Sparck Jones, K., Young, S. J., *Robust talker-independent audio document retrieval.* ICASSP, pp. I 311- I 314, 1996

JUN89: Junqua, J-C., *Contribution à l'amélioration de la robustesse des systèmes de reconnaissance automatique de mots isolés.* Rapport de thèse,Université de Nancy I, 1989

KAM93: Kamel, M. S., Shen, H. C., Wong, A.-K., Campeamu, R. I., *System for the recognition of human faces.* IBM Systems Journal, Vol. 32/, No. 2, 1993

KAPA93: Kapadia, S., Valtchev, V., Young, S. J., *MMI training for continuous phoneme recognition on the TIMIT database.* Proc. ICASSP, vol. II, pp. 491-494, 1993

KAS91: Kasturi, R., Jain, R., *Computer Vision: Principles.* IEEE Computer Society Press, Washington, pp. 469-480, 1991

KAUF76: Kaufman, G. J., Breeding, K. J., *The Automatic Recognition of Human Faces from Profile Silouhettes.* IEEE Trans. Syst., Man and Cybern., Vol SMC-6, pp. 113-121, 1976

KAY72: Kaya, Y., Kobayashi, K., *A Basic Study of Pattern Recognition.* S. Watanabe Ed., pp265, 1972

KIMU94: Kimura, T., Kuwano, H., Ishida, A., *Compact-size speaker independent speech recognizer for large vocabulary using "compacts" method.* ICSLP, pp. 1379-1382, 1994

KIYA93: Kiyama, J., Itoh, Y., Oka, R., *Spontaneous speech recognition by sentence spotting.* EUROSPEECH, pp. 1053-1056, 1993

KOMO92: Komori, Y., Rainton, D., *Minimum error classification training for HMM based keyword spotting.* ICSLP, pp. 9-12, 1992

KON94: Konig, Y., Morgan, N., *Modeling dynamics in connectionist speech recognition - The time index model.* ICSLP, pp. 1523-1526, 1994

KOO94: Koo, M., Park, S., Kong, K., Doh, S., *KT-stock : A speaker-independent large vocabulary speech recognition system over the telephone.* ICSLP, pp. 1387-1390, 1994

KOSA94: Kosaka, T., Matsunaga, S., Sagayama, S., *Tree-structured speaker clustering for speaker-independent continuous speech recognition.* ICSLP, pp. 1375-1378, 1994

LEE89: Lee, K.-F., *Automatic Speech Recognition.* Kluwer Academic publishers, 1989

LEC91: Leclerc, J., *Traitement du signal de parole.* Université de Liège, 1991

LEV83: Levinson, S. E., Rabiner, L. R., Sondhi, M. M., *An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition.* Bell System Technical Journal, vol. 62, pp. 1035-1074, Avril, 1983

LIPP87: Lippmann, R. P., Gold B, *An introduction to computing with neural nets.* IEEE Magazine on ASSP, vol.4, no. 2, pp. 4-22, 1987

LIPP88: Lippmann, R.P., Martin, E.A., Paul, D. P., *Multi-style training for robust isolated-word speech recognition.* Proc. of ICASSP, vol.2, pp. 705-708, 1988

LIPP94: Lippmann, R. P., Chang, E. I., Jankiwski, C. R., *Wordspotter training using figure of merit back propagation.* ICASSP, pp. I 389-I 392, 1994

LLEI93: Lleida, E., Marino, J.B., Salaverda, J., Bonafonte, A., Martinez, A., *Out of vocabulary word spotting modelling and rejection for keyword spotting.* EUROSPEECH, pp. 1265-1268, 1993

MAC88: McClelland, J. l., Rumelhart, D. E., *Exploration in parallel distributed Processing.* MIT press, 1988

MAD96: Madrane, N., *Indexation et représentation des documents vidéo.* Thèse de doctorat à l'Ecole Nationale Supérieure des Télécommunications, 1996

