

Vers une analyse acoustico-phonétique de la parole indépendante de la langue, basée sur ALISP¹

Jan Černocký (1), Geneviève Baudoin (2), Dijana Petrovska-Delacrétaz (3) et Gérard Chollet (3)

¹ Institut de Radioélectronique, FEI VUT, Brno, République Tchèque

² Département Signaux et Télécommunications, ESIEE, Paris, France

³ CNRS-LTCl, ENST, Paris, France

Mots clés :

Décodage acoustico-phonétique, codage à très bas débit, apprentissage non supervisé, reconnaissance de la parole, matrice de confusion

¹ Ce travail a été partiellement financé par le Ministère de l'Éducation de la République Tchèque, sous le projet N° VS97060, et par le Ministère français de la Recherche, sous le projet RNRT-SYMPATEX.

Résumé : De nombreux systèmes de synthèse et de reconnaissance automatique de la parole utilisent des unités de parole liées aux phones. Les phones sont les réalisations physiques des phonèmes correspondants et sont en général définis a priori et dépendants de la langue considérée. Nous présentons une alternative à cette approche : une détermination des unités de parole à l'aide des techniques ALISP « *Automatic Language Independent Speech Processing* » - Traitement Automatique de la Parole, Indépendant de la Langue). ALISP permet de choisir l'inventaire des unités de parole considérées à partir d'une analyse statistique de corpus de parole, sans a priori sur nos connaissances phonétiques et/ou phonologiques. Nous avons testé expérimentalement de telles unités dans un vocodeur à très bas débit : le débit moyen ainsi obtenu pour le codage des unités est de 120 bps. Nous présentons également les résultats de la comparaison d'une segmentation ALISP avec une segmentation acoustico-phonétique dans deux cas : mono et multi-locuteur.

Abstract: Numerous systems for speech synthesis and automatic speech recognition are based on speech units related to phones. The phones are the physical realisations of the corresponding phonemes. Such units are therefore defined a priori, and are language dependent. In this paper, we present an alternative approach : determination of speech units using ALISP (Automatic Language Independent Speech Processing) techniques. ALISP allows choosing the inventory of units from a statistical analysis of the speech corpus, requiring neither phonetic nor orthographic transcriptions of the speech data. Experimentally, such units have been tested in a very low bit-rate speech coder : the resulting average rate is 120 bps. These automatically derived speech units are also compared with an acoustic-phonetic segmentation on a speaker dependent and speaker independent basis.

1. Introduction

Les systèmes modernes de traitement automatique de la parole s'appuient sur des unités de type *sous-mot*. Dans les systèmes de reconnaissance vocale (à vocabulaire illimité), les unités constituent le niveau intermédiaire entre la description acoustique (ou paramétrique) et le niveau lexical. Dans la reconnaissance du locuteur, une pré-segmentation avec ces unités suivie de l'utilisation de plusieurs systèmes de décision permet d'obtenir de meilleurs résultats que les systèmes utilisant une modélisation « globale ». Finalement dans le codage à très bas débit, la transmission de l'information sur chaque trame acoustique n'étant plus possible, ces unités déterminent l'information symbolique transmise dans le canal ou stockée.

Dans tous les domaines cités, ces unités de base doivent être modélisées par des modèles mathématiques. Le schéma général d'estimation de ces modèles (applicable aussi bien pour la reconnaissance, la synthèse que pour le codage) est donné sur la Figure 1. En synthèse, par exemple, l'entrée du modèle est un texte. Celui-ci est converti en signal de parole qui est ensuite comparé avec ce même texte lu par un locuteur humain. Le modèle (classiquement des diphones) est ajusté de façon à minimiser la différence entre la parole synthétique et humaine. Le critère est ainsi clairement défini :

1. dans la **synthèse vocale**, la parole synthétique doit différer le moins possible du signal produit par un locuteur humain ;
2. dans la **reconnaissance de parole**, la compréhension du signal par la machine devrait s'approcher de la compréhension humaine (nous ne parlons pas seulement du texte, la dictée n'étant pas la seule application de la reconnaissance de la parole) ;
3. dans le **codage**, le signal après la chaîne codage-décodage doit différer le moins possible du signal original ;
4. dans la **vérification du locuteur**, le but est de mieux séparer les clients d'un système des non-clients (imposteurs).

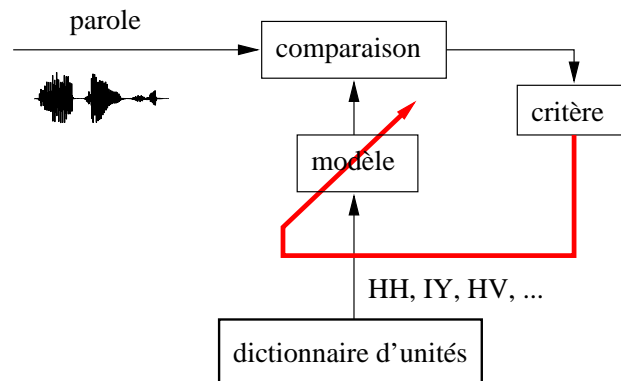


Figure 1: Estimation des modèles pour le traitement de la parole.

Classiquement, ces unités dérivent d'unités linguistiques telles que : les **phonèmes**, leurs dérivés (diphones, phonèmes en contexte...), syllabes, ou autres. Cependant, l'usage de ce type d'unités ne s'impose que dans le premier domaine et encore seulement à l'entrée d'un système de synthèse. Dans les autres domaines (2,3,4), ce choix se justifie historiquement, car dans les langues indo-européennes, les recherches en phonétique et acoustique de la parole sont classiquement très liées à l'orthographe (on pourrait se demander, quel serait l'état de ces sciences, si par exemple l'anglais était écrit en idéogrammes).

De plus, pour une application en traitement automatique de la parole, la définition d'un jeu d'unités et la détermination de leurs positions dans le signal de parole (alignement) nécessitent une très bonne expertise en phonétique et en linguistique. La transcription et/ou l'annotation manuelle de bases de données (BD ou corpus) sont des tâches très lourdes ; il est connu, que ces étapes sont les plus coûteuses et les plus sujettes aux erreurs humaines dans le procédé de création de corpus.

Aussi avons nous utilisé une autre approche s'appuyant sur la détermination *automatique* des jeux d'unités et la transcription *automatique* des corpus. Les techniques regroupées sous un nom générique

ALISP « Automatic Language Independent Speech Processing » - Traitement Automatique de la Parole, Indépendant de la Langue) se basent sur les *données* et tentent de limiter au minimum les connaissances nécessaires a priori. Recherchant un équilibre entre la précision de la description et son économie, ces techniques détectent des régularités dans le signal (ou sa paramétrisation) pour en faire émerger sa structure.

L'idée qu'il est possible d'apprendre des unités de base pour le traitement automatique de la parole, en exploitant uniquement le signal, est d'abord fondée sur des expériences « naïves » (les bébés apprennent à parler sans savoir lire, en écoutant la voix de leurs proches, la même remarque s'applique aux adultes analphabètes). De nombreux chercheurs se sont aussi penchés sur le problème de la recherche et de l'apprentissage des unités : Kohonen [12, 13], a développé une théorie des mémoires associatives, capables d'apprentissage non supervisé et de généralisation sur les données ; Kruskal et Sankoff [14], ont publié de nombreux travaux sur l'apprentissage des séquences d'ADN et Atal [2], a défini un hyper-espace de segments acoustiques, où il tente de mesurer l'information portée par le langage parlé. Enfin, les travaux de Marcken [15], portent sur l'acquisition non-supervisée du lexique de la parole continue. Ses algorithmes sont basés sur l'encodage optimal des séquences de symboles au sens d'une longueur de description minimale « Minimum Description Length » et utilisent une représentation hiérarchique du langage.

L'utilisation de ces unités est directe dans les domaines où la représentation symbolique ne constitue qu'un niveau intermédiaire entre deux signaux (le codage), ou auxiliaire (pré-segmentation dans la vérification du locuteur). Dans le cas où une transcription linguistique (reconnaissance vocale) serait nécessaire, des techniques de mise en correspondance des unités ALISP et des phonèmes doivent être élaborées. Mieux encore, il est possible de remplacer des dictionnaires de prononciation classiques par leurs homologues constitués à partir d'unités déterminées automatiquement.

Cet article est structuré de la façon suivante : dans la section 2, nous présentons les outils ALISP qui servent à déterminer automatiquement des unités dans un corpus de signaux de parole. La section 3 est consacrée aux expériences en codage de la parole à très bas débit (la vérification la plus simple de notre approche). La section 4 présente des comparaisons d'une segmentation ALISP avec une segmentation acoustico-phonétique fine (dans les cas mono-locuteur et multi-locuteur) et une segmentation en classes phonétiques larges, dans les cas mono-locuteur (les deux étant obtenues par un système de reconnaissance). La section 5 conclut notre article.

