La segmentation et le regroupement par locuteurs pour l'indexation de documents audio

Perrine Delacourt

Le 5 septembre 2000

En essayant continuellement, on finit par réussir.
Donc, plus ça rate, plus on a de chances que ça marche.

Devise Shadok

Science sans conscience,
n'est que ruine de l'âme

Rabelais.

Remerciements

Mes premiers remerciements vont à Mr Claude Guéguen et à Mr Christian Wellekens, respectivement directeur de l'Institut Eurécom et chef du département Multimédia à mon arrivée début 1997, qui m'ont accueilli et qui m'ont permis de réaliser cette thèse.

Je remercie les membres de mon jury, en commançant par Mr Denis Jouvet, qui a accepté de présider ce jury et qui m'a accueilli dans son service lors d'un séjour au CNET de Lannion.

Mme Régine André-Obrecht et Mr Frédéric Bimbot ont accepté d'être rapporteurs de cette thèse, lourde tâche qu'ils ont accepté malgrè leur emploi du temps chargé. Qu'ils en soient vivement remerciés.

Enfin, merci à Messieurs Jean-François Bonastre, Jean-Claude Junqua et Christian Wellekens, examinateurs de cette thèse. Je les remercie pour les (nombreuses !) lectures et pour leurs remarques constructives.

Je voudrais remercier l'ensemble du personnel d'Eurécom pour sa disponibilité et sa gentillesse. Je voudrais remercier également l'ensemble du département Multimédia, et plus particulièrement Sophie et Pascal, que j'ai embêtés de nombreuses fois et qui ont toujours répondu à mes attentes. Enfin, mes remerciements vont à l'ensemble des doctorants et des assistants qui se sentent encore un peu doctorants pour la bonne ambiance qui règne à Eurécom et en dehors d'Eurécom.

Au cours de ma thèse, j'ai travaillé en collaboration avec le Laboratoire d'Informatique d'Avignon. Les quelques séjours effectués sur place se sont avérés fructueux et j'en garde un excellent souvenir, tant sur le plan scientifique que sur le plan humain. Je remercie donc les membres du LIA et plus particulièrement l'équipe des RALeurs : Jef, Corinne, Teva et Sylvain qui m'ont accueilli comme si je faisais partie des leurs.

J'ai également passé deux semaines au CNET de Lannion et je voudrais remercier Denis Jouvet, Alexandre Ferrieux, Roger Lochou et Delphine Charlet pour l'accueil qu'ils m'ont réservé et pour l'aide qu'ils m'ont apportée durant ce séjour.

Je tiens à remercier mon directeur de thèse, Christian Wellekens, pour la pleine confiance qu'il m'a accordée au cours de ces trois années, que ce soit pour mon travail de thèse ou pour mon travail d'assitante. Je le remercie également pour son soutien dans mes activités annexes à la thèse. Enfin, je dois dire que j'apprécie sa bonne humeur, ses récits ô combien pitorresques et son érudition.

La thèse n'est pas faite que de bons moments. Aussi, je voudrais remercier vivement Stéphane Marchand-Maillet qui a su me redonner le courage et la motivation pour terminer cette thèse. Je remercie également Laure-Anne, Stéphane R., Stéphane MM., Sophie et Pierre, Franck et Guénaëlle, Laurence, Bip et Frédo et Cécile et Philippe pour leur soutien dans les moments difficiles.

Parmi les bons moments (parce qu'il y en a aussi heureusement), je crois que je retiendrai les Troisièmes Rencontres Jeunes Chercheurs que nous avons organisées avec Corinne. Je voudrais d'ailleurs la remercier car sans sa motivation et son énergie, ses rencontres n'auraient pas rencontrer ce succès. Elle m'a aussi appris à travailler conjointement, ce qui n'a pas été sans mal je l'avoue. Je lui souhaite une longue carrière dans la reconnaissance du locuteur. Je remercie également Jef, Teva et Franck qui nous ont énormément aidé et qui ont largement contribué à la réussite de ces rencontres.

Je tiens aussi à remercier Sophie, Pierre, Stéphane R., Katia, Neda, Sergio, Benoît, Fernanda, Stéphane D., Stéphane MM., Philippe, Cécile, Guënaelle, Franck, Laurence, Alain, Elisabeth... pour leur amitié.

Je remercie également mes parents, mes soeurs et frère et leurs petites familles respectives pour leur amour et leur soutien.

Enfin, je n'ai pas de mot assez fort pour exprimer ma reconnaissance envers Franck. Il m'a constamment soutenu, encouragé et entouré. Il m'a supporté (dans tous les sens du terme) dans les moments difficiles. Je l'ai également beaucoup sollicité et il a toujours accepté de m'aider, même si la tâche n'était pas des plus agréables. Il m'a dit un jour qu'il serait fier d'avoir une femme docteur et c'est ce qui m'a poussé à continuer. Aussi, je lui dédie cette thèse.

Table des matières

Introduction générale

Le traitement de l'information multimedia requiert de nouveaux outils tels des analyseurs de contenus ou indexeurs. Parmi ceux-ci, l'indexation par locuteurs d'un document audio, qui consiste à reconnaître la séquence de locuteurs engagés dans la conversation, tient une place essentielle. Il s'agit de savoir qui parle et quand afin de saisir la cohérence du dialogue.

Au cours de cette thèse, nous proposons tout d'abord un système d'indexation qui répond aux hypothèses que nous nous sommes fixés. Ces hypothèses sont les suivantes : aucune connaissance a priori sur les locuteurs ou sur le langage, le nombre de locuteurs est inconnu et les personnes ne parlent pas simultanément. Ce système d'indexation se décompose en plusieurs étapes : la segmentation en locuteurs, le regroupement des segments appartenant au même locuteur, la modélisation des locuteurs et enfin, la reconnaissance de la séquence de locuteurs à l'aide des modèles de locuteurs obtenus. Dans la suite de cette thèse, nous nous concentrons sur les deux premières étapes, à savoir la segmentation et le regroupement en locuteurs, tâches qui ont été peu étudiées jusqu'à présent.

En introduction de cet ouvrage, nous explicitons au chapitre 1 la problématique de l'indexation en général. Nous présentons plus particulièrement l'indexation par locuteurs d'un document audio et nous mentionnons le contexte applicatif. Au chapitre 2, nous détaillons un exemple concret de système d'indexation par locuteurs réalisé dans le cadre du contrôle aérien. Nous nous appuyons sur cet exemple, pour proposer notre propre système d'indexation par locuteurs.

La partie I de ce manuel aborde la segmentation en locuteurs. Nous en dressons tout d'abord un état de l'art au chapitre 3. Le chapitre 4 présente les deux techniques de segmentation en locuteurs que nous proposons. La première technique repose sur la détection des silences et la seconde technique repose sur la détection des changements de locuteurs. Enfin, le chapitre 5 présente tout d'abord, les méthodes d'évaluation, puis, les performances obtenues par ces deux techniques de segmentation. Enfin, nous concluons sur la segmentation en locuteurs.

La partie II de cette thèse est consacrée au regroupement des segments par locuteur. Comme pour la segmentation, nous passons en revue les techniques de regroupement existant dans la littérature au chapitre 7. Parmi les techniques de regroupement exposées, nous nous intéressons plus particulièrement au regroupement hiérarchique alliant rapport de vraisemblance et critère d'Information Bayesien. Nous lui adjoignons des pré-traitement et post-traitement, compte-tenu de notre contexte. Ces traitements sont décrit au chapitre 8. Finalement, les résultats expérimentaux obtenus par cette technique de regroupement sur différentes bases de données de parole sont analysés au chapitre 9. Ce dernier chapitre clôt également la partie consacrée au regroupement par locuteurs.

L'étape de modélisation des locuteurs n'est pas détaillée dans ce manuel car c'est un problème qui est largement étudié par ailleurs. Le volume de données disponibles, quoique faible encore, est cependant augmenté par le regroupement en locuteurs.

Enfin, la partie III conclut l'ensemble de ce travail en reprenant les résultats majeurs obtenus pour la segmentation et le regroupement en locuteurs. Elle détaille les perspectives, notamment les étapes suivantes du système d'indexation. Elle évoque également des directions de recherches pour l'amélioration du système d'indexation par locuteurs proposé dans ce document.

Chapitre 1

Position du problème

1.1 Pourquoi l'indexation ?

Avec la multiplication des chaînes de télévision et de radio et grâce à des capacités de stockage sans cesse grandissantes, des milliers d'heures d'émissions sont stockées chaque année par des instituts d'archivage, tel que l'Institut National de l'Audiovisuel (INA). Concernant ce dernier, voici quelques chiffres : 45 ans d'archives télévisuelles correspondant à 300.000 heures de programmes nationaux et 60 ans d'archives radiophoniques correspondant à 400.000 heures de programmes radio.

De plus, avec la numérisation de l'information, nous assistons à l'explosion de bases de données multimedia. Outre les problèmes de stockage et d'architecture inhérents à la construction de telles bases de données, le problème de l'accessibilité aux données se pose. La consultation doit en effet être aisée : la recherche de l'information doit se faire rapidement et facilement. Il est alors nécessaire d'indexer ces bases de données pour faciliter et accélérer la recherche de l'information souhaitée.

Parmi les données multimedia, certaines sont plus facilement accessibles que d'autres, au moins en termes de rapidité d'accès. Par exemple, lire un texte pour y retrouver un mot prend moins de temps que d'écouter ce texte pour y retrouver le même mot. Par ailleurs, il est préférable de pouvoir accéder directement aux zones d'intérêt du document audio plutôt que d'avoir à en écouter l'intégralité pour retrouver ces zones. D'où la nécessité d'indexer les documents multimedia, et plus particulièrement les documents audio. Il s'agit alors d'associer à chaque document audio un fichier décrivant sa structure, relativement à la clé d'indexation choisie.

Concernant les documents audio, plusieurs clés d'indexation, et par conséquent, de clés de recherche sont envisageables. La clé d'indexation/de recherche peut être un mot : des techniques de *wordspotting* (par exemple [Gelin et al. 97]) sont alors appliquées pour la recherche de ce mot dans le document audio. La clé d'indexation/de recherche peut également être un sujet, un thème. Dans ce cas, tous les mots ou expressions relatifs à ce sujet sont recherchés par des techniques de *topic detection* (par exemple [Fiscus et al. 99]). Une autre clés d'indexation/de recherche est l'identité du locuteur. Il peut être intéressant de trouver les moments où un locuteur en particulier parle. Des techniques de poursuite de locuteur (*speaker tracking*, par exemple [Rosenberg et al. 98, Magrin-Chagnol. et al. 99]) sont alors mises en œuvre. Il peut être aussi intéressant de connaître la séquence ou l'enchaînement des locuteurs dans un document audio. La connaisance de cette séquence repose sur l'indexation par locuteurs de ce document audio. C'est ce dernier problème que nous nous proposons de traiter dans le présent document.

1.2 Indexation par locuteurs d'un document audio

L'indexation par locuteurs d'un document audio consiste à reconnaître la séquence ou l'enchaînement des locuteurs. En d'autres termes, il s'agit de savoir qui parle et quand. Le résultat du processus d'indexation sera de la forme suivante : le locuteur A parle de l'instant $t1$ à l'instant $t2$, puis le locuteur B intervient de $t3$ à $t4$, puis A reprend la parole de $t5$ à $t6$, ensuite le locuteur C parle de $t7$ à $t8$, etc... Il est important de préciser que l'identité des locuteurs A, B ou C n'est pas pour autant connue. Dans notre contexte, il s'agit juste de reconnaître que c'est le même locuteur qui parle entre les instants $t1$ et $t2$ et entre les instants $t5$ à $t6$. Connaître l'identité des locuteurs est un problème d'identification du locuteur, problème qui a donné lieu à de multiples publications. C'est pourquoi nous ne traiterons pas ce sujet. Le lecteur pourra cependant consulter un état de l'art dans ce domaine [Furui 95].

Afin de laisser le champ à un large éventail d'applications, nous nous proposons d'indexer les documents audio par locuteurs avec les hypothèses suivantes :

- **pas de connaissance a priori sur les locuteurs et sur la langue** (pas de modèle, pas de phase d'entraînement). En effet, il n'y a pas toujours à disposition des données d'entraînement pour construire un modèle sophistiqué de locuteur. Par exemple, pour un journal radio-diffusé, il est rare de posséder des données d'entraînement pour une personne interviewée lors d'un reportage. Quant à la langue, il est possible qu'un document audio mélange plusieurs langues. Par exemple, dans un journal télévisé, il arrive qu'une personne s'exprime dans une langue étrangère et que ses propos soient traduits par la suite ou même simultanément.