MAK75: Makhoul, J., *Linear Prediction: A Tutorial Review.* Proc of IEEE, vol. 63, No. 4, pp. 561-580, 1975

MAR92: Marcus, J. N., *A novel algorithm for HMM word spotting, performance, evaluation and error analysis.* ICASSP, pp. II 89-II 92, 1992

MAR81: Mariani, J., *Reconnaissance de la parole continue par diphonèmes ; Processus d'encodage et de décodage phonétique.* GALF, Toulouse, pp. 97-115, 1981

MASA92: Masai, Y., Tanaka, S., Nitta, T., *Speaker-independent keyword recognition based on SMQ/HMM.* ICSPL, pp. 619-622 , 1992

MASA94: Masai, Y., Iwasaki, J., Tanaka, S., Nitta, T., Yao, M., Onogi, T., Nakayama, A., *A keyword-spotting unit for speaker-independent spontaneous speech recognition.* ICSLP, pp. 1383-1386, 1994

MERI88: Mérialdo, B., *Phonetic Recognition Using Hidden Markov Models and Maximum Mutual Information Training.* Proc. ICASSP, vol. 1, pp. 111-114, 1988

MORG91: Morgan, D. D., Scofield, C. L., Adcock, J. E., *Multiple neural network topologies applied to keyword spotting.* ICASSP, pp. 313-316, 1991

NAG91: Nagasaka, A., Tanaka, Y., *Automatic Video Indexing and Full-Video Search For Objects Appearances.* Proc. IFIP WG 2.6 @nd Working Conference on Visual Databases Systems, pp. 119-133, 1991

NAKA93: Nakamura, S., Akabane, T., Hamaguchi, S., *Robust word spotting in adverse car environments.* EUROSPEECH pp. 1045-1048, 1993

NEY84: Ney, H., *The use of a one-stage dynamic programming algorithm for connected word recognition.* IEEE Trans. on Acoustices, Speech and Signal Processing, ASSP-32, pp. 263-271, 1984

NEY94: Ney, H., Aubert, X., *A word graph algorithm for large vocabulary, continuous speech recognition.* ICSLP, pp. 1355-1358, 1994

NORM91: Normandin, Y., *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem.* Ph. D. Thesis, Mc Gill University Montreal, 1991

NORM94: Normandin, Y., *Optimal Splitting og HMM Gaussian Mixture Components with MMIE Training.* Proc. ICASSP, vol. 1, pp. 449-452, 1994

NORM94: Normandin, Y., Lacouture, R., Cardin, R., *MMIE Training for large vocabulary continuous speech recognition.* ICSLP, pp. 1367-1370, 1994

OKAN93: O'Kane, M. J., Kenne, P. E., *Word and phrase spotting with limited training.* EUROSPEECH, pp. 1269-1272, 1993

OKAW93: Okawa, S., Kobayashi, T., Shirai, K., *Word spotting in conversational speech based on phonemic unit likelihood by mutual information criterion.* EUROSPEECH, pp. 1281-1284, 1993

PESK93: Perskin, B., Gillick, L., Ito, Y., Lowe, S., Roth et al, *Topic ans Speaker Identification via Large Vocabulary Continuous Speech Recognition.* ARPA Workshop on Human Language Technology, Princeton, March, 1993

PHIL94: Phillips, M., Goddeau, D., *Fast match for segment-based large vocabulary continuous speech recognition.* ICSLP, pp. 1359-1362, 1994

RAB78: Rabiner, L.R., Schafer, R. W., *Digital Processing of Speech Signals.* Prentice-Hall inc. ISBN : 0-13-213603-1, 1978

RAB85: Rabiner, L. R., Juang, B.H., Levinson, S. E., Sondhi, M. M., *Recognition of isolated digits using hidden Markov Models with continuous mixture densities.* AT&T Technical Journal, vol 64, no.6, pp. 1211-1233, 1985

ROH93: Rohlicek, J. R., Jeanrenaud, P., Ng, K., Gish, H., Musicus, B., Siu, M., *Phonetic training and language modeling for word spotting.* ICASSP, pp. II 459-II 462, 1993