2. Outils ALISP - justifications théoriques et solutions techniques

Sur un corpus de parole donné, la détermination des unités s'effectue en deux étapes principales : dans la première, nous définissons le jeu d'unités et nous recherchons une segmentation initiale du corpus. Dans la deuxième étape, ces unités sont modélisées par des modèles stochastiques. Après la phase d'apprentissage, le système peut traiter un signal de parole inconnu.

Nous appelons les techniques utilisées pour cette extraction et cette modélisation des « outils » (voir la chaîne de traitement de la Figure 2). Certains parmi eux sont utilisés largement en traitement de la parole (paramétrisation, modèles de Markov cachés), les autres (décomposition temporelle, multigrammes) sont plus spécifiques aux approches ALISP. Ces outils sont hautement modulaires, et la position de certains d'entre eux dans la chaîne de traitement peut changer (c'est le cas pour les multigrammes). Les sous-sections suivantes donnent une description plus détaillée de ces outils.

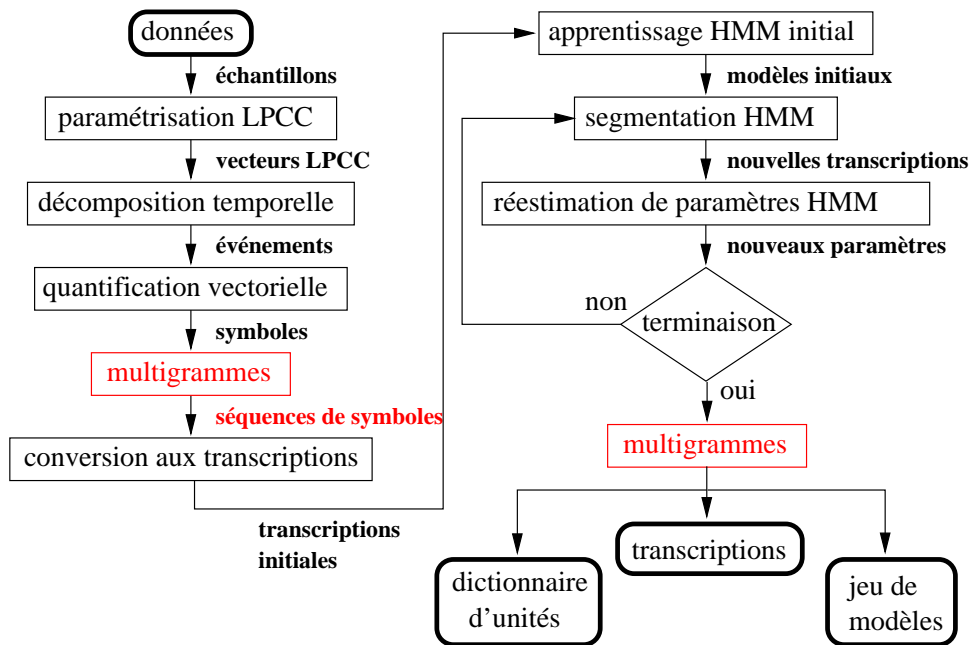


Figure 2: Outils utilisés dans la recherche automatique des unités pour le traitement de la parole.

2.1 Paramétrisation

L'une des premières étapes dans tout traitement automatique de la parole est sa **paramétrisation**. À partir d'un signal numérisé, nous devons extraire un nombre limité de paramètres décrivant le signal, et qui conviennent au traitement automatique de la parole. Ici, nous nous limitons aux paramètres d'un filtre numérique représentant le conduit vocal selon la Figure 3. Ces paramètres, dits de *prédiction linéaire LPC* [17], sont convertis en paramètres LPC-cepstraux [17], moins corrélés que les coefficients de la prédiction linéaire. Ces paramètres ne modélisent que l'enveloppe spectrale du signal parole (voir la Figure 4). Ceci peut engendrer des difficultés lors de l'application de nos méthodes sur les langues tonales. Il est cependant connu, que la fréquence fondamentale influence aussi cette enveloppe spectrale.

Les paramètres sont classiquement extraits sur des trames de longueur fixe (Figure 5). Cette limitation par rapport à des approches comme les ondelettes par exemple, est compensée par une notion de variabilité temporelle dans les autres outils décrits plus loin.

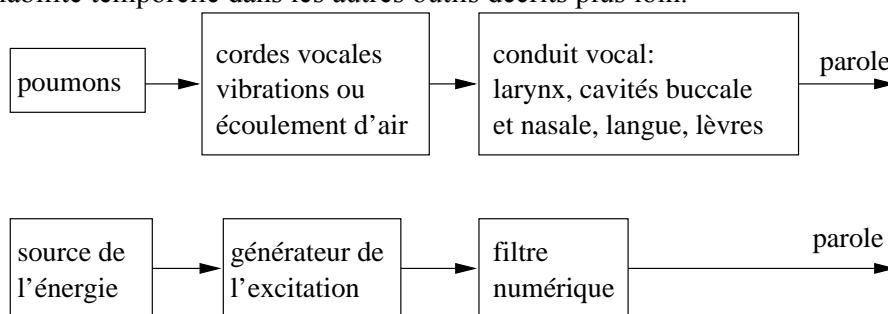


Figure 3: Modèle de production de la parole utilisé pour l'extraction des paramètres.

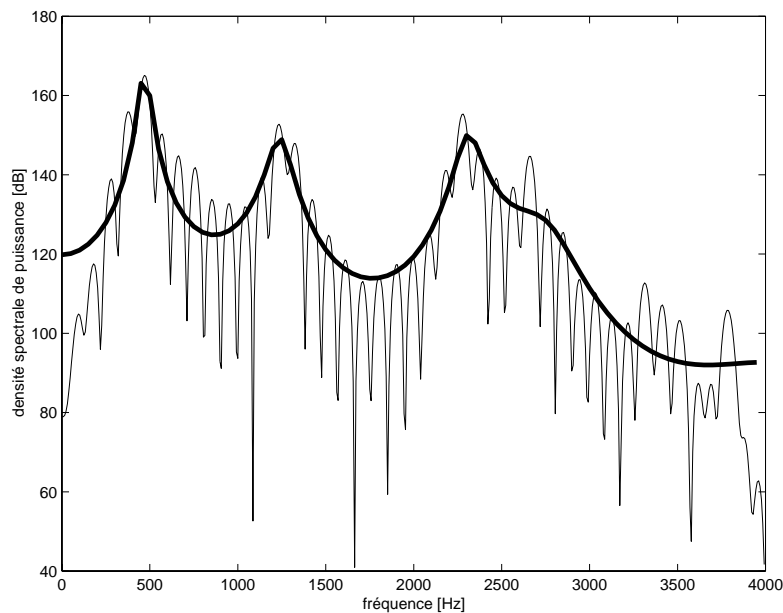


Figure 4: Spectre d'un segment voisé de parole avec son enveloppe.

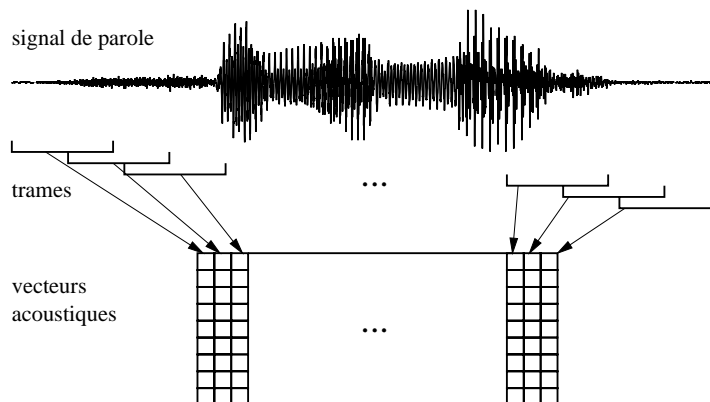


Figure 5: Signal de parole, découpage en « trames » et détermination d'un nombre limité de paramètres (constituant un vecteur acoustique) caractérisant chaque trame.

2.2 Décomposition temporelle

On applique la *décomposition temporelle* sur les vecteurs de coefficients LPC-cepstraux. Nous avons choisi cette méthode parmi les nombreuses techniques de segmentation de la parole (voir [5] pour un résumé), car non seulement elle utilise un critère de stabilité spectrale pour déterminer un segment, mais elle prend aussi en compte leurs transitions - celles-ci sont représentées par un recouvrement des fonctions d'interpolation. La décomposition temporelle, introduite par Atal [3] et utilisée par Bimbot [4], approche une matrice de paramètres par des vecteurs-cibles et des fonctions d'interpolation. Techniquement, la recherche des cibles et des fonctions d'interpolation de la décomposition temporelle se fait par une *décomposition en valeurs singulières* à court terme d'une sous-matrice \mathbf{Y} de la matrice des coefficients cepstraux \mathbf{X} : $\mathbf{Y}^T = \mathbf{U}^T \mathbf{D} \mathbf{V}$.