- **le nombre de locuteurs est inconnu**. Nous fixons cette hypothèse pour rester dans le cas le plus général possible et pouvoir ainsi traiter tout type de document. De plus, cette hypothèse est proche de la réalité : autant nous pouvons éventuellement supposer que le nombre de locuteurs est de deux dans une conversation téléphonique (si ces locuteurs ne sont pas en pluri-conférence), autant, il devient plus délicat de faire une hypothèse sur le nombre d'intervenants dans un journal télévisé ou encore sur le nombre de personnes qui ont laissé un message sur une boîte vocale.

- **les personnes ne parlent pas simultanèment**. Cette hypothèse peut sembler d'une part restrictive au regard des deux précédentes et d'autre part, peu réaliste. En effet, il arrive souvent qu'au cours d'une conversation, une personne commence à parler alors que la précédente n'a pas achevé sa phrase. Il y a donc recouvrement des paroles des deux locuteurs. Cependant, ce type d'événement est difficile à indexer, même "à la main". Faut-il considérer qu'il n'y a qu'une personne et dans ce cas, laquelle ou faut-il considérer qu'il y a deux personnes ? Enfin, cette hypothèse reste réaliste dans le cadre de journaux radio- ou télé-diffusés car la parole est bien souvent préparée et non spontanée. Il y a donc peu de recouvrement de parole.

- **uniquement de la parole**. Les documents audio que nous traitons contiennent uniquement de la parole. Des travaux existent sur la séparation Parole/Bruit/Musique/etc... (cf [Montacie et al. 98, Seck et al. 99, Williams et al. 99]). Nous pouvons donc supposer un pré-traitement par l'une de ces techniques de séparation et ne travailler alors qu'avec des documents contenant uniquement de la parole.

- **pas de contrainte de temps réel**. L'indexation est une tâche qui est réalisée une fois pour toute sur une base de données et c'est de plus un processus "hors-ligne" (*off-line*). Aussi, elle peut se permettre de prendre un temps non négligeable. Par contre, il est nécessaire que la recherche d'informations dans cette base de données se fasse en temps réel. Mais là n'est pas notre propos...

1.3 Applications

Maintenant que nous avons vu sous quelles hypothèses l'indexation par locuteurs va être réalisée, nous nous intéressons aux applications potentielles de cette indexation. L'indexation par locuteurs trouve ses applications dans des domaines aussi variés que :

- **l'indexation de bases de données audio**. C'est en effet son premier rôle. Couplée à un processus d'identification du locuteur, elle peut permettre par exemple la recherche de tous les discours prononcés par telle personnalité politique. Cela peut servir par exemple à calculer son temps de parole au cours d'une campagne électorale (ce calcul est fait à l'heure actuelle "manuellement" par le Conseil Supérieur de l'Audiovisuel). Cela peut aussi permettre la recherche des paroles du journaliste-présentateur et l'extraction à partir de ces paroles des thèmes abordés dans le journal télévisé.

- **la poursuite de locuteur ([Rosenberg et al. 98, Magrin-Chagnol. et al. 99])**. Dans un sytème de poursuite de locuteur, un modèle du ou des locuteurs "cible" est disponible. Le processus de vérification du locuteur ([Furui 95]) cible se fait en général sur quelques trames[1] (de l'ordre de quelques dizaines). Une pré-segmentation en locuteurs peut permettre de réaliser ce processus de vérification non plus sur quelques dizaines de trames mais sur quelques centaines de trames rendant ainsi la décision de vérification plus robuste. Ces quelques centaines de trames sont contenues soit dans un segment relatif à un seul locuteur, soit dans un groupe de segments, tous relatifs au même locuteur. Nous avons d'ailleurs appliqué ce principe lors des évaluations NIST 1999 sur la tâche de poursuite de locuteur. En annexe A se trouve un article reprenant ces travaux [Bonastre et al. 00a, Bonastre et al. 00b].

- **la recherche des messages par locuteurs sur un répondeur téléphonique ou sur une boîte vocale**. Avec l'explosion de la téléphonie, de plus en plus de services sont proposés. Parmi ces services, figureront peut-être bientôt la classification par locuteurs des messages déposés sur un répondeur téléphonique ou sur une boîte vocale. Cette classification reposera alors sur une indexation par locuteurs de ces messages.

- **la transcription automatique de documents audio, en particulier nouvelles radio- ou télé-diffusées ([Woodland et al. 97, Gauvain et al. 98])**. Les systèmes de transcription automatique utilisent des modèles de parole pré-entraînés sur de larges bases de données, de sorte qu'ils ne contiennent aucune spécifité du locuteur. Or, il

1. Le signal de parole est paramétrisé, i.e. des caractéristiques pertinentes sont extraites pour former les vecteurs acoustiques, appelés encore trames d'analyse.

a été prouvé que lorsque ces modèles sont adaptés aux locuteurs présents dans le document audio, alors le taux de reconnaissance s'en trouvait amélioré. De plus, cette adaptation au locuteur nécessite peu de données dudit locuteur ([Gauvain et al. 94, Leggetter et al. 95]). Aussi, une étape préliminaire dans ces systèmes de transcription automatique consiste à indexer par locuteurs. Les données d'un locuteur servent alors à adapter les modèles de parole à ce locuteur et le processus de transcription peut alors commencer sur les segments de ce locuteur. Et ainsi de suite pour tous les locuteurs présents dans le document audio.

Enfin, nous pouvons mesurer toute l'importance de l'indexation de documents multimedia, en remarquant qu'elle constitue une étape préliminaire à toute tâche de recherche d'informations (*Information Retrieval*) dans une base de données multimedia, domaine de recherches également en plein essor.

Chapitre 2

Indexation par locuteurs : un exemple, une architecture

Bien avant la vogue du multimedia, le probème de l'indexation de communications pilotes/contrôleurs aériens a été traité par la société américaine BBN. Les principes de l'indexation, ainsi que les problèmes afférents, sont définis et sont présentés dans ce chapitre. Suite à cette présentation, nous mettons en évidence les points forts et les points faibles de cette application. Enfin, à partir de cette analyse, nous proposons l'architecture de notre propre système d'indexation.

2.1 Segmentation et indexation par locuteurs : un exemple

L'application réalisée chez BBN [Gish et al. 91, Siu et al. 92] consiste à retrouver automatiquement à partir des dialogues enregistrés entre contrôleurs aériens et pilotes, les instructions données dans le but d'améliorer le trafic aérien de l'aéroport de Dallas-Fort Worth. Ces dialogues sont des radio-transmissions. Les contrôleurs pouvant utiliser la même fréquence, il est possible qu'il y ait plusieurs contrôleurs aériens engagés dans le dialogue. Il en est de même pour les pilotes.

Cette application se décompose en plusieurs phases :

1. **Segmentation du signal d'entrée** : les dialogues enregistrés doivent être segmentés et indexés par locuteurs de manière à obtenir des transmissions ne contenant les paroles que d'un seul locuteur.
2. **Formation des dialogues** : un enregistrement pouvant contenir plusieurs dialogues, il faut reconstituer chaque dialogue pilote-contrôleur.
3. **Identification des vols** : le contenu des transmissions est utilisé pour identifier le vol.
4. **Classification du scénario** : le contenu des transmissions est utilisé pour déterminer le scénario complet suivi par le vol considéré.

Pour une vision complète du système développé, notamment pour les aspects intégration et temps réel, le lecteur se reportera à [Rohlicek et al. 92] et à [Denenberg et al.93].

Dans ce qui suit, nous nous intéressons plus particulièrement à la première phase, la segmentation et indexation par locuteurs, pour laquelle nous détaillons les différentes étapes.

Les hypothèses sont les suivantes :

- aucune information a priori sur les locuteurs n'est disponible
- le nombre de locuteurs est inconnu

La procédure employée peut être qualifiée d'apprentissage séquentiel car les informations acquises à l'issue d'une étape sont utilisées pour améliorer les modèles à l'étape suivante. Cette procédure est aussi non supervisée puisqu'aucune information sur les locuteurs n'est disponible.

2.1.1 Segmentation acoustique

Le but de cette segmentation acoustique est d'obtenir des segments n'appartenant qu'à un seul locuteur ou des segments de silence, les plus longs possibles.

Les vecteurs acoustiques sont composés de coefficients Mel-cepstraux ([Rabiner et al. 78]) calculés toutes les 10 ms par des fenêtres d'analyse de 20 ms. Ces vecteurs forment les trames acoustiques. Les segments sont des séquences de trames acoustiques adjacentes.

La première méthode utilisée s'appuie sur une recherche par programmation dynamique des points de discontinuité les plus probables dans le spectre ([Cohen 81]).

La deuxième méthode utilisée est beaucoup plus basique mais se révèle tout aussi efficace pour cette application et est d'une complexité bien moindre. Les auteurs considèrent tout simplement des segments de 20 trames acoustiques qui sont étiquetés par la suite.

2.1.2 Séparation Parole/Bruit

L'étape suivante consiste à séparer les segments de parole des segments de bruit. Cette étape se déroule de la manière suivante :

1. identifier des segments de parole et des segments de bruit avec un haut niveau de confiance. Les radio-transmissions étant réalisées avec un contrôle automatique de gain, il est impossible de se baser sur l'énergie pour détecter de la parole. Le processus de séparation s'appuie donc sur la variabilité de l'énergie. L'hypothèse est que la variabilité de l'énergie est plus importante pour un segment de parole que pour un segment de bruit. La mesure de variabilité de l'énergie utilisée est la valeur médiane de la déviation absolue de l'énergie, appelée score MAD ("Median Absolute Deviation"). La déviation absolue correspond à la valeur absolue de la différence entre l'énergie de la trame et l'énergie médiane du segment. Les segments obtenant un score MAD élevé sont identifiés comme segments de parole. A l'inverse, les segments dont le score MAD est parmi les plus faibles sont identifiés comme segments de silence.

2. utiliser ces segments pour initialiser un modèle de mixtures de Gaussiennes (*Gaussian Mixture Model* ou *GMM*, cf [Reynolds 95]) à deux états : un état pour la parole et un état pour le bruit.

3. entraîner ce modèle à l'aide de l'algorithme EM (Expectation-Maximisation [Dempster et al. 77, Moon 96]))

4. utiliser ce modèle pour classifier tous les segments en parole ou en bruit. La vraisemblance $p(x)$ d'observer une séquence de vecteurs acoustiques x peut s'écrire :

$$\begin{aligned} p(x) &= p(x,B) + p(x,P) && \text{(2.1a)} \\ &= p(x|B)p(B) + p(x|P)p(P) && \text{(2.1b)} \end{aligned}$$

   $p(x, B)$ est la vraisemblance d'observer la séquence x et d'observer du bruit B. De même, $p(x, P)$ est la vraisemblance d'observer la séquence x et de la parole P. $p(B)$ et $p(P)$ sont respectivement les probabilités a priori du bruit et de la parole. Enfin, $p(x|B)$ est la vraisemblance conditionnelle d'observer x sachant que nous observons du bruit et $p(x|P)$ la vraisemblance conditionnelle d'observer x sachant que nous observons de la parole.

   Chaque segment est étiqueté bruit ou parole selon que le terme correspondant au bruit $p(x, B)$ ou que le terme correspondant à la parole $p(x, P)$ soit le plus élevé.

Les auteurs précisent qu'étant donnée la nature de leur signal, notamment les différences de canaux entre les transmissions des pilotes et celles des contrôleurs aériens et les silences qui séparent les transmissions, il est relativement aisé d'obtenir des segments ne contenant qu'un seul locuteur, contrairement à d'autres applications.

2.1.3 Regroupement en segments longs

L'étape suivante consiste à regrouper les segments précédemment obtenus en segments plus longs, chacun étant soit du bruit, soit un segment n'appartenant qu'à un seul locuteur.

L'approche la plus simple consiste à regrouper les segments contigus ayant la même étiquette. Cette approche peut amener à regrouper des segments appartenant à des locuteurs différents dont les paroles ne sont pas séparées par des intervalles de bruit ou alors à séparer les paroles d'un même locuteur contenant des petits intervalles de silence. Néanmoins, la segmentation obtenue sert à l'initialisation des modèles de locuteurs.