ROS83: Rosenberg A. E., Rabiner, L. R., Wilpom, J. G., Kahn, D., *Using concatenated demi-syllabes in an isolated word recognition system.* 11ème I.C.A., Toulouse, 1983

ROSE90: Rose, R. C., Paul, D. B, *A Hidden Markov Model based keyword recognition system.* ICASSP, pp. 129-132, 1990

ROSE91: Rose, R.C., Chang, E. I., Lippmann, R. P., *Techniques for information retrieval from voice messages.* ICASSP, pp. 317-320, 1991

ROSE92: Rose, R. C., *Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech.* ICASSP pp. II 105-II 108, 1992

ROSE93a: Rose. R. C., Hofstetter, E. M., *Task independent wordspotting using decision tree based allophone clustering.* ICASSP, pp. II 467-II 450, 1993

ROSE93b: Rose, R. C., *Definition of subword acoustic units for wordspotting.* EUROSPEECH, pp. 1049-1052, 1993

RUM86: Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing, Explorations in the microstructure of Cognition. Volume 1: Foundations.* MIT press, 1986

RUS81: Ruske, G., Schotola, T., *The efficiency of demi-syllable segmentation in the recognition of spoken words.* Proc. ICASSP, pp. 971-974, 1981

SCHW91: Schwartz, R., Austin, S. *A comparison of Several Approximate Algorithms For Finding Multiple (N-BEST) Sentece Hypotheses.* ICASSP, pp. 701- 704. 1991

SONG93: Song, J., *Continuous HMM for word spotting and rejection of non vocabulary word in speech recognition over telephone networks.* EUROSPEECH, pp. 1563-1566, 1993

SOON91: Soong, F. K., Huang, E.-F. *A Tree-Treillis Based Fast Search for Finding the N Best Sentece Hypotheses in Continuous Speech Recognition.* ICASSP, pp. 705-708, 1991

STEIN91: Steinboss, V., *A Search Organization for Large-Vocabulary Recognition Based on N-Best Decoding.* Eurospeech 1991, Vol. 3, pp. 1217-1220, Italy, 1991

STO84: Stonham, T. J., *Practical Face Recognition and Verification with WISARD.* Aspects of Face Processing Dadrecht Ed., pp 426-441, 1984

SUKK93: Sukkar, R. A., Wilpon, Jay, *A two pass classifier for utterance rejection in keyword spotting.* ICASSP, pp. II 451-II 454, 1993

TER88: Terzopoulos, D., Watkin, A., Kass, M., *Constraints on Deformables Models: #-D Shape on Non Rigid Motion.* Artificial Intelligence., Vol. 36., pp.91-123, 1988

TUB89: Tubach, J. P. & al., *La parole et son traitement automatique.* Masson et CNET-ENST, 1989

TUR91: Turk, A., Pentland, A. P., *Face Recognition Using Eigenfaces.* Proc. Int. Conf. on Pattern Recognition, pp. 586-591, 1991

VALT94: Valtchev, V., Odell, J.J., Woodland, P.C., Young, S.J., *A dynamic network decoder design for large vocabulary speech recognition.* ICSLP, pp. 1351-1354, 1994

VILL93: Villarrubia, L., Acero, A., *Rejection techniques for digit recognition in telecommunication applications.* ICASSP, pp. II 455-II 458, 1993

VIN68: Vintsyuk, T. K., *Speech discrimination by dynamic programming.* Kibernetika 4:81-88, 1968

WAKK77: Wakita, H., *Normalization of vowels by vocal tract length and its application to vowel identification.* Trans. on speech and signal proc, vol assp-25, N2, pp. 183-192, 1977

WELL87: Wellekens, C. J., *Explicit Time Correlation in Hidden Markov Models for Speech Recognition.* ICASSP, vol. 1. pp. 384-386, Dallas, 1987

WEIN93: Weintraub, M., *Keyword spotting using sri's decipher large vocabulary speech recognition system.* ICASSP, pp. II 463-II 466, 1993