On assemble ensuite les lignes de la matrice \mathbf{U} pour trouver une fonction d'interpolation concentrée sur une fenêtre rectangulaire. La ré-estimation de la fonction d'interpolation et l'adaptation de la fenêtre sont itérées pour obtenir une compacité maximale de la fonction d'interpolation. Le post-traitement des fonctions d'interpolation contient un lissage, une dé-corrélation et une normalisation. Dans l'étape suivante, le calcul des cibles est effectué en utilisant la pseudo-inverse de la matrice des fonctions d'interpolation. Enfin, les cibles et les fonctions d'interpolation sont affinées localement.

Les fonctions d'interpolation, déterminant ainsi des parties quasi-stationnaires du signal, définissent une première *segmentation* de la parole. La Figure 6 montre un exemple de la décomposition temporelle.

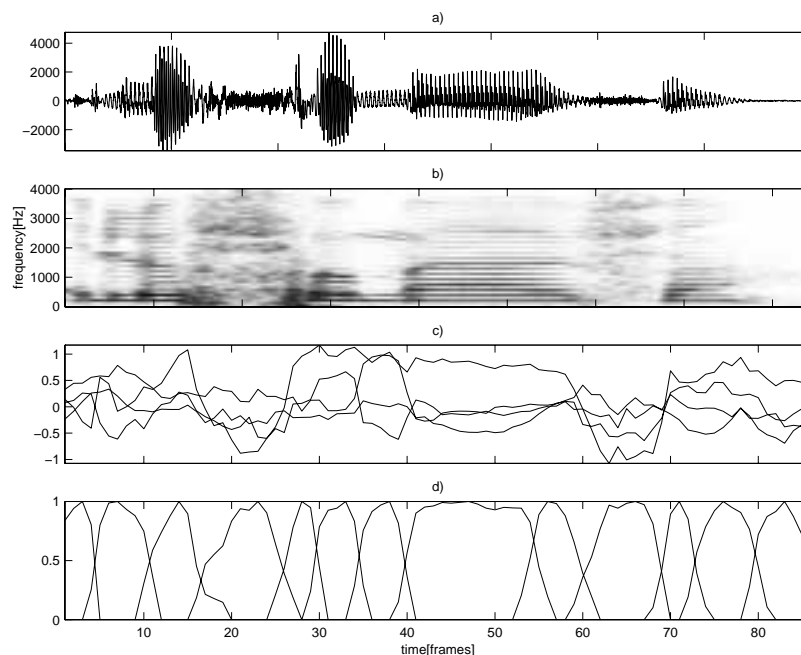


Figure 6: Exemple d'une décomposition temporelle – « le chômage » : a) le signal ; b) le spectrogramme ; c) les trajectoires des quatre premiers coefficients LPC-cepstraux et d) les fonctions d'interpolation de la décomposition temporelle.

2.3 Quantification vectorielle

Les segments trouvés par la décomposition temporelle subissent une classification non-supervisée, réalisée ici par une *quantification vectorielle*. Il existe plusieurs méthodes [10], de recherche de classes en fonction de la proximité des vecteurs de paramètres dans un espace à P dimensions : à côté de la quantification vectorielle, ce sont par exemple les *Modèles de Markov cachés ergodiques*, dans lesquels on attribue les vecteurs aux états en fonction d'une vraisemblance, ou les « *Self-Organizing Maps* » de Kohonen, où la proximité dans un espace à grande dimension se traduit par une proximité des classes dans une espace de petite dimension (typiquement 2 - une surface). Le but de toutes ces méthodes est de réunir dans une même classe les vecteurs qui se ressemblent, et de mettre dans des classes différentes les vecteurs distincts. Mathématiquement parlant, il nous faut minimiser les distances intra-classe, tout en maximisant les distances inter-classe.

La quantification vectorielle est une réponse simple à ce problème. Les vecteurs sont représentés par un dictionnaire de vecteurs-codes (nous allons utiliser le terme anglais « codebook » dans la suite, pour ne pas confondre ce dictionnaire avec le dictionnaire des unités ALISP): $\mathbf{Y} = \{\mathbf{y}_i; 1 \leq i \leq L\}$, où L est le nombre de classes. Ce « codebook » doit être *appris* sur une base de données en minimisant une distance moyenne globale entre les vecteurs d'apprentissage et les vecteurs-codes. Cet apprentissage est mis en œuvre par l'algorithme Linde-Buzo-Gray [10] avec des éclatements successifs du « codebook » : $L = 1,2,4,\dots$. L'ensemble d'apprentissage est constitué des vecteurs cepstraux originaux situés aux centres de gravité des fonctions d'interpolation.

Une fois le « codebook » appris, nous pouvons procéder à une *quantification* : dans cette étape, on attribue à chaque événement de la décomposition temporelle le numéro (étiquette) de la classe qui lui est la plus proche. Pour cette quantification, nous avons utilisé tous les vecteurs d'un segment prédéterminé par la décomposition temporelle en utilisant une distance cumulée.

À l'issue de la décomposition temporelle suivie de la quantification vectorielle, on effectue une *transcription initiale* (bornes temporelles et étiquettes) de la base de données de parole.

2.4 Multigrammes

Il se peut que nous ayons besoin d'unités plus longues que celles déterminées par une combinaison « décomposition temporelle - quantification vectorielle ». Bien que nous travaillions avec des unités

déterminées automatiquement, nous pouvons nous approcher ainsi des techniques syllabiques ou diphoniques utilisées dans les traitements classiques. Ce séquençage a de nombreux avantages. Par exemple, lors du codage, l'utilisation d'unités plus longues implique une diminution du débit binaire (le dictionnaire d'unités devient plus grand, mais le nombre d'unités à transmettre par seconde décroît). La diminution du nombre de transitions entre unités permet aussi d'atténuer les effets indésirables dus à la concaténation de segments courts. On appelle « *multigramme* » une séquence formée d'un nombre variable de symboles ; et *n*-multigrammes les multigrammes, dont la longueur est limitée à *n*. La technique utilisée pour ce séquençage est appelée décomposition en multigrammes [6]. Cette méthode, dont nous connaissons plusieurs variantes - discrètes ou continues - permet de détecter des *séquences caractéristiques* d'unités dans le corpus d'apprentissage.

En supposant que les événements de la décomposition temporelle ont déjà été étiquetés par la quantification vectorielle, nous avons une chaîne de symboles, représentée schématiquement dans la Figure 7. Pour un dictionnaire de multigrammes $\{x_i\}$ donné, la segmentation d'une chaîne d'observations discrètes et sa transcription en multigrammes se fait en maximisant la vraisemblance de la segmentation et de l'étiquetage :

$$(S^*, X^*) = \arg \max_{\forall (S, X)} L(O, S, X | \{x_i\}), \quad (1)$$

où *O* est la chaîne d'observations, *S* sa segmentation et *X* l'attribution des multigrammes. Dans le cas des multigrammes discrets, le dictionnaire contient les séquences x_i , ainsi que leurs probabilités π_i . Nous pouvons écrire : $s_j \equiv x_i$; $L(O, X | \{x_i\}) = P(x_{i1}) P(x_{i2}) \dots P(x_{iq})$.

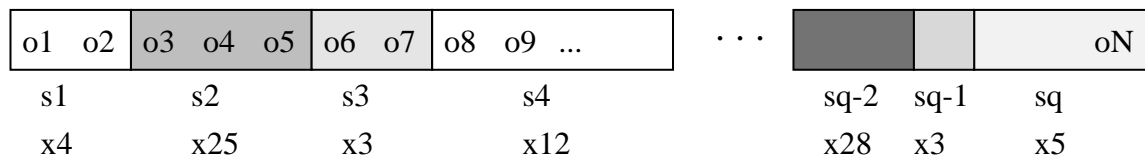


Figure 7: Séquençage des symboles par les multigrammes.

Le dictionnaire de multigrammes n'est pas connu a priori et doit être appris sur une base de données de symboles. Cet apprentissage commence par une *initialisation*. On initialise les valeurs des probabilités π_i de toutes les séquences possibles de longueur 1 à *n* par le nombre d'occurrences de ces séquences dans la base de données d'apprentissage. Après cette initialisation, on itère plusieurs étapes de segmentation au sens du maximum de vraisemblance (Éq. 1). À l'étape *n*, on effectue la segmentation en utilisant le dictionnaire déterminé à l'étape (*n-1*), puis on met à jour les probabilités π_i des multigrammes à partir de la nouvelle segmentation. Durant ces itérations, le dictionnaire est élagué des multigrammes rares en imposant un nombre d'occurrences minimal.

On peut utiliser la méthode des multigrammes à deux niveaux de la chaîne de traitement. Elle peut être appliquée à :

- **des événements de la décomposition temporelle classés par quantification vectorielle** (les multigrammes servent ici à initialiser des HMMs - introduits dans la sous-section suivante - avec un nombre variable d'états) ;
- **des symboles générés par une segmentation par les HMMs** (les multigrammes aident ici à la création d'unités plus longues).

Les multigrammes constituent ainsi un module dont la position peut varier dans le schéma de la Figure 2.