Pour ce qui suit, seuls les segments correspondant à de la parole sont conservés.

2.1.4 Construction des modèles de locuteurs

Le but est ici de construire les modèles de locuteurs pour les pilotes et pour les contrôleurs. Plusieurs pilotes peuvent intervenir dans la conversation mais dans le contexte de cette application, ils sont considérés comme ne formant qu'un seul pilote. Par contre, chaque contrôleur présent dans la conversation est pris en compte. Pour construire les modèles de locuteurs, la procédure est la suivante :

1. **Découpage en intervalles** : la conversation est découpée arbitrairement en intervalles I_m $(m = 1, .., M)$ de 50 segments (hormis les segments de bruit). Ce choix de 50 segments assure dans la majeure partie des cas des intervalles où un seul et unique contrôleur intervient.

2. **Séparation pilote(s)/contrôleur** L'étape suivante consiste à regrouper les segments de pilote(s) d'une part et les segments de contrôleur d'autre part dans chaque intervalle I_m (les segments de bruit ne sont plus considérés). Pour ce faire, une méthode de regroupement hiérarchique (*hierarchical clustering*) est utilisée. Elle est détaillée dans

[Gish et al. 91] et nous l'exposons ici succintement. Nous reviendrons sur ce regroupement au chapitre 7.

Une distance est calculée entre toutes les paires de segments possibles dans l'intervalle I_m. Puis, à chaque itération, les segments ou groupes de segments les plus proches au sens de la distance utilisée sont regroupés. Ce processus est répété jusqu'à n'obtenir plus que deux groupes de segments, l'un correspondant au(x) pilote(s), l'autre au contrôleur.

La distance inter-segments utilisée dans cette application repose sur le test d'hypothèses suivant :

- H_0 : les segments x_i et x_j sont générés par le même locuteur
- H_1 : les segments x_i et x_j sont générés par des locuteurs différents

Le rapport de vraisemblance correspondant à ce test d'hypothèses est calculé à l'aide du modèle multi-Gaussien dans lequel les paramètres inconnus sont remplacés par leurs estimées par maximum de vraisemblance. Si $L(x_i; \mu_i; \Sigma_i)$ désigne la vraisemblance de la séquence x_i et $L(x_j; \mu_j; \Sigma_j)$ la vraisemblance de la séquence x_j, alors la vraisemblance que les deux segments aient été générés par des locuteurs différents $L_1(i, j)$ est donnée par : $L_1(i, j) = L(x_i; \mu_i; \Sigma_i) \times L(x_j; \mu_j; \Sigma_j)$. La vraisemblance $L_0(i, j)$ que les deux segments aient été générés par le même locuteur est : $L_0(i, j) = L(x_{i+j}, \mu_{i+j}; \Sigma_{i+j})$ où x_{i+j} est la réunion des segments x_i et x_j. $\overline{\mu}_{i+j}$ et $\overline{\Sigma}_{i+j}$ sont respectivement la moyenne et la matrice de covariance de la réunion des deux segments. Si $R(i, j)$ désigne le rapport de vraisemblance alors il est égal à :

$$\begin{aligned} R(i,j) &= \frac{L_0(i,j)}{L_1(i,j)} \\ &= \frac{L(x_{i+j}, \overline{\mu}_{i+j}; \overline{\Sigma}_{i+j})}{L(x; \overline{\mu}_i; \overline{\Sigma}_i) \times L(y; \overline{\mu}_j; \overline{\Sigma}_j)} \end{aligned} \tag{2.2}$$

où les lettres barrées désignent les estimées par maximum de vraisemblance.

La distance est finalement obtenue en prenant l'opposé du logarithme de ce rapport de vraisemblance :

$$d(i, j) = -\log R(i, j) \tag{2.3}$$

Cette procédure est efficace pour séparer contrôleur et pilotes dans le cas d'un seul contrôleur. Si au contraire, plusieurs contrôleurs interviennent dans l'intervalle considéré alors cette méthode de regroupement ne permet pas de distinguer les différents contrôleurs.

3. **Initialisation et entraînement du modèle** Pour chaque intervalle I_m, un modèle de mixtures de Gaussiennes à deux états est construit : un état pour le(s) pilote(s) et un état pour le contrôleur. Chaque état est initialisé avec chacun des groupes de segments trouvés précédemment (séparation pilote(s)/contrôleur) pour l'intervalle I_m. Ensuite, les données de parole des intervalles I_m et I_{m-1} servent à entraîner le modèle de l'intervalle I_m.

4. **Reconnaissance** : une fois les M modèles estimés, les segments de l'intervalle I_m sont assignés au modèle le plus probable de l'intervalle I_m ou I_{m-1}. Il faut ensuite déterminer quelle est la composante correspondant aux pilotes et celle correspondant au contrôleur. Deux méthodes sont envisagées. La première consiste à calculer le déterminant de la matrice de covariance du modèle : les pilotes parlant en milieu bruité, la dynamique de leurs paroles est plus faible. Le déterminant est donc plus faible. La seconde méthode consiste à examiner la durée des segments : les contrôleurs ont en général des transmissions plus longues donc des segments plus longs que les pilotes.

2.1.5 Séparation des contrôleurs

Une fois les pilotes et les contrôleurs séparés, le but est de rattacher à chaque contrôleur présent dans l'enregistrement les modèles qui lui correspondent. Pour ce faire, le nombre des contrôleurs présents est supposé connu et un regroupement (*clustering*) basé sur une matrice de distances entre modèles de contrôleurs est opéré (les auteurs ne fournissent pas plus de détails sur la manière dont le regroupement est réalisé et sur la distance entre deux modèles). La distance utilisée est la distance de Kullback-Leibler. Cependant, le principe doit être analogue au regroupement hiérarchique décrit au paragraphe 2.1.4. Cette distance est présentée dans [Siegler et al. 97] ou dans ce document, au paragraphe 3.2.1.

2.1.6 Amélioration du processus

Ce processus peut être amélioré en utilisant la connaissance acquise au cours de celui-ci pour raffiner les segmentations initiales et éviter par exemple de regrouper des segments appartenant à des locuteurs différents.

Une deuxième alternative consiste à reprendre les modèles issus du premier passage et d'effectuer les étapes suivantes :

1. initialiser des modèles de mixtures de bruit, pilotes et contrôleurs avec la segmentation obtenue lors du premier passage

2. entraîner ces modèles avec l'algorithme EM (Expectation-Maximisation)

3. utiliser les composantes du modèles pour classifier les différents segments

2.1.7 Résultats expérimentaux

Les expériences sont menées sur 7 conversations d'une heure environ. Le rapport signal sur bruit est entre 10 et 15 dB. La durée moyenne d'une phrase est de 3 secondes et il y a de un à trois contrôleurs dans chaque conversation.

Connaissant la segmentation de référence, une erreur est détectée quand la fin ou le début des segments ne correspondent pas aux marques de référence. Avec cette mesure, les segments générés sont corrects à 84.4%.

Pour évaluer les performances des modèles de locuteurs, la pureté des modèles est calculée. La pureté d'un modèle est définie comme le pourcentage des paroles provenant du groupe de locuteurs (pilotes ou contrôleur) majoritaire reconnues par ledit modèle. Sur 7 conversations, la pureté des modèles a été évaluée à 94.5%. La précision de l'identification de locuteur est de 94% pour les paroles correctement segmentées.

Une fois les modèles obtenus à l'issue du premier passage, ils peuvent servir pour améliorer la segmentation bruit/parole. En particulier, ils peuvent permettre de distinguer deux locuteurs quand ceux-ci ne sont pas séparés par du bruit donc réunis dans le même segment. 40% des segments contenant deux locuteurs devraient pouvoir être corrigés.

2.2 Analyse

Les travaux de BBN que nous venons de présenter ont le mérite d'être les premiers travaux existant dans le domaine de l'indexation par locuteurs et amènent à des résultats tout à fait honorables. Ils mettent également en évidence les étapes incontournables pour un système d'indexation par locuteurs sans connaissance a priori sur les locuteurs présents dans la conversation. Ces étapes sont les suivantes : la séparation des différents segments de locuteurs et éventuellement des segments de Bruit/Silence/Musique, le regroupement des segments appartenant à un même locuteur, la construction des modèles de locuteurs et enfin, la reconnaissance à partir de ces modèles construits en ligne.

Cependant, même si les étapes du système de BBN se généralisent à n'importe quel système d'indexation par locuteurs, les procédés utilisés sont parfois spécifiques au signal traité : des transmissions radio pilotes-contrôleurs. Il est donc difficile de les appliquer à d'autres types de signaux. Dans ce qui suit, nous mettons en évidence pour chacune des étapes du système d'indexation présenté les techniques qui peuvent se généraliser à tout type de signaux - (et en adéquation avec nos hypothèses de travail) - et, à l'opposé, les techniques qui s'appuient sur la nature particulière des transmissions radio ou qui vont à l'encontre de nos hypothèses de travail.

Segmentation acoustique (2.1.1), séparation Parole/Bruit (2.1.2) et regroupement en segments longs (2.1.3) : Ces trois étapes visent à obtenir les segments ne contenant que de la parole appartenant à un seul locuteur et les plus longs possible. Au cours de ces étapes, les auteurs ne décrivent pas la nature du signal traité mais nous pouvons deviner qu'il existe de longs silence inter-locuteurs lors des transmissions pilotes/contrôleurs. De plus, les canaux de transmission n'étant pas identiques, les auteurs signalent eux-mêmes que la tâche d'obtenir des segments ne contenant qu'un seul locuteur s'en trouve largement facilitée. Aussi, le découpage arbitraire en segments de 20 trames lors de la segmentation acoustique ou le regroupement des segments adjacents ayant la même étiquette Parole ou Silence lors du regroument en segments longs ne sont pas des méthodes généralisables à tout type de signaux. La construction du modèle de silence suppose également d'avoir suffisamment de données de silence et aussi de connaître les probabilités à priori d'avoir de la parole et du silence (cf équation 2.1b).

Par contre, la mesure de variabilité de l'énergie pour distinguer les segments de Parole et de Silence semble une mesure intéressante, même si elle ne semble pas suffisante pour classifier tous les segments.

Construction des modèles de locuteurs (2.1.4) : cette étape débute avec un découpage arbitraire en intervalles de 50 segments de manière à n'obtenir qu'un seul contrôleur dans chaque intervalle. Ce découpage en intervalles est typiquement dépendant du signal traité et, par conséquent, ne peut être repris dans un autre système d'indexation par locuteurs. De plus, les pilotes sont considérés comme un seul et même locuteur. Par conséquent, le problème se résume à distinguer deux locuteurs, ou plus précisément deux catégories de locuteurs, dans chaque intervalle. Cela permet de stopper le regroupement hiérarchique quand il n'y a plus que deux groupes de segments, chacun des groupes de segments étant relatif à une catégorie de locuteurs. Cela permet de construire un modèle à deux états pour séparer les deux catégories de locuteurs, etc.... Les techniques utilisées lors de cette étape peuvent se généraliser à plusieurs locuteurs. Néanmoins, elles nécessitent la connaissance du nombre de locuteurs. C'est en effet ce nombre qui fixe le critère d'arrêt de l'algorithme de regroupement hiérarchique et qui fixe le nombre d'états ou de Gaussiennes du *GMM*. Ceci ne correspond donc pas à nos hypothèses de travail.

Par contre, si nous arrivons à déterminer le nombre de locuteurs présents dans le document audio, ces techniques sont applicables. Par ailleurs, la distance utilisée pour séparer les différents locuteurs (cf équation 2.3) est une distance tout à fait intéressante et nous l'utilisons par la suite (cf chapitre 4).

Séparation des contrôleurs (2.1.5) : au cours de cette étape, le nombre de contrôleurs est supposé connu. Or, une des hypothèses était que le nombre de locuteurs était inconnu. Cette hypothèse, que nous faisons également pour notre système d'indexation, n'est donc pas respectée.

Amélioration du processus (2.1.6) : le principe de cette étape est à retenir. La connaissance apprise lors de la première passe est utilisée lors d'une seconde passe pour raffiner l'indexation et ainsi de suite pour les passes suivantes. Comme nous ne nous fixons pas de contraintes de temps réel, nous pouvons exploiter ce principe pour raffiner successivement le résultat de l'indexation, jusqu'à obtenir un résultat relativement stable.