WIL89: Wilpon, J. G., Lee, C.H., Rabiner, L.R., *Application of hidden markov models for recognition of a limited set of words in unconstrained speech.* ICASSP, pp. 129-132, 1989

WIL90: Wilpon, J. G. , Rabiner, L. R., Lee, C. , Glodman, E.R., *Automatic recognition of keywords in unconstrained speech using hidden markov models.* Trans ASSP vol. 38, no. 11, 1990

WIL91: Wilpon, J.G., Miller, L.G., Modi, P., *Improvements and applications for keyword recognition using hidden markov modeling techniques.* ICASSP, pp. 309-312, 1991

WILC92: Wilcox, L. D., Bush, M. A., *Training and search algorithms for an interactive wordspotting system.* ICASSP, pp. II 97-II 100, 1992

WOOT94: Wooters, C., Stolcke, A., *Multiple-pronunciation lexical modeling in a speaker independent speech understanding system.* ICSLP, pp. 1363-1366, 1994

XU96: Xu, D., Fancourt, C., Wang C., *Multi-channel HMM.* ICASSP, pp. 841-844, 1996

YANG94: Yang, Y.-J., Lin, S.-C., Chien, L.-F., Chen, K.-J., Lee, L.-S., *An intelligent and efficient word-class-based chinese language model for mandarin speech recognition with very large vocabulary.* ICSLP, pp. 1371-1374, 1994

YOUN94: Young, S. R., *Detecting misrecognitions and out of vocabulary words.* ICASSP, pp. II 21-II 24, 1994

ZEPP92: Zeppenfeld, T., Waibel, A. H., *A Hybrid neural network, dynamic programming word spotter.* IICASSP, pp. II 77-II 80, 1992

ZEPP93: Zeppenfeld, T., Houghton, R., Waibel, A., *Improving the MS-TDNN for word spotting.* ICASSP, pp. II 475-II 478, 1993

ZHA93: Zhang, H.J.Kankanhalli, A., Smoliar, S. W., *Automatic Partitioning of Full-Motion Video.* Multimedia Systems, 1(1), 10-28 July, 1993

ZWI80: Zwicker, E., Terhardt, E., *Analytical Expressions for Critical Band Rate and Critical Bandwidth as a Function of Frequency.* J. Acoust. Soc. Am., vol 68, pp. 1523-1525, 1980

ZWI81: Zwicker, E., Feldtkeller, R., *Psychoacoustique: l'oreille récepteur d'information.* Masson, 1981

Index

A

acoustique
    modélisation
        HMM 25, 26
        REMAP 58
    prédiction 66
    séquence v
    vecteur v

B

Baum-Welch
    contribution locale 31
    estimation des paramètres 33
    probabilité
        arrière 30
        avant 29
        émission 31
        transition de 31
    récurrence
        arrière 31
        avant 30
    réestimation avant-arrière 29
    réseau de neurones 35
        entraînement 36

C

catégories 194
chemin vi
    section de vi
    valide vi
classes 194, 200
coarticulation 21
confusion matrice de 138
contribution locale
    Baum-Welch 31
    REMAP 66
convergence
    distance de 51

D

discriminant
    REMAP critère 61
distance de convergence 51

E

entropie 75
erreur
    exactitude au niveau du mot 76
    fausse alarme 76
    gain de temps 79
        moyen 80
    non détection 76
    pourcentage correct 76
    précision 77
        moyenne 78
        standard 78
exactitude au niveau du mot 76

F

fausse alarme 76
Figure Of Merit 77
fonction d'activation 14

G

gain de temps 79
moyen 80
groupe d'hypothèses 138

H

HMM
modélisation
acoustique 25, 26
langage 24, 26
probabilité
de transition 24
émission 25
hypothèse
détection 108
groupe 138
phonétique 106