2.5 Modèles de Markov cachés (HMM)

Dans la deuxième étape de traitement, les unités trouvées par la combinaison « décomposition temporelle et quantification vectorielle » ou « décomposition temporelle, quantification vectorielle et multigramme » sont modélisées par les *Modèles de Markov Cachés* « *Hidden Markov Models* » (*HMM*). Cependant ce formalisme, utilisé largement en reconnaissance de parole, ne sert pas seulement à produire des modèles, mais contribue lui-même à un affinement du jeu d'unités par des itérations de segmentation du corpus (un alignement des HMM avec les données) et de ré-estimation des paramètres des modèles.

La théorie des HMM [16, 21] est assez complexe et ne peut pas être traitée ici en détail. La reconnaissance de la parole à l'aide des HMM est basée sur la maximisation de la vraisemblance de l'observation et des modèles :

$$\arg \max_{\{M_1^N\}} L(\mathbf{O} | M_1^N) L(M_1^N),$$

où \mathbf{O} est une chaîne d'observations (vectorielles cette fois-ci) et M_1^N une séquence de modèles. La vraisemblance $L(\mathbf{O} | M_1^N)$ dite « acoustique », quantifie la correspondance entre les données et les modèles. Quant à la vraisemblance $L(M_1^N)$ du « modèle de langage », elle donne une probabilité a priori de la séquence de modèles M_1^N .

Un choix important est celui de l'architecture des HMM. Nous avons choisi l'architecture la plus simple gauche-droite (Figure 8). Le nombre de modèles est déterminé par la taille L du « codebook » de quantification vectorielle ou par la taille Z du dictionnaire des multigrammes. Le nombre d'états émetteurs des HMM est défini comme $(2i+1)$, où i est le nombre d'unités dans un multigramme. Au cas où l'on ne travaille pas avec les multigrammes, ce nombre est $2 \times 1 + 1 = 3$. Dans la plupart de nos travaux, la notion de modèle de langage n'a pas été utilisée et nous avons attribué la même probabilité a priori à tous les modèles.

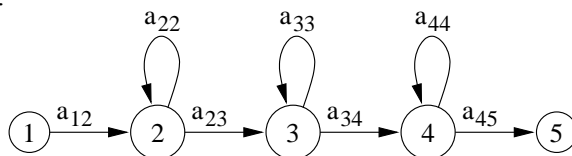


Figure 8: Modèle de Markov caché avec une architecture gauche-droite et trois états émetteurs.

L'apprentissage des HMM se fait sur le même corpus que celui utilisé pour la décomposition temporelle et la quantification vectorielle. L'initialisation des HMM prend en compte la transcription initiale T^0 obtenue par la combinaison « décomposition temporelle - quantification vectorielle » ou « décomposition temporelle - quantification vectorielle - multigramme ». Les modèles sont appris sans contexte et en contexte (avec un apprentissage itéré), voir [21], pour aboutir à un jeu de paramètres initiaux Λ^0 :

$$\Lambda^0 = \{\lambda_i^0\} = \arg \max_{\forall \Lambda} L(\mathbf{O}, \Lambda | T^0). \quad (2)$$

On répète ensuite, les étapes de segmentation à l'aide des modèles préalablement appris et de ré-estimation des paramètres de ces modèles :

- **segmentation**: $T^{m+1} = \arg \max_{\{M_1^N\}} L(\mathbf{O}, M_1^N | \Lambda^m, LM^m)$;
- **ré-estimation** des paramètres HMM : $\Lambda^{m+1} = \arg \max_{\{\Lambda\}} L(\mathbf{O}, \Lambda | T^{m+1})$;
- **terminaison** : on arrête si l'augmentation de la vraisemblance n'est plus significative, ou si le nombre d'itérations est plus grand qu'un seuil. Sinon, retour à la segmentation.

Nous avons observé que l'utilisation de cette technique d'affinement améliore la cohérence des modèles avec les données (au sens d'une augmentation de la vraisemblance), ainsi que la cohérence des segments acoustiques dans des différentes classes (la ressemblance des segments dans une classe devient meilleure).

Les techniques utilisées fournissent donc trois types de résultats : un *dictionnaire d'unités* (déterminé sur le corpus d'apprentissage), une *transcription* du corpus d'apprentissage utilisant ces unités et un *jeu de modèles HMM*.

3. Expériences - codage de parole à très bas débit

Le codage à très bas débit à l'aide des unités ALISP nous a servi de première vérification de nos approches. Pour une application de codage, où le passage au niveau lexical n'est pas indispensable, le critère pour évaluer la qualité des unités créées est simple : la parole à la sortie du décodeur doit différer le moins possible de la parole originale, et le débit binaire nécessaire pour transmettre

l'information sur les unités (et une information auxiliaire, comme nous allons le voir plus loin) doit rester bas.

L'approche consistant à effectuer une segmentation en phonèmes pour le codage de la parole à très bas débit, connue sous le nom de « *codage phonétique* » n'est pas nouvelle (cf. travaux de Ismail et Ponting [11], Ribeiro et Trancoso [18, 19], et autres). Notre approche consiste à trouver ces unités de manière automatique et sans aucune supervision.

On peut distinguer deux phases dans la technique proposée : la phase d'apprentissage et la phase de codage à très bas débit. Deux types d'unités sont définis : les unités de codage et les unités de synthèse (ou représentants). Dans la phase *d'apprentissage*, la détermination des unités de synthèse se fait pour chaque classe d'unités de codage. Pour chaque unité de codage, nous disposons de l'ensemble de segments du corpus d'apprentissage qui ont été étiquetés par cette unité. Les unités de synthèse sont obtenues en choisissant un nombre de représentants limité dans cet ensemble. Les critères de choix de ces représentants sont liés à la prosodie (longueur du segment en particulier).

La phase de *codage* à très bas débit utilise les unités obtenues lors de la phase d'apprentissage. En effet, le codeur-décodeur effectue deux opérations :

1. Le codeur effectue la reconnaissance des unités de codage dans le signal à coder. Puis il transmet les indices de ces unités au décodeur. Après reconnaissance des unités de codage, le codeur choisit pour chaque segment reconnu, une unité de synthèse qui lui est la plus proche au sens d'un critère spectral, parmi l'ensemble des représentants de synthèse affectés à cette classe de codage. Le codeur transmet de plus des paramètres représentant la prosodie du signal (fréquence fondamentale, énergie et longueur des segments).
2. Le décodeur utilise ces informations pour reconstituer un signal de parole par synthèse vocale. Lors de la synthèse, on concatène les unités de synthèse (en fait des segments de parole naturelle) pour obtenir de la parole synthétique.

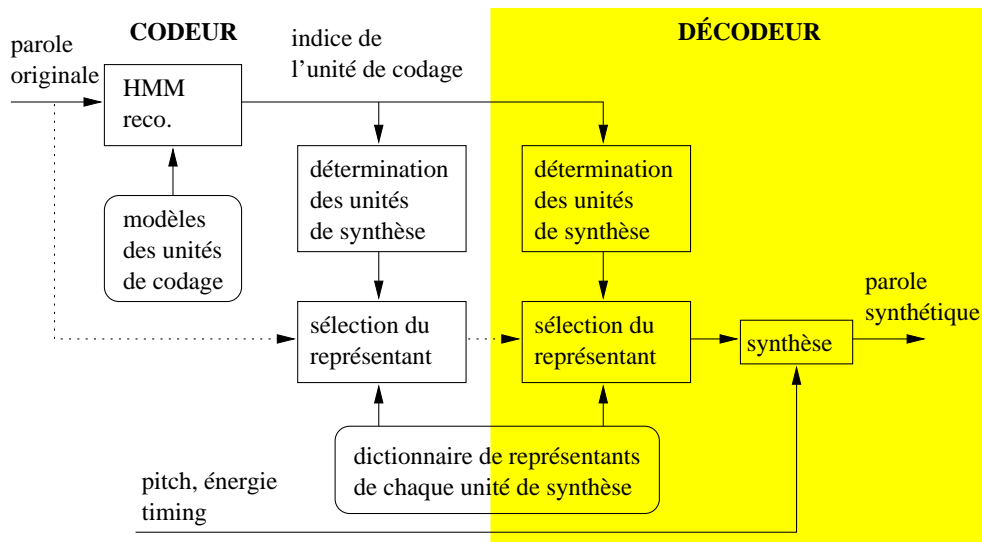


Figure 9: Codage et décodage de la parole: unités de codage, unités de synthèse et représentants.

Pour évaluer le *débit binaire* nécessaire pour la transmission de l'information concernant les unités, nous n'avons pas pris en compte les probabilités a priori des unités (nous n'avons pas effectué de codage entropique [9]), mais nous avons calculé le nombre de bits nécessaire pour la transmission de chaque unité M_i par $\log_2 Z$, où Z est la taille du dictionnaire. Le débit binaire moyen en bps (bits par seconde) est ainsi défini par:

$$R_u = \frac{\log_2 Z \sum_{i=1}^Z c(M_i)}{T_f Z \sum_{i=1}^Z c(M_i) l(M_i)} \quad (3)$$

où $c(M_i)$ est le nombre d'occurrences de M_i dans la chaîne encodée et T_f est le décalage entre trames acoustiques exprimé en secondes. La *qualité* de la parole après codage-décodage a été évaluée subjectivement par des tests informels.