2.3 Architecture de notre système d'indexation par locuteurs

Ayant mis en évidence les points forts et les points faibles (ou en tout cas inapplicables étant donné notre contexte) du système proposé par BBN, nous présentons l'architecture de notre propre système d'indexation. Même si certaines méthodes utilisées dans le système proposé par BBN sont spécifiques au signal traité, la suite des tâches est en revanche valable pour tout système d'indexation par locuteurs, sans connaissance a priori.

Notre système d'indexation s'inspire fortement de celui proposé par BBN. Nous distinguons quatre grandes étapes que nous détaillons ci-après. La succession de ces étapes est illustrée à la figure 2.1.

1. La **segmentation en locuteurs** constitue la première étape : la bande sonore est découpée en segments de parole ne contenant qu'un seul locuteur et ces segments doivent être aussi longs que possible.

2. La deuxième étape consiste à **regrouper les segments de parole appartenant à un même locuteur** pour avoir suffisamment de données d'un même locuteur pour construire un modèle fiable.

3. La **modélisation des locuteurs** à partir de chacun des groupes de segments obtenus à l'étape précédente constitue la troisième étape. A l'issue du regroupement par locuteurs, le volume de données par locuteur est plus important que le volume de données contenu dans chaque segment pris individuellement. Cela permet donc de construire des modèles plus sophistiqués et plus fiables.

4. Nous utilisons ces modèles de locuteurs construits en ligne pour l'étape ultime de **reconnaissance de la séquence de locuteurs** engagés dans la conversation. A ce stade,

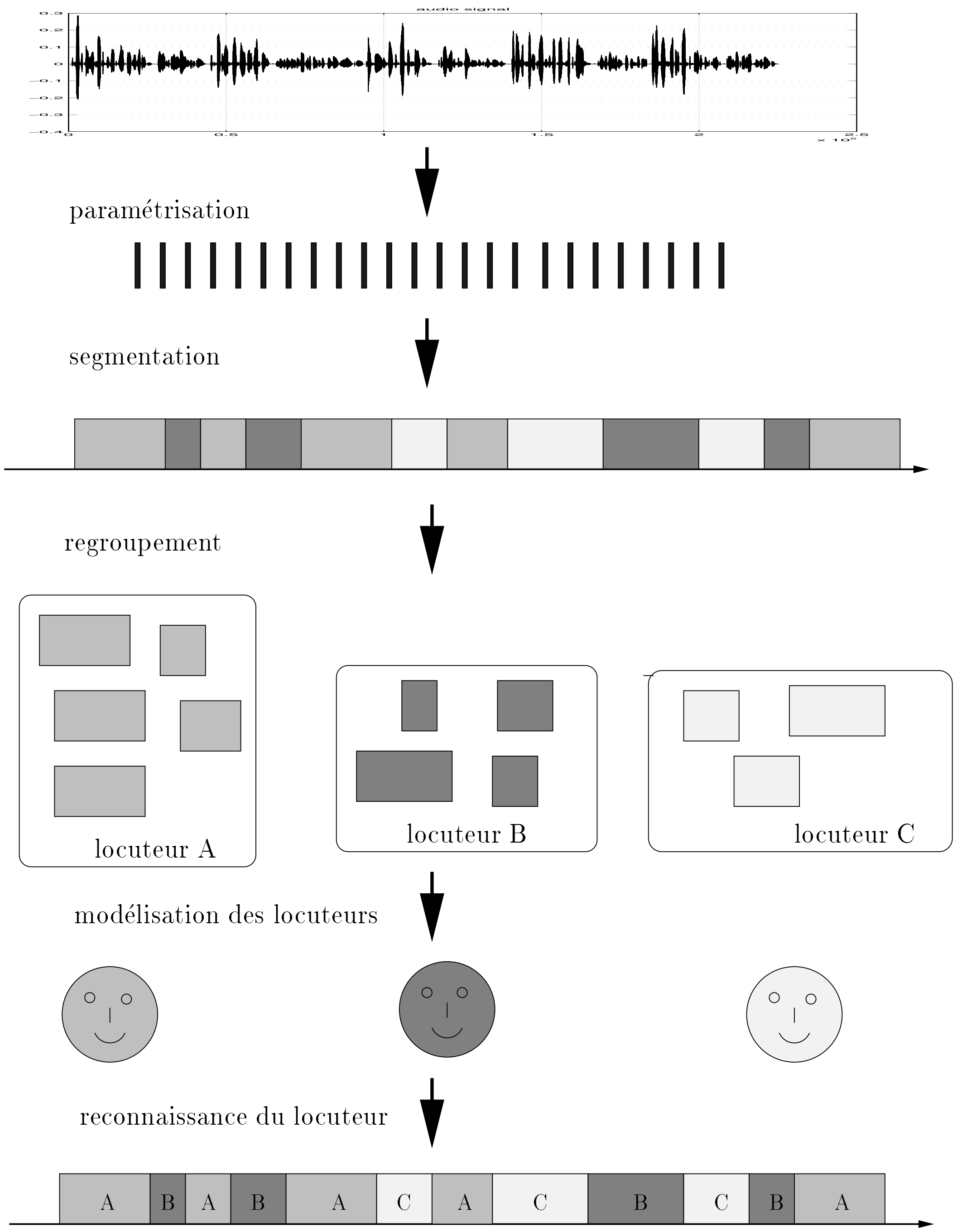

FIG. 2.1 – *Architecture du système d'indexation par locuteurs*

nous opérons également un raffinement de la segmentation obtenue lors de la première étape.

Dans la suite de ce document, nous nous intéressons plus particulièrement aux deux premières étapes. La segmentation en locuteurs est abordée à la partie I de ce manuel. Le regroupement des segments par locuteur fait l'objet de la partie II.

Première partie

Segmentation en Locuteurs

Introduction

Dans cette partie, nous abordons le problème de la segmentation en locuteurs qui est une des étapes de notre système d'indexation par locuteurs. **La segmentation en locuteurs d'un document sonore consiste à découper le flux audio en segments homogènes les plus longs possibles et ne contenant les paroles que d'un seul locuteur**. A l'issue de la segmentation, les segments ne sont pas encore étiquetés ; c'est-à-dire que le locuteur qui a prononcé les paroles contenues dans un segment n'est pas encore identifié.

Nous rappelons les hypothèses que nous nous sommes fixées dans le cadre du système d'indexation par locuteurs et qui importent pour la segmentation :

- **Nous n'avons à notre disposition ni modèle de locuteur, ni modèle de langage, ni modèle d'aucune sorte. Les seules informations disponibles sont contenues dans le document sonore considéré.**
- **Les personnes ne parlent pas simultanément**

Nous distinguons trois types de techniques de segmentation :

1. Le premier type de segmentation qui vient à l'idée, le plus simple, est la **segmentation par détection de silences**. Elle repose sur l'hypothèse implicite que les différents locuteurs sont séparés par des silences significatifs. Un silence est significatif si il a une durée suffisante pour être détectable et si ce silence n'est pas trop bruité. Selon le type de documents audio à traiter, cette hypothèse peut être jugée irréaliste. Par exemple, dans une conversation téléphonique, il est rare que les paroles des locuteurs soient séparées par des silences significatifs.

2. Le deuxième type de segmentation est la **segmentation par détection de changements de locuteurs**, i.e. des ruptures dans le signal audio. Le principe de ce type de segmentation est le suivant : deux portions de signal sont considérées et une distance, ou plus généralement une mesure de similarité, est calculée entre ces deux portions. Selon la valeur de cette mesure, soit les deux portions sont proclamées comme ayant été prononcées par un même locuteur, soit un changement de locuteur est détecté à la frontière commune aux deux portions, comme décrit dans le système d'indexation proposé par BBN (2.1). Le problème de ce type de segmentation réside d'une part dans le choix de la distance et du seuil de décision et d'autre part, dans la manière de considérer les portions de signal (les portions sont-elles consécutives, se recouvrent-elles, comment sont-elles décalées entre deux itérations, etc...)

3. Le troisième type de segmentation est la **segmentation par identification de trames acoustiques** qui nécessite l'emploi de modèles. Le principe consiste à identifier un groupe de trames acoustiques avec le modèle qui lui correspond le mieux. Ces modèles peuvent être de différentes natures selon les applications : modèles de locuteurs, modèles de genre, modèle de bruit/parole/musique, etc.... Ce type de segmentation sortant du cadre de notre étude (nous supposons en effet qu'aucun modèle d'aucune sorte n'est à notre disposition), nous renvoyons le lecteur aux références bibliographiques pour plus de détails.

Ce sont surtout les deux premiers types de segmentation qui vont nous importer pour la suite. Cependant, nous reviendrons sur le dernier type à la partie III de ce document.

Nous nous intéressons tout d'abord au chapitre 3 aux techniques de segmentation existant dans la littérature. Au chapitre 4, nous présentons les méthodes de segmentation que nous proposons : SILHYST et DISTBIC. Le chapitre 5 est consacré aux méthodes d'évaluation des algorithmes de segmentation et aux expériences menées. Enfin, nous tirons les conclusions qui s'imposent à l'issue de ces expériences.

Chapitre 3

Etat de l'Art

A l'exception des travaux de H.Gish et al., que nous mentionnons au chapitre 2 et qui sont des travaux précurseurs dans ce domaine, la segmentation en locuteurs n'a attiré l'intérêt des chercheurs que très récemment. La majorité des travaux que nous détaillons dans ce chapitre ont été réalisés dans le cadre des évaluations DARPA des systèmes de transcription automatique de journaux radio-diffusés (*http://www.itl.nist.gov/div894/894.01/proc/darpa97/, /darpa98/ et /darpa99/*). Ces évaluations se font sur la bases de données HUB-4 qui se compose de 200 heures de nouvelles des chaînes télévisées ABC, CNN, et CSPAN et de la radio PRI. Cette base de données est décrite sur le site Web du *Linguistic Data Consortium* (*http://www.ldc.upenn.edu/ldc/about/broadcast97.html*).

Dans la suite de ce chapitre, nous nous concentrons essentiellement sur les deux premiers types de segmentation présentés dans l'introduction, à savoir la segmentation par détection de silences et la segmentation par détection de changements de locuteurs.

3.1 Segmentation par détection de silences

Un silence (s'il n'est pas trop bruité) étant caractérisée par un faible niveau d'énergie, la détection de silences repose en général sur le calcul de l'énergie du signal (paramétrisé ou non). Nous allons voir quelques approches possibles.

3.1.1 Utilisation de la puissance moyenne

La première approche de détection de silences repose sur le calcul de la puissance moyenne. Le niveau d'énergie d'un silence étant faible, il en est de même de la puissance moyenne. Le problème est alors de définir un seuil sur la puissance moyenne pour séparer Silence et Parole. En effet, ce seuil peut être très variable d'un document audio à l'autre et même au sein d'un même document.

Cette approche est utilisée par exemple par [Nishida et al. 98, Nishida et al. 99] pour découper le flux audio issu d'un journal télévisé en zones de paroles séparées par du silence. Ils calculent la puissance moyenne toutes les secondes et si la valeur de celle-ci est inférieure à un certain seuil alors un silence est détecté.

Les auteurs ne précisent pas la manière dont ils choisissent le seuil. Un seuil absolu reste toujour sun choix très dépendant de l'application. Par ailleurs, aucune évaluation de cette technique de détection de silences n'est fournie.

3.1.2 Utilisation de l'histogramme de l'énergie

Pour séparer le Silence/Non Silence (Parole/Musique/Bruit), l'histogramme de l'énergie à court-terme est construit sur des segments de 15 secondes. Pour chaque segment, la moyenne μ et la déviation standard σ des énergies calculées sur des trames de 10 ms sont évaluées. Si l'histogramme est approximé par une Gaussienne alors 95% de ces énergies sont situées entre $\mu - 2\sigma$ et $\mu + 2\sigma$. Le segment est alors homogène et est étiqueté Silence ou Non Silence. Si le segment n'est pas homogène, alors les moyennes et déviations standards du Silence/Non Silence sont recherchées par l'algorithme de regroupement *K-Means* (cf [Rabiner et al. 78]). Un seuil en est déduit et sert dans un automate pour une ultime segmentation.