I

information mutuelle 88
Intégration temporelle 106

L

langage
modélisation
HMM 24, 26
REMAP 59

M

matrice de confusion 138
maximisation
a posteriori 27
information mutuelle 27
vraisemblance 27, 28
maximum d'information mutuelle 27
maximum de vraisemblance
maximisation 28
modèle de Markov 22
modélisation
acoustique
HMM 25, 26
REMAP 58
langage
HMM 24, 26
REMAP 59

N

noeud 136
non détection 76

P

paramètres, estimation des
Baum-Welch 33
perplexité 74, 75
grammaire 75
phonème
hypothèse 106
pourcentage correct 76
précision 77
moyenne 78
standard 78
précision moyenne 95
prédiction acoustique 66
premature backtracking 50
probabilité
a posteriori
globale, REMAP 61
maximisation 27
arrière
Baum-Welch 30
REMAP 64

avant
Baum-Welch 29
REMAP 64
de transition
HMM 24
émission
Baum-Welch 31
HMM 25
transition de
Baum-Welch 31
conditionnelle, REMAP 57, 66
visite de
REMAP 68
vraisemblance 26
programmation dynamique 21

R

R.O.C. 77
récurrence
arrière
Baum-Welch 31
REMAP 66
avant
Baum-Welch 30
REMAP 65
Viterbi 39
réestimation avant-arrière
Baum-Welch 29
REMAP
contribution locale 66
critère discriminant 61
entraînement 70
modélisation
acoustique 58
langage 59
objectifs 69
probabilité
a posteriori globale 61
arrière 64
avant 64
transition conditionnelle de . 57, 66
visite de 68
récurrence
arrière 66
avant 65
réseau de neurones 57
réseau de neurones
Baum-Welch 35
entraînement
Baum-Welch 36
REMAP 70
fonction d'activation 14
REMAP 57
Viterbi 41

S

section de chemin vi
séquence acoustique v

T

treillis
hypothèse 106

V

Valeur de mérite 77
vecteur
acoustique v
apprentisage d' 195, 200
entrée d' 194
sortie de 195, 200, 204
vecteur d'apprentisage 200, 204
vecteurs
modèles 194
Viterbi
algorithme 39
récurrence 39
réseau de neurones 41
retour arrière prématuré 50
vraisemblance 26
maximisation 27

Curriculum Vitae

Nom: Philippe Gelin.

Date de naissance: 25 Mars 1969, en Belgique.

Académique:

1993-1997: Préparation d'une thèse à l'Institut Eurécom, en vue de l'obtention d'un doctorat en "Système de Communication" à l'EPFL. Cette thèse sera défendue le 30 Avril 1997. L'intitulé de la thèse est "Détection de mots clés dans un flux de parole : Application à l'indexation de documents multimédia".

1987-1993: Etude d'ingénieur civil électricien, tendance informatique, option intelligence artificielle. Diplômé avec distinction de l'université de Liège, Belgique.
Le travail de fin d'étude s'intitulait "Utilisation des chaînes de Markov pour la reconnaissance de la parole" et fut effectué chez Lernout & Hauspie Speechproducts, à Bruxelles.

Publications:

Ph. Gelin & Chris. J. Wellekens: *Keyword Spotting for Multimedia Document Indexing*. Soumis à "SPIE's International Symposium on Voice,Video and Data Communications" 1997 devant se tenir à Dallas en Novembre 1997.

Ph. Gelin & Chris. J. Wellekens: *REMAP for Video Soundtrack Indexing.* Accepté à "International Conference on Acoustics, Speech, and Signal Processing" 1997 devant se tenir à Munich en Avril 1997.

Ph. Gelin & Chris. J. Wellekens: *Keyword Spotting Enhancement for Video Soundtrack Indexing.* "International Conference on Spoken Language Processing", Octobre 1996, Philadelphie,

Ph. Gelin & Chris. J. Wellekens: *Keyword Spotting for Video Soundtrack Indexing*. "International Conference on Acoustics, Speech, and Signal Processing", Mai 1996, Atlanta. SP8. Vol. 1 pp. 299-302.