Nous avons effectué les expériences en mode mono-locuteur, avec les données de deux corpus : Boston University Radio Speech Corpus (anglais américain) et Martin Ruzek (tchèque). Puis nous avons travaillé en mode multi-locuteur sur le corpus BD-BREF (français) distribué par ELRA².

3.1 Boston University Radio Speech Corpus

Les données de ce corpus américain distribué par « Linguistic Data Consortium »³ sont de qualité « Hi-Fi » (fréquence d'échantillonnage 16 kHz). Le corpus contient les enregistrements de 7 présentateurs professionnels. Nous avons utilisé les données d'un locuteur masculin - 78 minutes et d'un locuteur féminin - 83 minutes. Selon la provenance des enregistrements, les données ont été divisées en un corpus d'apprentissage (celles enregistrées à la radio) et de test (données enregistrées au studio de l'université de Boston).

Nous avons effectué la paramétrisation avec 16 coefficients LPC-cepstraux en trames de 20 ms (recouvrement 10 ms). La soustraction de la moyenne cepstrale a été faite pour chaque phrase. Nous avons ensuite appliqué la décomposition temporelle, ajustée afin de produire 15 cibles par seconde en moyenne. Sur les segments obtenus, nous avons appris un « codebook » de quantification vectorielle à 64 vecteurs-codes. L'apprentissage des HMM a été fait directement sur les transcriptions obtenues par la combinaison « décomposition temporelle - quantification vectorielle » (sans pré-traitement par les multigrammes). Le nombre réduit de 64 modèles HMM a permis un affinement avec 5 itérations de segmentation et de ré-estimation. Nous avons vérifié que la vraisemblance d'alignement des données avec les modèles augmentait. Nous avons ensuite testé une application des multigrammes sur la dernière segmentation HMM. Nous avons ainsi obtenu des dictionnaires de séquences de longueur variable de 1 à 6, de tailles 722 (pour le locuteur féminin) et 972 (pour le sujet masculin).

Pour le décodage, nous avons utilisé des unités de synthèse équivalentes à celles de codage, et nous avons choisi 8 représentants pour chaque classe d'unités. Nous avons réalisé une synthèse par prédiction linéaire et nous n'avons pas considéré le codage de la prosodie (les contours de F_0 et de l'énergie originaux étant introduits directement dans le synthétiseur). Les débits binaires (seulement pour le codage des unités et incluant les 3 bits nécessaires pour le codage du choix de représentant) obtenus sont donnés dans le Tableau 1 :

locuteur	féminin		masculin	
	apprentissage	test	apprentissage	test
Débit binaire sur l'ensemble de :				
HMM 6-ème génération [bps]	189.27	190.28	189.75	195.51
HMM 6-ème génération+multigrammes [bps]	135.91	145.09	141.86	156.02

Tableau 1 : Débits binaires obtenus pour le codage des unités.

La parole synthétisée a été jugée intelligible, avec une meilleure qualité pour les multigrammes (moins de distorsions sur les transitions).

² ELRA = European Language Resources Association.

³ Université de Pennsylvanie, <http://www ldc.upenn.edu/>

3.2 Corpus de Martin Ruzek

Cette base de données tchèque a été créée en coopération entre l'Université Technique de Brno, l'Université Masaryk de Brno⁴ et la Radio Tchèque, station Brno⁵. Nous avons numérisé à 11025 Hz deux bandes avec des textes lus par le célèbre acteur Martin Ruzek. Les longs paragraphes ont été éclatés en recherchant les minima de l'énergie, les fichiers ainsi obtenus ayant une longueur de 6 à 18 secondes. Nous avons rejeté les fichiers contenant des bruits de fond (musique, et autres). Ce corpus a été divisé en une partie réservée à l'apprentissage (7/8) et une partie réservée aux tests (1/8).

Nous avons paramétrisé les données dans les trames de 220 échantillons avec un recouvrement de 110 échantillons (20 et 10 ms environ). Nous avons utilisé 12 coefficients cepstraux. Nous avons également calculé la fréquence fondamentale, par une méthode FFT-cepstrale sur des trames plus longues (500 échantillons). Les autres traitements sont similaires aux expériences précédentes : environ 15 cibles de décomposition temporelle par seconde ; un « codebook » de 64 vecteurs-codes ; la même architecture de HMM et le même procédé de ré-estimation. Celle-ci a duré environ 8 heures sur un Pentium 233 MMX sous Linux.

Le codage a été réalisé de façon similaire aux expériences effectuées sur le Boston University corpus, mais nous n'avons pas utilisé l'allongement des unités par les multigrammes. Nous avons obtenu des débits de **173.26 bps** sur le corpus d'apprentissage et de **175.44 bps** sur le corpus de test.

La parole après codage-décodage est intelligible, mais n'est pas naturelle et souffre d'artéfacts audibles. Ces imperfections sont probablement dues à l'absence de lissage sur les frontières des unités, ou à la faiblesse de la technique de synthèse utilisée (une simple synthèse LPC). L'usage de techniques plus avancées : HNM « Harmonic Noise Model » ou PSOLA « Pitch-Synchronous Overlap and Add » devrait améliorer considérablement la qualité de la parole synthétique, tout en conservant un faible débit de transmission.

3.3 Corpus BREF

Pour évaluer nos algorithmes dans le cas indépendant du locuteur, nous avons utilisé une partie des locuteurs masculins de la base de données BREF. C'est un corpus grand vocabulaire, de textes français lus, dans un environnement non bruité. Les données textuelles sont choisies dans le journal « Le Monde » et sont réparties sur les 120 locuteurs. Nous avons choisi 40 locuteurs masculins pour nos expériences multi-locuteur.

Ces données sont échantillonnées à 16 kHz. Le protocole expérimental est identique à celui du Boston University corpus. La différence principale réside dans la quantité de données parole utilisée. Dans les expériences mono-locuteur, nous avons utilisé environ une heure de parole pour chaque locuteur, tandis que pour le cas indépendant du locuteur, nous avons utilisé environ 40 heures de parole provenant de 40 locuteurs différents. Pour les expériences de codage à très bas débit, sans l'utilisation de la technique multigrammes, nous avons obtenu des débits de **133 bps** pour le cas multi-locuteur. La qualité de la parole après codage-décodage est intelligible (du même ordre d'intelligibilité que celle obtenue dans le cas mono-locuteur). Dans ce travail, nous nous sommes surtout intéressés à la correspondance des unités obtenues automatiquement (que nous nommons unités ALISP), avec des unités phonétiques.

4. Étude des correspondances entre une segmentation ALISP et une segmentation acoustico-phonétique

4.1 Cas mono-locuteur

⁴ Nous remercions Ludek Bartek pour son support technique et sa patience pendant les enregistrements.

⁵ Nous remercions M. Jaroslav Vojacek, son directeur commercial, de nous avoir permis d'utiliser ces enregistrements pour nos recherches.

Pour le corpus Boston University, nous avons pu comparer la segmentation obtenue sur les données du locuteur féminin avec une segmentation acoustico-phonétique. Cette dernière a été réalisée par un système de reconnaissance de phonèmes et d'unités sub-phoniques (nous allons appeler les deux classes « unités phonétiques ») à l'université de Boston, et est jointe au corpus. Pour quantifier cette correspondance, nous avons d'abord mesuré les recouvrements des unités ALISP avec les unités phonétiques. Nous avons ensuite évalué la matrice de confusion \mathbf{X} , de taille $n_p \times n_a$ à l'aide des recouvrements relatifs :

$$x_{i,j} = \frac{\sum_{k=1}^{c(p_i)} r(p_{ik}, a_j)}{c(p_i)}$$

où $c(p_i)$ est le nombre d'occurrences du phonème p_i dans le corpus, et $r(p_{ik}, a_j)$ est un recouvrement relatif de la $k^{\text{ème}}$ occurrence de l'unité phonétique p_i avec l'unité ALISP a_j . Le recouvrement relatif se calcule à partir du recouvrement absolu (cf. Figure 10), par une simple normalisation par la longueur de l'unité phonétique: $r(p_{ik}, a_j) = R(p_{ik}, a_j) / L(p_{ik})$. La valeur $x_{i,j}$ quantifie alors la correspondance de la $i^{\text{ème}}$ unité phonétique avec la $j^{\text{ème}}$ unité ALISP sur tout l'ensemble de données étudiées. Un exemple de segmentation en unités phonétiques et unités ALISP du mot « wanted » est montré sur la Figure 11.