Cette approche est utilisée par [Montacie et al. 98]. Ils testent cette méthode sur le film "Conte de printemps" d'Eric Rohmer. Aucune erreur n'est trouvée. Il faut cependant noter que le style du réalisateur (dialogues largement entrecoupés de silences, long monologues, quasi-absence de bande originale, plans longs, etc...) en fait un film qui présente des avantages certains pour une telle segmentation.

3.1.3 Utilisation de la variabilité de l'énergie

Cette approche a été détaillée au chapitre 2. Elle consiste à calculer la variabilité de l'énergie pour une portion de signal. Si cette variabilité est faible alors la portion de signal est un silence, sinon c'est de la parole. Cette technique nécessite également le choix d'un seuil, qui dépend également du type de signal audio. D'ailleurs, seuls les segments de parole et de silence obtenant un haut niveau de confiance avec cette mesure, sont conservés par [Gish et al. 91].

3.1.4 Utilisation du taux de passages par zéro

Un silence (avec ou sans bruit de fond) est caractérisé, outre par son faible niveau d'énergie, par un taux de passages par zéro (*zero-crossing rate*) élevé. Le taux de passages par zéro représente le nombre de fois que le signal a une amplitude nulle par unité de temps. Comme les approches précédentes, il s'agit de déterminer un seuil au-delà duquel le taux de passages par zéro sera significatif d'un silence.

[Tsekeridou et al. 99] utilisent cette technique couplée à une méthode de détection de début et de fin de phrase. Ils déterminent le seuil à partir d'un signal audio connu a priori. D'une part, ceci est contraire à nos hypothèses et d'aute part, cela nécessite d'apprendre le seuil pour chaque type de document audio.

Les approches par détection de silences font toutes appel à un seuil qui dépend du document audio considéré et il n'existe aucune méthode a priori pour déterminer de manière optimale ce seuil. Au chapitre 4, nous proposons une méthode de détection de silences SIL-HYST qui s'affranchit de ce problème.

3.2 Segmentation par détection de changements de locuteurs

Les techniques détaillées dans cette section ont été utilisées pour la plupart dans le cadre de la transcription automatique de nouvelles radio-diffusées ([Bakis et al. 97, Chen et al. 98a]).

Comme vu précédemment, les taux de reconnaissance de parole sont améliorés quand les modèles de parole sont adaptés au locuteur ou plus généralement aux conditions acoustiques. Une étape préliminaire consiste donc à segmenter le flux audio en segments homogènes en termes de locuteurs ou en termes de conditions acoustiques.

3.2.1 Utilisation de la distance de Kullbach-Leibler

La distance de Kullback-Leibler (KL) (ou entropie croisée relative) mesure la distance entre les distributions de probabilité des deux variables. Cette distance est donnée par :

$$\begin{aligned} KL(X,Y) &= \int_{-\infty}^{+\infty} p_X(X) \log \frac{P_X(X)}{P_Y(X)} dX && (3.1a) \\ &= E_X \left(\log P_X(X) - \log P_Y(X) \right) && (3.1b) \end{aligned}$$

où $E_X()$ désigne l'espérance calculée avec la distribution de probabilité P de X.

Pour symétriser cette mesure, la mesure $KL2$ est définie par :

$$KL2(X,Y) = KL(X,Y) + KL(Y,X) \tag{3.2}$$

Quand les variables aléatoires X et Y ont des distributions Gaussiennes, la mesure $KL2$ devient (cf annexe B) :

$$KL2(X,Y) = \frac{1}{2}\left(\frac{\sigma_X^2}{\sigma_Y^2} + \frac{\sigma_Y^2}{\sigma_X^2}\right) + \frac{1}{2}(\mu_X - \mu_Y)^2 \left(\frac{1}{\sigma_X^2} + \frac{1}{\sigma_Y^2}\right) - 1 \tag{3.3}$$

Le principe de la segmentation est le suivant. Pour chaque point du flux audio, deux portions de signal adjacentes sont considérées de part et d'autre du point et d'une durée de 2 secondes chacune. Pour chaque portion, la moyenne et la covariance sont estimées, ce qui permet de déduire la distance $KL2$ entre les deux fenêtres. Ce processus est répété en chaque point du flux audio et quand la distance $KL2$ atteint un maximum local entre deux fenêtres adjacentes, une nouvelle limite de segment est détectée.

[Siegler et al. 97] utilisent la distance de Kullback-Leibler pour segmenter le flux audio en classes acoustiques homogènes. Les classes acoustiques qu'ils distinguent sont les suivantes : parole préparée (proche de la parole lue), parole spontanée, parole de moyenne qualité (qualité téléphonique par exemple), parole en présence d'un fond musical, parole en présence d'un bruit de fond, parole de personnes non natives, autres types de paroles.

Pour tester l'efficacité de la segmentation, les auteurs ont comparé les segments de la segmentation de référence (faite à la main et fournie dans le cadre des évaluations DARPA) d'une durée supérieure à 2 secondes avec les segments trouvés par l'algorithme précédent. 64% des segments ont été détectés. Par contre, les auteurs ne précisent pas le nombre de fausses alarmes, c'est-à-dire le nombre de limites de segments qui ont été trouvées et qui n'existent pas. De plus, la segmentation de référence est une segmentation en classes acoustiques et non en locuteurs. Enfin, rien n'est précisé sur la détection des maxima locaux des valeurs de distance.

3.2.2 Utilisation d'une distance discriminante

Ce paragraphe présente un algorithme de segmentation basé sur une mesure de distance entre deux fenêtres adjacentes. Ce couple de fenêtres est déplacé itérativement le long du signal paramétrisé de parole. Ces fenêtres sont d'une durée de 1 seconde chacune. Le calcul de la distance discriminante repose sur le principe suivant. Les vecteurs acoustiques contenus dans chaque fenêtre sont répartis en trois classes à l'aide de l'algorithme *K-Means*. La première classe correspond au silence : les vecteurs acoustiques qui la composent sont similaires entre deux fenêtres adjacentes. La deuxième classe correspond aux vecteurs acoustiques associés à la parole communs à tous les locuteurs. Enfin, la dernière classe contient les vecteurs acoustiques relatifs à la parole mais cette fois-ci, spécifiques à chaque locuteur. Chaque classe est ensuite modélisée par une distribution Gaussienne multi-dimensionnelle et est caractérisée par son vecteur moyen, sa matrice de covariance (supposée diagonale) et le nombre de vecteurs qui la composent.

Trois distances sont ensuite calculées respectivement entre les deux classes les plus proches, puis entre les deux plus proches suivantes, et enfin, entre les deux plus éloignées. L'idée est que la plus grande des distances représente la distance entre les caractéristiques acoustiques spécifiques au locuteur. Cette distance est normalisée par la plus petite des distances représentative des segments de silence. Une autre distance d est calculée en considérant une seule Gaussienne par fenêtre. Cette différence correspond à la différence totale entre les deux fenêtres. Enfin, la distance discriminative D est calculée de la manière suivante :

$$D^i = \frac{\mu d_3^i / d_1^i}{d^i} \tag{3.4}$$

où d_3^i représente la plus grande distance, d_1^i la plus petite et d^i la distance totale pour la $i$ème paire de fenêtres adjacentes. μ est la valeur moyenne des d^i calculées sur toutes les paires de fenêtres adjacentes.

Ce calcul de D^i est répété pour chaque paire de fenêtres adjacentes et une courbe de distances est obtenue à l'issue du processus. Les changements acoustiques correspondent à de fortes valeurs de D. Pour détecter ces points de changement, la courbe de distances D est écrêtée à un seuil fixe α, c'est-à-dire que seules les portions de la courbe situées au-dessus de la droite d'abscisse α sont conservées. Pour chaque portion, un maximum est recherché. Si deux maxima sont proches l'un de l'autre, c'est-à-dire séparés par un intervalle de temps très court, alors seul le maximum des deux est conservé.

[Beigi et al. 98] utilisent cette méthode pour segmenter un flux audio dans le cadre des évaluations DARPA. Les données utilisées pour tester cet algorithme contiennent à la fois des segments de parole, des segments de musique, de silence, etc... Une erreur de 30% est trouvée pour la détection correcte d'un point de changement entre deux locuteurs adjacents. Selon eux, ce taux élevé d'erreurs est dû à la forte variation d'échelle lors de la détection. En effet, un changement entre un segment de musique et un segment de parole est beaucoup plus flagrant en terme de distance D qu'un changement entre deux locuteurs consécutifs. Ces fortes variations de distances au sein d'un même document audio montrent l'inadéquation d'un seuil fixe pour la détection des changements de locuteurs. De plus, les auteurs ne précisent pas comment leur seuil α est déterminé et s'il varie en fonction des enregistrements.

Nous pouvons aussi nous interroger sur la validité en terme de fiabilité de l'algorithme *K-Means* sur 1 seconde de signal. Est-ce que les classes obtenues sont vraiment représentatives du signal contenu dans la fenêtre ?

3.2.3 Utilisation du Critère d'Information Bayesien (BIC)

Le flux audio peut être modélisé comme un processus Gaussien dans l'espace cepstral. L'approche utilisée se base sur le maximum de vraisemblance et la décision d'un changement de classe acoustique repose sur le Critère d'Information Bayesien BIC (Bayesian Information Criterion). Ce critère est également connu sous le nom de MDL (Minimum Description Length) ou encore critère de Rissanen [Rissanen 89, Hayes 96].

Critère de sélection des modèles

Le critère BIC est un critère de vraisemblance pénalisé par la complexité du modèle, i.e. le nombre de paramètres du modèle. Soit $X = \{x_1, ..., x_{N_X}\}$ les données à modéliser et M le modèle paramétrique envisagé. La fonction de vraisemblance $L(X, M)$ est maximisée pour le modèle. m désigne le nombre de paramètres du modèle. Le critère BIC pour le modèle M est défini par :

$$\mathrm{BIC}(M) = \log L(X, M) - \lambda \frac{m}{2} \log N_X \tag{3.5}$$

Le premier terme reflète l'ajustement du modèle aux données et le deuxième terme correspond à la complexité du modèle. λ est un poids de pénalité, en théorie égal à 1.

Le critère BIC permet de sélectionner un modèle parmi plusieurs modèles pour les mêmes données : c'est le modèle qui maximise ce critère, donc le modèle qui correspond le plus aux données en terme de vraisemblance et dont la complexité reste raisonnable.

Détection d'un changement de locuteur à l'aide du critère BIC

Soit $X = \{x_1, ..., x_{N_X}\}$ une séquence de N_X vecteurs cepstraux. Soit le test d'hypothèses suivant pour un changement à l'instant i :

- H_0 : la séquence a été prononcée par un seul et même locuteur. Alors la séquence est supposée être générée par un seul processus Gaussien multi-dimensionnel :

$$(x_1, ..., x_{N_X}) \sim N(\mu_X, \Sigma_X)$$

- H_1 : la séquence a été prononcée par deux locuteurs différents. Alors les deux sous-séquences correspondant à chacun des locuteurs sont générées par des processus Gaussiens multi-dimensionnels différents :

$$(x_1, ..., x_i) \sim N(\mu_{X_1}, \Sigma_{X_1}) \text{ et } (x_{i+1}, ..., x_{N_X}) \sim N(\mu_{X_2}, \Sigma_{X_2})$$

où μ représente la moyenne et Σ représente la matrice de covariance supposée pleine.

Le rapport de maximum de vraisemblance entre l'hypothèse H_0 et l'hypothèse H_1 est donné par (nous démontrons cette formule en Annexe C):

$$R(i) = \frac{N_X}{2} \log |\Sigma_X| - \frac{N_{X_1}}{2} \log |\Sigma_{X_1}| - \frac{N_{X_2}}{2} \log |\Sigma_{X_2}| \tag{3.6}$$

où Σ_X, Σ_{X_1}, et Σ_{X_2} sont les matrices de covariance respectivement de toutes les données, de $\{x_1 ... x_i\}$ et de $\{x_{i+1} ... x_{N_X}\}$ et N_X, N_{X_1}, et N_{X_2}, sont respectivement le nombre de vecteurs acoustiques dans la séquence complète, de la sous-séquence $\{x_1, ..., x_i\}$,et de la sous-séquence $\{x_{i+1}, ..., x_{N_X}\}$.