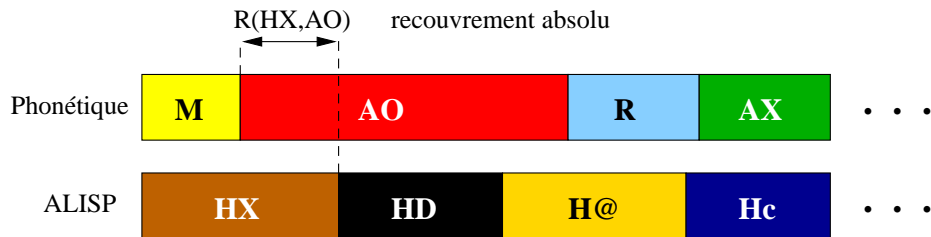


Figure 10: Représentation schématique du recouvrement des unités ALISP avec des unités phonétiques.

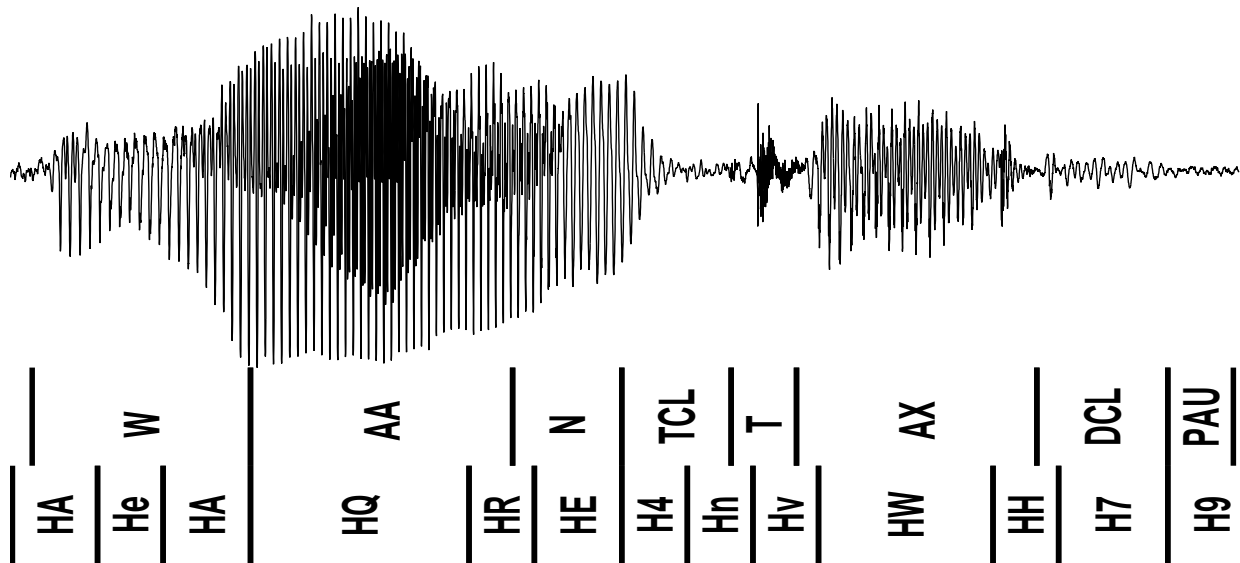


Figure 11: Le mot « wanted » (locuteur féminin de Boston University corpus), avec sa segmentation en unités phonétiques et unités ALISP.

Les unités ALISP utilisées dans cette expérience sont celles de la sous-section 3.1 (sans le post-traitement par les multigrammes). Les unités phonétiques sont données dans le Tableau 2. Le corpus distingue les voyelles accentuées des non accentuées mais nous ne nous sommes pas servis de ce classement. L'alphabet des étiquettes phonétiques est similaire à « l'ARPABET » défini pour la base de données classique TIMIT ([8] donne la conversion en alphabet phonétique [1]).

La matrice de confusion résultante (après réarrangement des colonnes pour obtenir une forme pseudo-diagonale) est donnée sur la Figure 13. Il est clair, que cette matrice quantifie la variabilité des unités. Elle montre, que la correspondance est consistante, mais pas biunivoque. Nous pouvons par exemple observer, que l'unité ALISP HA correspond à toutes les occlusives, mais aussi à une pause, et que l'unité H\$ présentant une corrélation prononcée avec l'unité SH est aussi liée à son correspondant voisin ZH et aux affriquées JH et CH, qui lui sont très proches acoustiquement. La Figure 12 présente une comparaison de la voyelle AA avec son correspondant ALISP HQ dans le domaine temporel et sur des spectrogrammes.

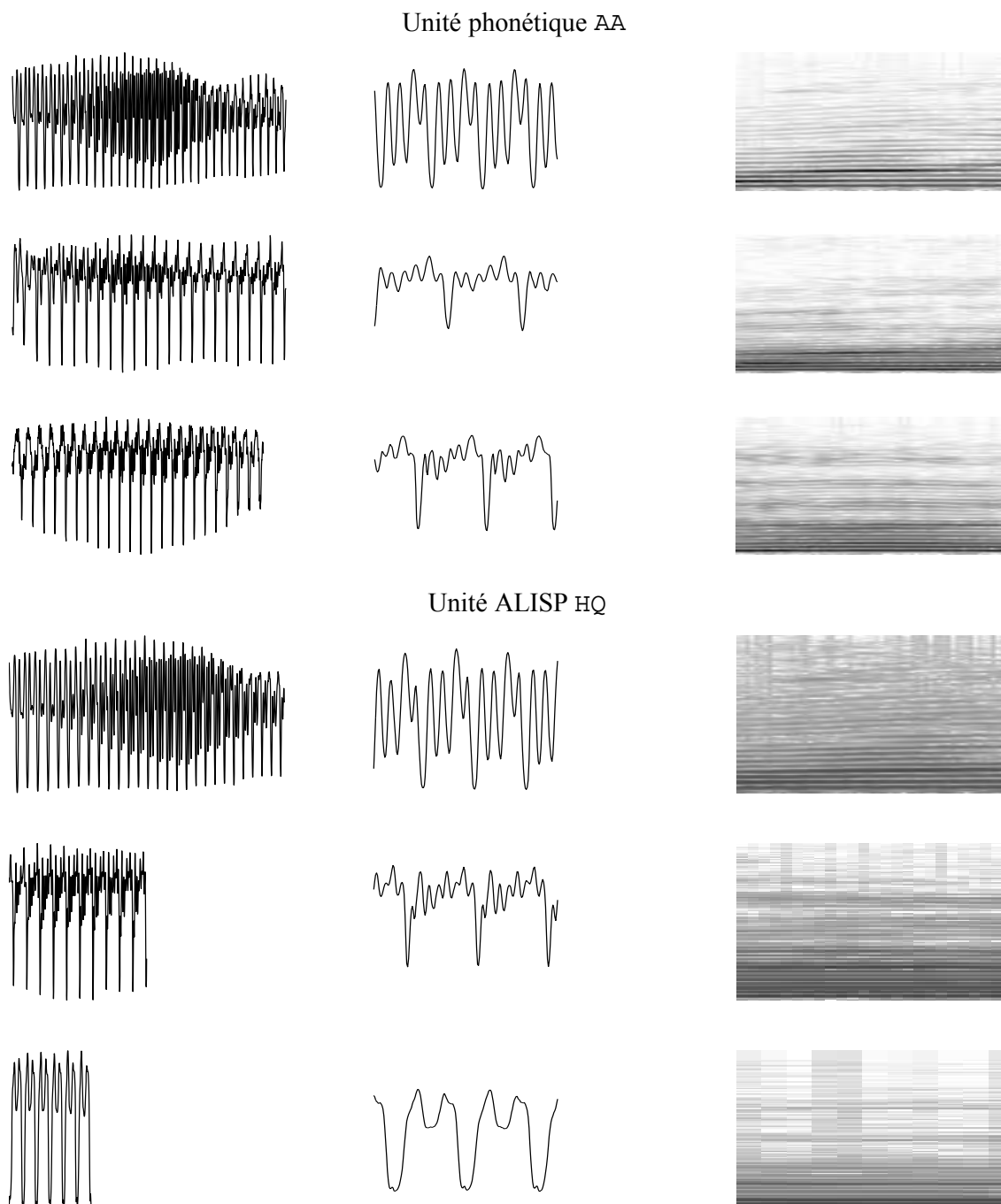


Figure 12: Comparaison de l'unité phonétique AA (3 exemples) avec son correspondant ALISP HQ (3 exemples). Gauche : domaine temporel ; centre : détail de 200 échantillons et droite : spectrogramme.

Nous avons également étudié la correspondance des unités ALISP avec des classes phonétique larges. Les étiquettes des unités phonétiques (Figure 10) ont été remplacées par celles des classes phonétiques CLO, AFF, FRI, NAS, DVG, VOY, AUT (cf. Tableau 2), et les recouvrements relatifs ont été calculés de la même façon que précédemment. La matrice de confusion est donnée sur la

Figure 14. Nous pouvons observer une cohérence des unités ALISP avec des classes phonétiques. Cette correspondance est la plus prononcée pour l'unité H@ liée aux affriquées. Pour mettre en correspondance de façon plus nette les unités ALISP avec les classes phonétiques, nous pourrions effectuer un « *clustering* » des unités de façon que les regroupements correspondent le mieux possible aux classes phonétiques.

Pour construire un système de reconnaissance de parole avec les unités ALISP, il faudrait trouver une correspondance plusieurs↔plusieurs entre les phonèmes et les unités ALISP (les travaux de Deligne [6] sur les multigrammes conjoints apportent des perspectives intéressantes). Une autre alternative consiste à construire le dictionnaire de prononciations directement à l'aide de ces unités (Fukada [7] propose une composition des unités apprises automatiquement, dans les mots et les phonèmes).

Classe phonétique	abbrev.	phonèmes
occlusions	CLO	BCL, DCL, GCL, PCL, KCL, TCL
Relâchement d'occlusion	OCC	B, D, G, P, T, K, DX
affriquées	AFF	JH, CH
Fricatives	FRI	S, SH, Z, ZH, F, TH, V, DH
Nasales	NAS	M, N, NG, EM, EN, NX
consonnes vocaliques	DVG	L, R, W, Y, HH, HV, EL
voyelles	VOY	IY, IH, EH, EY, AE, AA, AW, AY, AH
		AO, OY, OW, UH, UW, ER, AX, AXR
autres	AUT	PAU, H#, brth

Tableau 2 : Le jeu d'unités phonétiques du corpus Boston University, utilisé dans la comparaison monolocuteur.

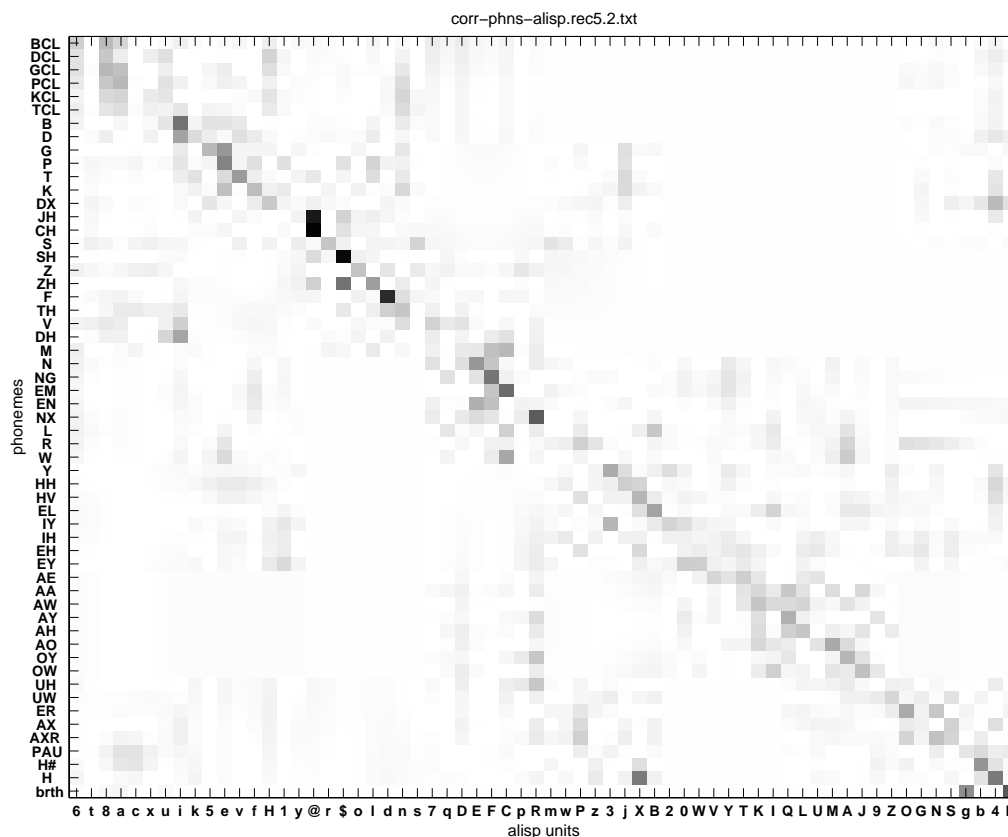


Figure 13: Correspondance de la segmentation ALISP avec une segmentation acoustico-phonétique pour le locuteur féminin du corpus Boston University. La couleur blanche correspond à une corrélation nulle, la noire à la valeur maximale de $x_{i,j}=0.806$.

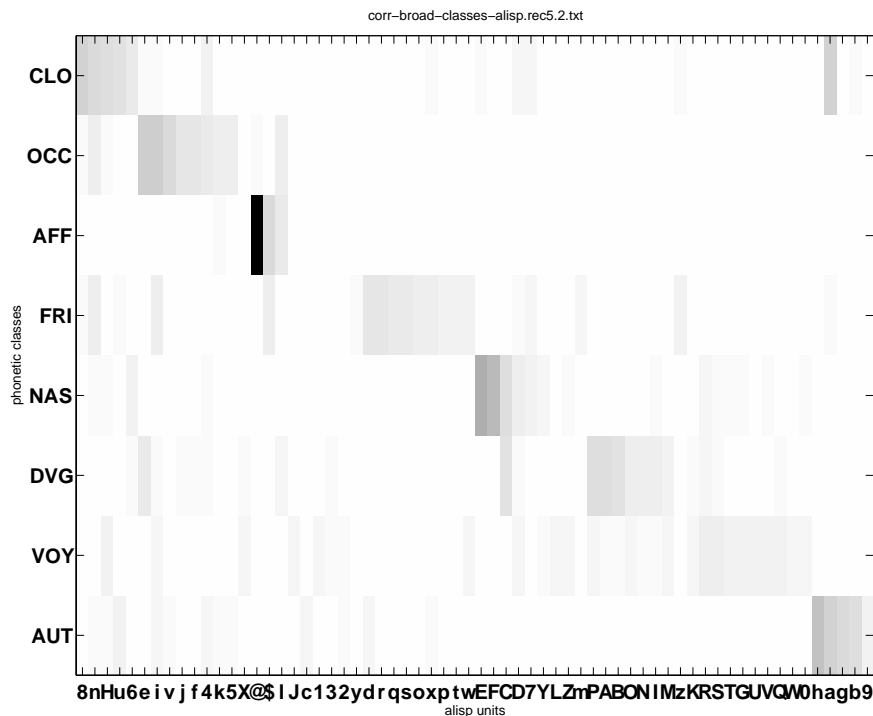


Figure 14: Correspondance de la segmentation ALISP avec des classes phonétiques larges pour le locuteur féminin du corpus Boston University. La couleur blanche correspond à une corrélation nulle, la noire à la valeur maximale de $x_{i,j}=0.7596$.

4.2 Cas multi-locuteur

Pour le corpus Bref, nous avons comparé la segmentation des unités ALISP, obtenue à partir des données de 40 locuteurs masculins, avec une segmentation acoustico-phonétique. La segmentation phonétique étant obtenue par une procédure automatique de phonétisation de textes, suivie d'un alignement par l'algorithme de Viterbi⁶.

Classe phonétique	abbrev.	phonèmes
occlusions	CLO	c l, v c l
relâchement d'occlusion (explosion)	OCC	b, d, g, k, p, t
Fricatives	FRI	S, Z, f, s, v, z
Nasales	NAS	m, n
consonnes vocaliques	DVG	R, j, l, w
voyelles	VOY	@, o~, A, E, U~, O, a~
		E, i, u, o, 2, 9, y
autres	AUT	#

Tableau 3: Le jeu d'unités phonétiques de Bref utilisé dans la comparaison multi-locuteur.

⁶ Les transcriptions phonétiques proviennent du projet Sirocco (<http://www.enst.fr/sirocco>). Nous remercions tout particulièrement Guillaume Gravier et François Yvon d'avoir mis ces transcriptions à notre disposition.