Alors l'estimée par maximum de vraisemblance du point de changement est donnée par :

$$\widehat{t} = \arg\max_i R(i) \tag{3.7}$$

Ce test d'hypothèses peut être aussi vu comme la comparaison de deux modèles : un modèle de données avec deux Gaussiennes (hypothèse H_1) et un modèle de données avec une seule Gaussienne (hypothèse H_0). La différence entre les valeurs BIC de ces deux modèles s'écrit :

$$\Delta\mathrm{BIC}(i) = -R(i) + \lambda P \tag{3.8}$$

où le rapport de vraisemblance $R(i)$ est celui défini à l'équation (3.6) et la complexité est donnée par :

$$P = \frac{1}{2}(p + \frac{1}{2}p(p+1)) \log N_X \tag{3.9}$$

p est la dimension de l'espace des vecteurs acoustiques. En effet, il y a p paramètres pour estimer le vecteur moyen et $\frac{p(p+1)}{2}$ paramètres à estimer pour la matrice de covariance qui est symétrique. Le poids de pénalité λ est en théorie égal à 1 (cf [Rissanen 89]).

Ainsi si l'équation (3.8) est négative, alors le modèle à deux Gaussiennes est privilégié. Cette modélisation correspond à l'hypothèse H_1 qui suppose que la séquence a été prononcée par deux locuteurs différents. En résume, il y a un changement si :

$$\{\min_i \Delta\mathrm{BIC}(i)\} > 0 \tag{3.10}$$

L'estimée par maximum de vraisemblance du point de changement peut alors aussi s'exprimer par :

$$\widehat{t} = \arg\min_i \Delta\mathrm{BIC}(i) \tag{3.11}$$

[Chen et al. 98c] utilisent ce critère BIC pour segmenter les données fournies pour les évaluations DARPA. Selon eux, la procédure BIC a l'avantage ne pas faire intervenir de seuil, contrairement aux méthodes de segmentation précédentes. En effet, la détermination du seuil optimal est en général complexe et le choix de ce seuil est en général passé sous silence par les différents auteurs. Le critère BIC peut être vu comme le seuillage de la distance par log-vraisemblance avec un seuil automatiquement fixé à $\frac{1}{2}\lambda(p + \frac{1}{2}p(p+1)) \log N_X$ où N_X est la taille de la fenêtre de décision et p la dimension de l'espace de paramètres.

Cependant, [Chen et al. 98c] oublient le poids de pénalité λ qui en pratique n'est pas forcément égal à 1 (cf chapitre 5).

Malgré cela, la méthode de segmentation par critère BIC présente l'avantage d'avoir un seuil relativement systématique, quel que soit le type de données. Aussi, nous en détaillons l'implémentation décrite dans [Tritschler 98] pour la détection de plusieurs changements de locuteurs. En effet, l'une des étapes de cette implémentation va servir dans la méthode de segmentation que nous proposons (cf chapitre 4).

Détection de plusieurs changements à l'aide du critère BIC

L'implémentation du critère BIC se déroule en trois étapes :

1. Une première passe est opérée pour localiser grossièrement les changements de locuteurs potentiels. La valeur de ΔBIC est calculée entre deux fenêtres adjacentes $[a, b]$ et $[b, c]$, où

les bornes a et c sont fixes. La borne b est située entre a et c et sa position est incrémentée à chaque itération d'un certain pas. La fenêtre $[a, c]$ est augmentée tant qu'aucune valeur négative de ΔBIC n'a été trouvée. Par contre, quand une valeur négative est trouvée, donc un changement de locuteur détecté, ce point de changement devient la nouvelle borne a.

2. La deuxième passe s'appuie sur le même principe pour raffiner la localisation des points de changement de locuteurs trouvés lors de la première passe. L'intervalle $[a, c]$ est choisi plus court et est centré sur le point de changement potentiel. De même, le pas qui sert à incrémenter la position de la borne b est choisi plus petit. Ces deux premières passes sont illustrées à la figure 3.1.

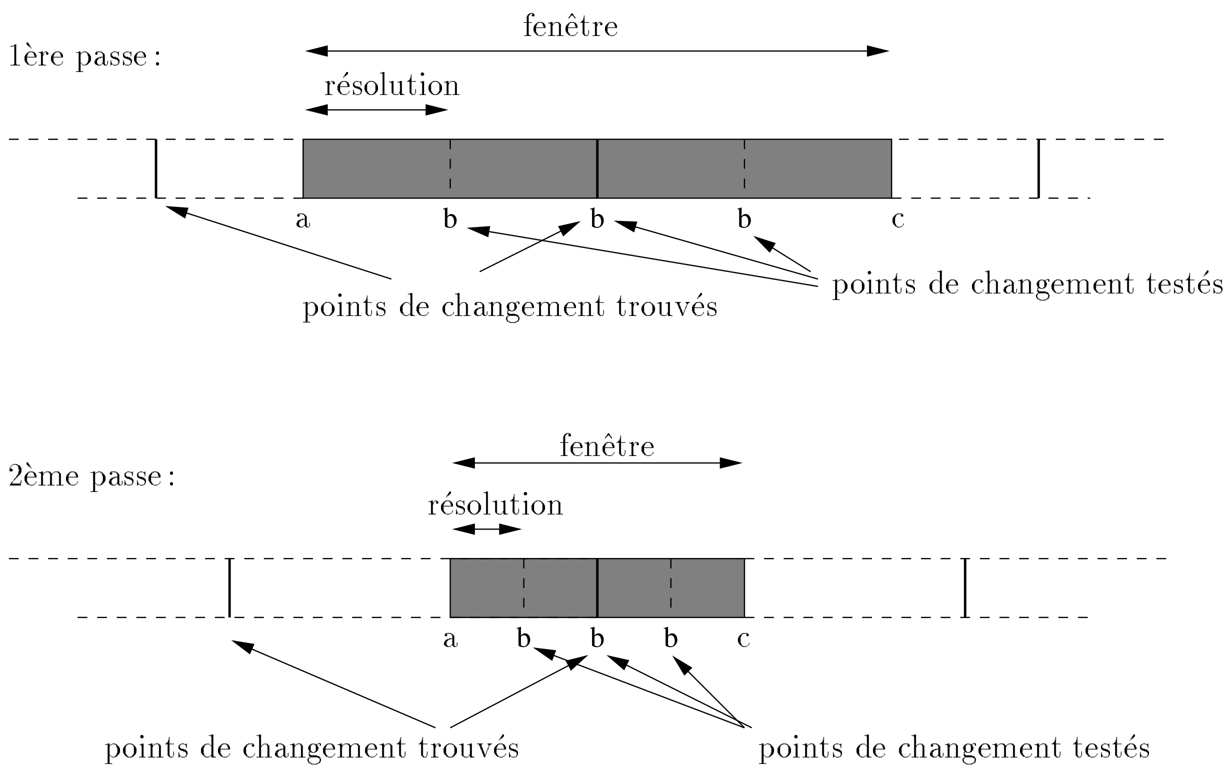

FIG. 3.1 – *Première et deuxième passes de l'implémentation du critère BIC*

3. La troisième passe valide les résultats de la seconde passe. Si $\{s_1, ..., s_N\}$ est l'ensemble des points de changements potentiels résultant de la deuxième passe, une valeur de ΔBIC est calculée pour chaque couple de fenêtres $[s_{i-1}, s_i]$ $[s_i, s_{i+1}]$. Si la valeur est négative, un changement de locuteur est détecté à l'instant i. Sinon, le point s_i est retiré de l'ensemble des points de changements potentiels, de telle sorte que la prochaine valeur de ΔBIC est calculée sur le nouveau couple de fenêtres $[s_{i-1}, s_{i+1}]$ $[s_{i+1}, s_{i+2}$ (avec les anciens indices), comme indiqué à la figure 3.2.

[Chen et al. 98c] distinguent deux types d'erreurs. Ils ont d'abord examiné si les points détectés correspondaient à de vrais points de changements en fonction de la segmentation standard fournie dans le cadre de l'évaluation DARPA. Parmi les 462 changements détectés,

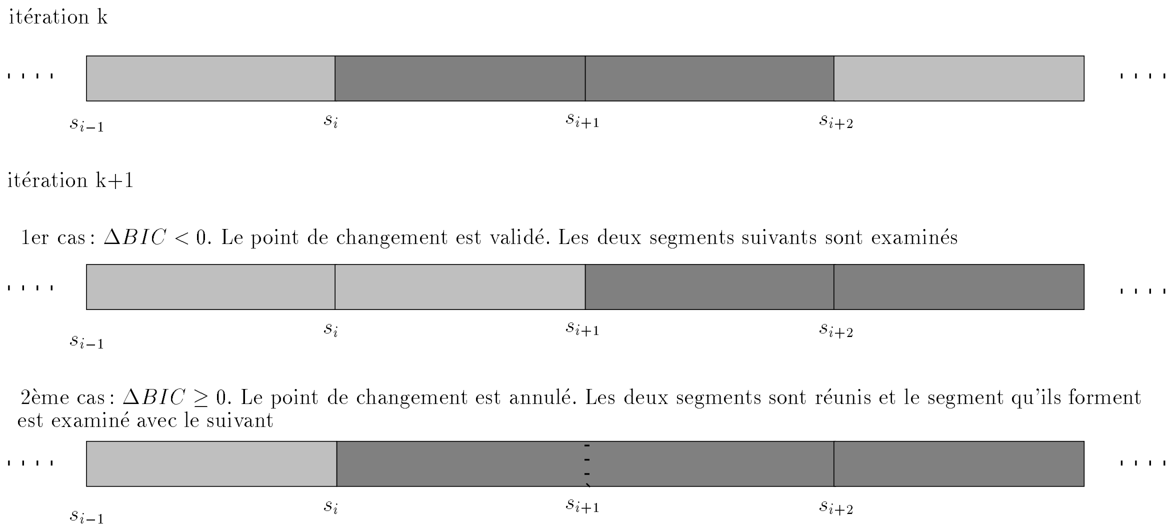

FIG. 3.2 – *Troisième passe de l'implémentation du critère BIC*

4.1% ont lieu au milieu du discours d'un locuteur (pas gênant pour l'indexation), 3.0% ont lieu au sein de segments de musique et 4.3% ont été détectés avec un décalage inférieur à 1 seconde. L'autre type d'erreurs correspond aux changements non détectés par l'algorithme. 33.4% de changements n'ont pas été détectés. 25% d'erreurs sont dues à des changements intervenant à moins de 2 secondes les uns des autres.

3.2.4 Utilisation de modèles vectoriels auto-régressifs et distances associées

Les techniques utilisées dans ce paragraphe pour segmenter un flux audio en Locuteur, Musique ou Bruit sont basées sur des modèles de vecteurs auto-régressifs (Auto-Regressive Vector ARV). Soit $X = \{x_n\}$, $(n = 1, ..., N)$ une succession de N vecteurs cepstraux de dimension p. Leur évolution est décrite par le modèle ARV estimé sur la séquence de vecteurs :

$$x_n = \sum_{i=1}^{q} A_i x_{n-i} + e_n \tag{3.12}$$

où les $\{A_i\}$, $(i = 1, ..., q)$ sont des matrices $p \times p$ et $\{e_n\}$, $(n = 1, ..., N)$ est un bruit blanc vectoriel ayant une matrice de covariance D.

La mesure inter-locuteurs choisie IS_1 ([Montacie et al. 92]) est basée sur la distance d'Itakura [Itakura 75]. Soit $Y = \{y_n\}$, $(n = 1, ..., M)$ une autre séquence de M vecteurs cepstraux de dimension p. Le résiduel de la séquence $\{y_n\}$ filtrée par le modèle $\{A_i\}$ est défini par :

$$e_{YA_n} = y_n - \sum_{i=1}^{q} A_i y_{n-i} \tag{3.13}$$

Soit la matrice de covariance D_{YA} du résiduel e_{YA_n}. Alors la distance IS_1 est définie par :

$$IS_1(X, Y) = Trace(D_{YA}) \tag{3.14}$$

La méthode de segmentation automatique repose sur les modèles ARV. C'est une extension de la méthode Forward-Backward Divergence (FBD) présentée dans [André-Obrecht 88]. Les coefficients utilisés sont des coefficients Mel-cepstraux.

Trois fenêtres glissantes w_0, w_1 et w_2 sont considérées : w_1 et w_2 sont adjacentes et w_0 est la réunion de w_1 et w_2. Trois modèles ARV sont calculés sur ces fenêtres et sont comparés les uns aux autres pour détecter des discontinuités spectrales. La distance IS_I entre les fenêtres w_1 et w_2 est calculée et normalisée par la distance IS_I calculée entre les fenêtres w_1 et w_0. Si ce rapport est supérieur à un certain seuil alors un changement de type de sons (Locuteur, Musique, Bruit..) a lieu.

[Montacie et al. 97] utilisent cette approche pour segmenter des données vidéo en Silence, Bruit, Musique ou Parole, le but ultime étant d'aligner un script sur la vidéo. Les tests ont été réalisés d'une part sur la base de données TIMIT et d'autre part sur des données audio issues du film "Un indien dans la ville". Les données TIMIT ont été obtenues en concaténant 200 phrases prononcées par 200 personnes différentes issues de TIMIT. Pour décider de la qualité de la segmentation obtenue, le taux de reconnaissance des frontières R est calculé. Ce taux est égal au taux d'identification des frontières (Id) moins le taux d'insertion de frontières. Pour la séparation des 200 locuteurs de TIMIT, le taux d'erreur est de 9%. Pour le signal audio du film, 4 types de sons sont considérés : Bruit, Musique, Parole et Silence. La première difficulté réside dans le fait que la bande audio contient peu de segments homogènes (i.e. ne contenant qu'un type de sons) de durée supérieure à 3 s. Or, cette durée (minimale) est nécessaire au bon fonctionnement des techniques de reconnaissance du locuteur. La deuxième difficulté provient de la haute variabilité des types de sons. Par conséquent, le taux d'erreur augmente sensiblement par rapport à la précédente expérience. Les auteurs ne précisent rien quant au choix du seuil.

Parmi les méthodes présentées, l'un d'elles retient toute notre attention : la segmentation par utilisation du critère BIC. Elle présente l'intérêt de fonctionner avec un seuil relativemnt systématique, quel que soit le type de signal audio. Par contre, elle n'est pas en mesure, de par son implémentation, de détecter des points de changements proches les uns des autres. Par ailleurs, les techniques de segmentation basées sur des mesures de distance entre portions de signal peuvent a priori résoudre ce genre de problème, à condition d'en améliorer la détection des changements de locuteurs et de la rendre systématique pour tous les types de signaux audio. C'est ce que nous faisons au chapitre suivant en proposant une méthode de segmentation DISTBIC, qui allie mesure d'une distance et critère BIC.

Chapitre 4

Techniques de segmentation proposées

Dans ce chapitre, nous proposons deux méthodes de segmentation qui répondent aux hypothèses que nous nous sommes fixées (cf chapitre I). SILHYST est une technique de segmentation par détection de silences et est présentée à la section 4.1. DISTBIC est une méthode de segmentation qui combine des algorithmes de détection de changements de locuteurs. Elle est décrite à la section 4.2.

4.1 SILHYST : segmentation par détection de silences

Nous avons détaillé au paragraphe 3.1 quelques techniques de segmentation par détection des silences. La plupart de ces techniques s'appuient sur l'énergie ou sur la puissance du signal ou encore utilisant la variation de ces deux grandeurs. La méthode que nous proposons repose elle aussi sur l'énergie du signal. Le principe est simple : il s'agit de comparer l'énergie locale à l'énergie moyenne du signal.

L'énergie locale E_{loc} est l'énergie de l'échantillon temporel du signal audio s à l'instant t. Elle est calculée comme suit :

$$E_{ech}(t) = [s(t)]^2 \tag{4.1}$$

L'énergie des signaux audio sur lesquels nous travaillons peut être fort variable, soit du fait des conditions d'enregistrement diverses (journaux télévisés), soit du fait de la variabilité du canal de transmission (conversations téléphoniques). Pour cette raison, l'énergie moyenne E_{moy} correspond dans notre cas à une énergie locale moyenne. Si t_{debut} et t_{fin} désignent respectivement les instants de début et de fin de la portion de signal considérée pour calculer l'énergie moyenne locale, nous avons la formule suivante :

$$E_{moy}(t_{debut} \to t_{fin}) = \frac{1}{t_{fin} - t_{debut}} \sum_{t=t_{debut}}^{t_{fin}} [s(t)]^2 \tag{4.2}$$

Afin de réduire la dynamique des valeurs prises par l'énergie, nous en prenons le logarithme.

Pour comparer l'énergie d'un échantillon E_{ech} à l'énergie moyenne E_{moy}, nous définissons deux seuils, un seuil bas α $(0 < \alpha < 1)$ et un seuil haut $\beta > \alpha$ $(0 < \beta < 1)$, et les règles suivantes (ce seuillage s'inspire fortement du seuillage par hystérésis utilisé en traitement d'images [Canny 83]) :

- si $\log E_{ech} < \alpha \log E_{moy}$ alors l'échantillon est étiqueté *SILENCE* (S)
- si $\log E_{ech} \geq \beta \log E_{moy}$ alors l'échantillon est étiqueté *PAROLE* (P)
- si $\alpha \log E_{moy} \leq \log E_{ech} < \beta \log E_{moy}$ alors l'échantillon se trouve dans la zone d'incertitude, illustrée la figure 4.1. L'analyse des échantillons précédents, constituant le contexte local gauche, permet de lever l'incertitude. L'étiquette majoritairement attribuée aux échantillons du contexte local gauche est assignée à l'échantillon considéré. Le contexte local gauche est choisi ici de longueur fixe et regroupe plusieurs centaines d'échantillons.

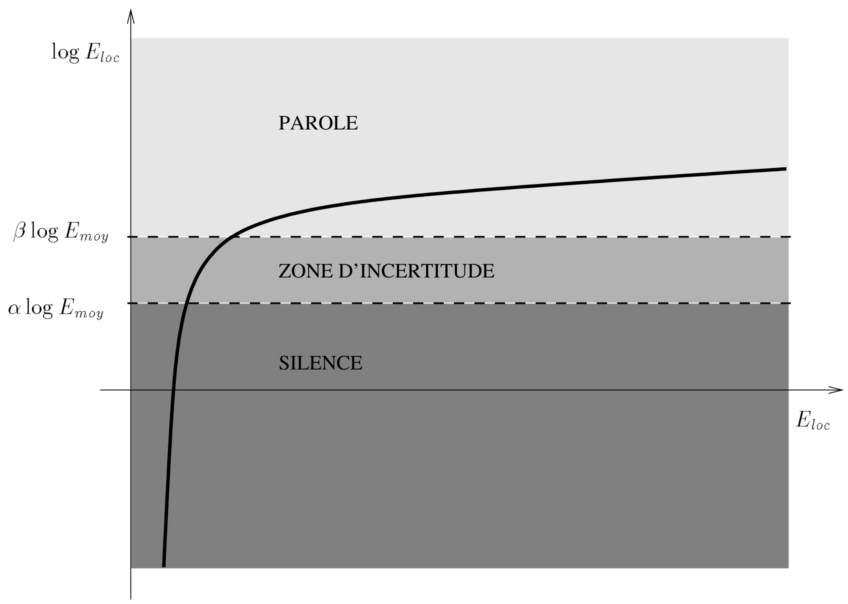

FIG. 4.1 – *Principe d'étiquetage des échantillons ou trames*

Une fois l'ensemble des échantillons étiquetés, il reste à déterminer les silences. Un silence correspond à une séquence significative en terme de longueur (de l'ordre de quelques dixièmes de secondes) d'étiquettes contigües *SILENCE* comme indiqué à la figure 4.2 (l'échelle de temps n'est pas respectée pour une meilleure compréhension).

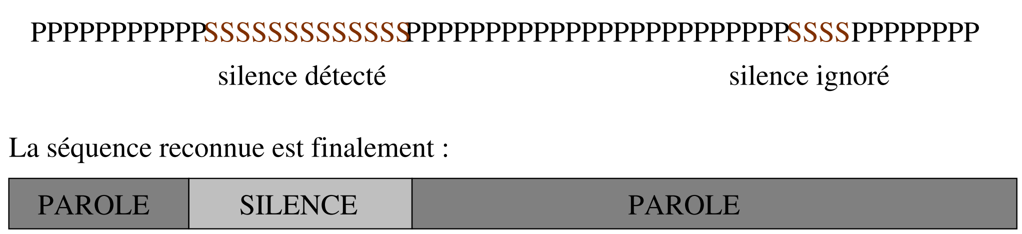

FIG. 4.2 – *Reconnaissance des silences*

Le choix des valeurs des paramètres est discuté au chapitre 5.2.

Comme nous l'avons vu au paragraphe 3.1, l'utilisation des techniques de segmentation par détection de silences suppose des silences suffisamment longs entre les différents locuteurs. Or, c'est rarement le cas, en particulier dans une conversation téléphonique. Néanmoins, certains types de signaux contiennent de tels silences : les journaux télévisés ou les messages laissés sur un répondeur téléphonique ou dans une boîte vocale. Dans le cas des journaux télévisés, il peut être intéressant d'utiliser la segmentation par détection de silences comme un pré-traitement du signal et d'appliquer un autre type de segmentation plus robuste sur chaque segment obtenu. Dans le cas d'une boîte vocale ou d'un répondeur téléphonique, la segmentation par détection de silences suffit sans doute à récupérer des segments homogènes ne contenant qu'un seul locuteur.

4.2 DISTBIC : segmentation par détection de changement de locuteurs

Nous présentons ci-après une méthode de segmentation qui se déroule en deux passes (nous n'avons aucune contrainte de temps réel dans le cadre de l'indexation). La première passe détecte les changements de locuteurs les plus probables à l'aide d'une segmentation basée sur le calcul d'une distance. La procédure de détection des changements fait intervenir un seuil aussi robuste que possible pour ce type de méthode. Robuste signifie dans notre contexte que le seuil est d'une part calculé de manière automatique et d'autre part, efficace en termes de résultats, quelque soit le type de signaux audio traités (parole spontanée, parole préparée, parole téléphonique...). La deuxième passe valide ou non les changements de locuteurs trouvés lors de la première passe à l'aide du Critère d'Information Bayesien (BIC) détaillé au paragraphe 3.2.3. Cette technique de segmentation, nommée DISTBIC est illustrée à la figure 4.3.

Un couple de fenêtres adjacentes est déplacé le long du signal audio paramétrisé. Pour chaque couple de fenêtres, la distance qui les sépare est calculée. Une fois obtenue la courbe des distances en fonction du temps pour l'ensemble du signal, les instants de changements de locuteurs sont détectés. Cette segmentation préliminaire est alors raffinée à l'aide du critère BIC : les instants de changements trouvés précédemment sont validés ou non.

Dans un premier temps, nous détaillons la première passe : nous nous attardons sur le cas particulier de la détection d'un seul changement, notamment sur les distances qui peuvent être employées (cf 4.2.1). Nous nous intéressons ensuite à la détection de plusieurs changements de locuteurs : d'une part, la manière dont sont choisies les fenêtres glissantes (cf 4.2.2) et d'autre part, la détection des changements à partir de la courbe des distances (cf 4.2.3). Enfin, nous terminons ce chapitre par la deuxième passe, à savoir le raffinement à l'aide du critère BIC et la façon dont il est appliqué (cf 4.2.4).

Comme nous faisons l'hypothèse que nous n'avons aucune connaissance a priori sur les locuteurs présents dans la conversation, notre algorithme de détection de changements de locuteurs est proche d'un algorithme classique de détection de ruptures dans un signal. Cependant, il est appliqué à un signal paramétrisé de parole, i.e. qui contient des informations spécifiques du locuteur. Et ce sont les changements de ce signal paramétrisé qui nous importent et donc a fortiori les changements de locuteurs.