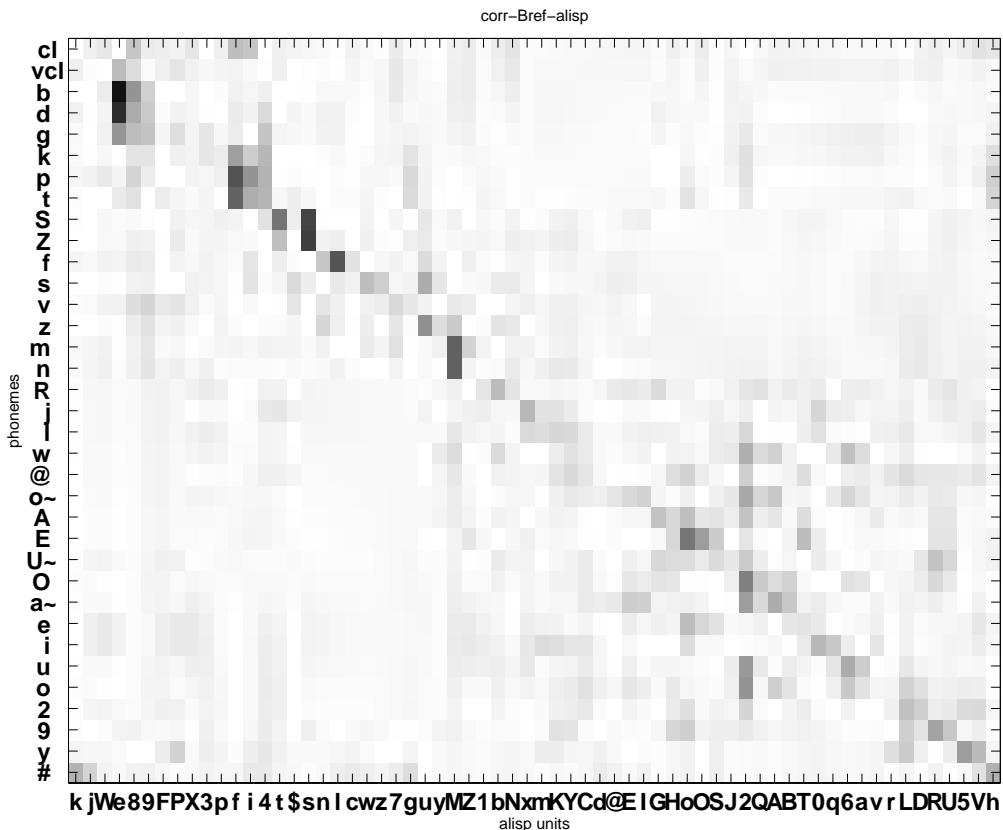


Figure 15: Correspondance de la segmentation ALISP avec une segmentation acoustico-phonétique pour 40 locuteurs masculins du corpus Bref. La couleur blanche correspond à une corrélation nulle, la noire à la valeur maximale de $x_{i,j}=0.45$.

La procédure de mise en correspondance entre les unités ALISP et les unités phonétiques est identique au cas mono-locuteur. Le nombre d'unités ALISP utilisé est égal à 64. Il est clair que ce ne sont pas les mêmes unités ALISP que les 64 unités ALISP mono-locuteur. L'ensemble de phonèmes utilisé pour le décodage phonétique de la comparaison multi-locuteur est résumé dans le Tableau 3.

La matrice de confusion résultante (après réarrangement des colonnes pour obtenir une forme pseudo-diagonale) est donnée sur la Figure 15. Il faut remarquer que le nombre d'unités phonétiques utilisé dans ce cas est de 35, alors que pour les expériences mono-locuteur, nous avons utilisé 57 unités phonétiques. On peut constater que la matrice de confusion présente une plus grande variabilité. Ceci est dû à la variabilité de la parole prononcée par des locuteurs différents. Dans le cas mono-locuteur, seule la variabilité intra-locuteur intervient. La variabilité inter-locuteur explique la correspondance plus floue entre les unités ALISP et les phones, dans le cas multi-locuteur. On peut néanmoins observer une corrélation prononcée entre les unités ALISP multi-locuteur H_e et H₈ et les relâchements d'occlusion (explosion) b, d et g. De même les unités ALISP H_f et H_i correspondent aux relâchements d'occlusion (explosion) p, t et k et à l'occlusion cl. On observe aussi une correspondance nette entre les unités ALISP H_s et H_t et les fricatives S et Z, qui sont très proches acoustiquement. La fricative f a son correspondant ALISP dans l'unité H_I. Les nasales m et n sont aussi relativement bien représentées par l'unité ALISP H_M. Parmi les unités ALISP qui sont les plus confuses, on peut indiquer l'unité H₂ (correspondant aux phonèmes A, U, a, k, o, u et w), et l'unité ALISP H₄ (correspondant à @, d, g, k et t). Un regroupement des phones en classes phonétiques larges donnerait vraisemblablement de meilleurs résultats.

5. Conclusions

Un des objectifs de l'Association Phonétique Internationale est de répertorier les unités segmentales (phones) utilisées dans les langues parlées. Un signal de parole quelconque peut alors être transcrit (par un phonéticien) comme une séquence de ces unités. Ce « codage » correspond en moyenne à un débit de l'ordre de 50 bps (correspondant à l'élocution d'environ 10 phones par seconde). Les efforts

pour automatiser ce processus (dénommé « décodage acoustico-phonétique » par les spécialistes de la reconnaissance automatique de la parole) n'ont pas encore abouti.

L'approche proposée dans cet article consiste à découvrir un ensemble d'unités à partir d'échantillons de parole sans aucun a priori sur la fonction linguistique de ces unités. L'ensemble des unités est déterminé automatiquement (de manière non-supervisée) sur un corpus d'apprentissage et de validation. Des outils (tels que la décomposition temporelle, la quantification vectorielle, les « multigrammes », la modélisation stochastique,...) ont été adaptés pour atteindre cet objectif. Une validation a été réalisée sur plusieurs langues pour le codage de la parole à très bas débit.

Il est alors intéressant d'analyser le degré de correspondance entre une transcription à l'aide des unités ALISP et une transcription phonétique traditionnelle. Le fait que cette correspondance soit imparfaite devrait permettre d'améliorer le codage lexical (et morphologique) en intégrant des variantes découvertes automatiquement aux nombreuses bases de données pour lesquelles une transcription orthographique est disponible (par exemple SpeechDat [20]).

Références bibliographiques

- [1] *International Phonetic Association (IPA) homepage*. <http://www.arts.gla.ac.uk/IPA/ipa.html>.
- [2] ATAL, B., Automatic Speech Recognition: a Communication perspective. *Proc ICASSP'99*, pp. I-457, 1999.
- [3] ATAL, B. S., Efficient coding of LPC parameters by temporal decomposition. *Proc. IEEE ICASSP 83*, pp. 81-84, 1983.
- [4] BIMBOT, F., *An evaluation of temporal decomposition*. Acoustic research department AT&T Bell Labs, 1990.
- [5] CERNOCKY, J., *Traitement de la parole s'appuyant sur des unités segmentales déterminées automatiquement : applications au codage à très bas débit et à la vérification du locuteur*. Université Paris XI Orsay, 1998.
- [6] DELIGNE, S., *Modèles de séquences de longueurs variables: Application au traitement du langage écrit et de la parole*. École nationale supérieure des télécommunications (ENST), Paris, 1996.
- [7] FUKADA, T., M. BACCHIANI, K. PALIWAL Y. SAGISAKA, Speech recognition based on acoustically derived segment units. *Proc. ICSLP 96*, pp. 1077-1080, 1996.
- [8] GAROFOLO, J.S., L.F.LAMEL, W.M. FISHER, J.G. FISCUS, D.S. PALLETT N.L. DAHLGREN, *DARPA-TIMIT acoustic-phonetic speech corpus*. NISTIR 4930, U.S. Department of Commerce, National Institute of Standards and Technology, Computer Systems Laboratory, 1993.
- [9] GERSHO, A., Advances in speech and audio compression. *Proc. IEEE*, **82(6)**, pp. 900-918, 1994.
- [10] GERSHO, A., *Vector quantization and signal compression*. Kluwer Academic Publishers, 1996.
- [11] ISMAIL, M. K. PONTING, Between recognition and synthesis - 300 bits/second speech coding. *Proc. EUROSPEECH 97*, pp. 441-444, Rhodes, Greece, 1997.
- [12] KOHONEN, T., *Self organization and associative memories*. Springer Verlag, Berlin, 1984.
- [13] KOHONEN, T., *Self-Organizing Maps, 30 Series in Information Sciences*. Springer Verlag, Berlin, 1997.

- [14] KRUSKAL, J. B., An Overview of Sequence Comparison. In SANKOFF, D J. B. KRUSKAL (editors): *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pp. 1-44, Reading, Massachusetts, 1983. Addison-Wesley Publishing Co.
- [15] MARCKEN, C. de, *The unsupervised acquisition of a lexicon from continuous speech*. A.I.Memo No. 1558, C.B.C.L. Memo No. 129, Massachusetts Institute of Technology, Artificial Intelligence Lab. and Center for Biological and Computational Learning, Dpt. of Brain and Cognitive Sciences, 1996.
- [16] RABINER, L.R., A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77(2)**, pp. 257-286, 1989.
- [17] RABINER, L. R. L. W. SCHAEFFER, *Digital processing of speech signals*. Prentice Hall, 1978.
- [18] RIBEIRO, C.M. I.M. TRANCOSO, Application of speaker modification techniques to phonetic vocoding. *Proc. ICSLP 96*, pp. 306-309, Philadelphia, 1996.
- [19] RIBEIRO, C.M. I.M. TRANCOSO, Phonetic vocoding with speaker adaptation. *Proc. EUROSPEECH 97*, pp. 1291-1294, Rhodes, Greece, 1997.
- [20] WINSKI, R., *Definition of Corpus, Scripts and Standards for Fixed Networks*. SpeechDat-II, 1997. Deliverable SD 1.1.1., workpackage WP1, <http://www.speechdat.org>.
- [21] YOUNG, S., J. JANSEN, J. ODELL, D. OLLASON P. WOODLAND, *The HTK book*. Entropics Cambridge Research Lab., Cambridge, UK, 1996.