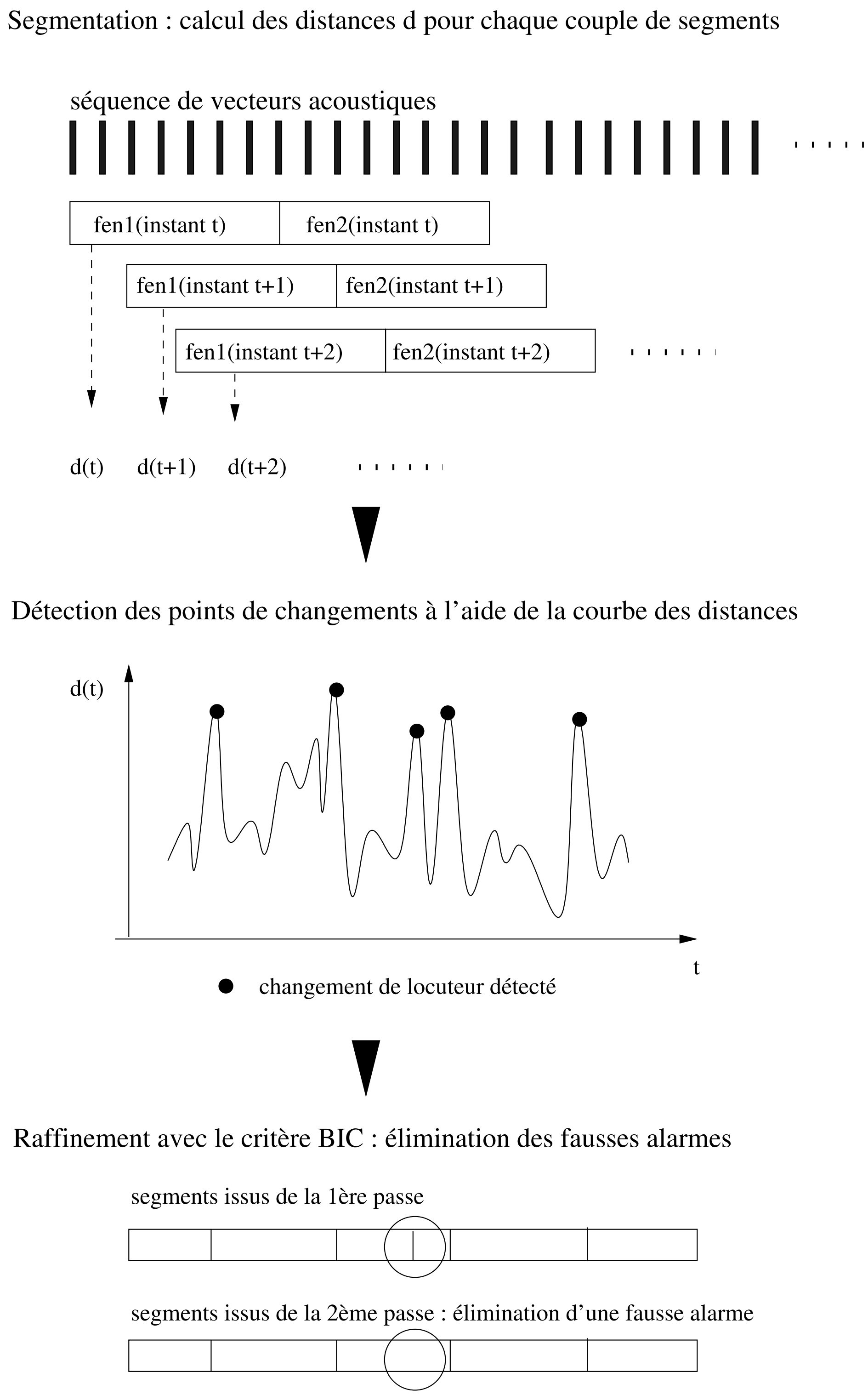

FIG. 4.3 – *Les différentes étapes de notre méthode de segmentation*

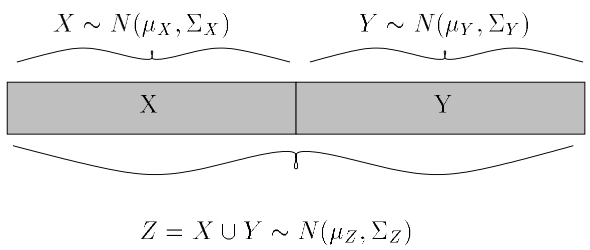

FIG. 4.4 – *Modélisation des segments de vecteurs acoustiques*

4.2.1 Détection d'un changement de locuteur

Soient deux segments de signal paramétrisé adjacents, à la frontière desquels nous supposons l'existence éventuelle d'un changement de locuteur. Autrement dit, nous supposons que les deux segments aient été prononcés par deux locuteurs différents. Pour tester cette hypothèse, nous mesurons à l'aide d'une distance la similarité entre les deux portions de signal. Une forte similarité, i.e. une faible valeur de la distance indique que les deux segments proviennent du même locuteur. Un résultat opposé indique que chaque segment est prononcé par un locuteur différent.

Soient les deux segments X et Y adjacents en question comme l'indique la figure 4.4. Nous avons ici deux segments de même longueur mais le principe peut être appliqué à des segments de longueur différente. Un segment est une séquence de vecteurs acoustiques $X = \{x_1...x_{N_X}\}$. Pour les distances que nous décrivons ci-après, nous supposons que chaque segment est prononcé par un seul locuteur et que la séquence de vecteurs acoustiques associée, X, est régie par un processus Gaussien multi-dimensionnel $X \sim N(\mu, \Sigma)$ où μ_X est le vecteur moyen, Σ_X la matrice de covariance (supposée pleine) et p la dimension des vecteurs acoustiques. Nous désignons par $\mathcal{Z}$ la réunion des deux segments.

Nous verrons au paragraphe (4.2.2) qu'il y a un compromis à faire sur la durée des fenêtres et qu'en pratique, elles durent deux secondes. Cette durée ne nous autorise pas à avoir des modèles de locuteurs très sophistiqués (comme par exemple des mixtures de Gaussiennes *GMM*). En effet, nous n'avons pas suffisamment de données pour estimer de manière fiable et robuste de tels modèles, d'où l'utilisation d'une simple Gaussienne.

Nous présentons ci-après plusieurs mesures de distance que nous avons testées (cf chapitre 5). Il est à noter que nous employons le terme de distance à tort car les mesures proposées ci-dessous ne vérifient pas les propriétés d'une distance, notamment l'inégalité triangulaire.

Le rapport de vraisemblance

Le rapport de vraisemblance généralisé (*Generalized Likelihood Ratio (GLR)*) est utilisé par [Gish et al. 91, Gish et al. 94] pour l'identification du locuteur. Ce rapport de vraisemblance a été détaillé au paragraphe 2.1. Nous en reprenons ci-dessous les grandes lignes. Soit le test d'hypothèses suivant :

- H_0 : les segments sont prononcés par le même locuteur. Alors la réunion des deux segments est générée par un unique processus Gaussien multi-dimensionnel.
- H_1 : les segments sont prononcés par des locuteurs différents. Alors les segments sont générés par des processus Gaussiens multi-dimensionnels différents.

Le rapport de vraisemblance associé à ce test est donné par :

$$R = \frac{L(z, \widehat{\mu}; \widehat{\Sigma})}{L(x; \widehat{\mu}_1; \widehat{\Sigma}_1) L(y; \widehat{\mu}_2; \widehat{\Sigma}_2)} \tag{4.3}$$

où $L(X, N(\mu_X, \Sigma_X))$ représente la vraisemblance de la séquence de vecteurs acoustiques X étant donné le processus Gaussien multi-dimensionnel $N(\mu_X, \Sigma_X)$. Pour obtenir une mesure de distance entre les deux segments, l'opposé du logarithme du rapport de vraisemblance est considéré :

$$d_{\mathrm{GLR}} = -\log R \tag{4.4}$$

La distance de Kullbach-Leibler

La distance de Kullbach-Leibler est utilisée par [Siegler et al. 97] et est détaillée au paragraphe 3.2.1. Nous en rappelons l'expression générale :

$$\mathrm{KL}(X, Y) = \mathrm{E}_X \left\langle \log(P_X(X)) - \log(P_Y(X)) \right\rangle \tag{4.5}$$

où $E_X < . >$ est l'espérance mathématique calculée avec la densité de probabilité P de X. Ayant fait l'hypothèse de processus Gaussien pour chaque segment, la formule 3.3 s'applique. Par contre, nos modèles étant multi-dimensionnels, nous généralisons cette formule pour toute dimension. Nous démontrons la formule suivante de la distance de Kullbach-Leibler dans le cas multi-dimensionnel en annexe B :

$$\begin{aligned} KL(X,Y) &= \frac{1}{2}(\mu_Y - \mu_X)^T (\Sigma_X^{-1} + \Sigma_Y^{-1})(\mu_Y - \mu_X) \\ &+ \frac{1}{2} tr((\Sigma_X^{\frac{1}{2}} \Sigma_Y^{\frac{-1}{2}})(\Sigma_X^{\frac{1}{2}} \Sigma_Y^{\frac{-1}{2}})^T) \\ &+ \frac{1}{2} tr((\Sigma_X^{\frac{-1}{2}} \Sigma_Y^{\frac{1}{2}})(\Sigma_X^{\frac{-1}{2}} \Sigma_Y^{\frac{1}{2}})^T) - p \end{aligned} \tag{4.6}$$

où tr désigne la trace d'une matrice.

Mesures de similarité

Toutes les mesures de similarité présentées dans ce paragraphe sont décrites en détail dans [Bimbot et al. 95]. Ces mesures reposent sur l'hypothèse que deux segments de signal paramétrisé, X et Y, ont des matrices de covariance similaires, respectivement Σ_X et Σ_Y, si ces segments ont été générés par le même locuteur. Plus formellement, pour mesurer la similarité entre les segments X et Y, la matrice $\Gamma = \Sigma_X \Sigma_Y^{-1}$ est considérée. Si les deux segments ont été prononcés par le même locuteur, alors $\Sigma_X = \Sigma_Y$ et Γ est la matrice identité I_p.

La première mesure de similarité est définie par :

$$\mu_G(X, Y) = a - \log g + \frac{1}{p}(\mu_X - \mu_Y)\Sigma_X^{-1}(\mu_X - \mu_Y)^T - 1 \tag{4.7}$$

où μ_X et μ_Y sont les vecteurs acoustiques moyens respectifs de X et Y, a est la moyenne arithmétique des valeurs propres λ_i de la matrice Γ et g est la moyenne géométrique. Cette mesure de similarité provient du rapport de vraisemblance généralisé (cf [Bimbot et al. 95]). Si $\Sigma_X = \Sigma_Y$ (i.e. $X = Y$), alors $\mu_G = 0$, sinon $\mu_G > 0$.

Une seconde mesure de similarité est déduite de la première. Elle repose sur le fait que les vecteurs moyens peuvent être affectés par le canal de transmission et par conséquent, ne doivent pas être pris en compte :

$$\mu_{GC}(X,Y) = a - \log g - 1 \tag{4.8}$$

La troisième mesure de similarité est un test de sphéricité de la matrice Γ:

$$\mu_{SC}(X,Y) = \log \frac{a}{g} \tag{4.9}$$

La dernière mesure de similarité μ_{DC} calcule la déviation absolue entre les valeurs propres de la matrice Γ et l'unité :

$$\mu_{DC}(X,Y) = \frac{1}{p}\sum_{i=1}^{p} |\lambda_i - 1| \tag{4.10}$$

Ces mesures ne sont pas symétriques telles qu'elles sont définies. Elles sont symétrisées de la manière suivante:

$$\mu_S(X,Y) = \mu(X,Y) + \mu(Y,X) \tag{4.11}$$

où μ représente μ_G, μ_{GC}, μ_{SC} ou μ_{DC}.

4.2.2 Application à la détection de plusieurs changements de locuteur

Pour généraliser le principe de la détection d'un changement de locuteur à la détection de plusieurs changements de locuteurs, nous utilisons un couple de fenêtres adjacentes et glissantes. Il n'y a pas de recouvrement entre les deux fenêtres, comme l'indique la figure 4.5. Il faut faire un compromis sur la durée des fenêtres : d'une part, la durée doit être suffisamment courte pour supposer que la fenêtre ne recouvre les paroles que d'un seul locuteur et d'autre part, la durée doit être suffisamment longue pour que l'estimation des paramètres à partir des données contenues dans les fenêtres pour le calcul des distances soit fiable. Un bon compromis est de fixer la durée des fenêtres à deux secondes.

A chaque itération, les fenêtres sont déplacées de quelques dizaines de millisecondes (en général, 100 ms). Cela signifie qu'il y a recouvrement des couples de fenêtres d'une itération à l'autre. Cela signifie également que la position des points de changements est évaluée avec une précision d'au moins quelques dizaines de millisecondes, i.e. la durée du décalage entre deux itérations.

Pour chaque couple de fenêtres, une distance est calculée comme expliqué au paragraphe 4.2.1 et est stockée. Ce processus est répété tout le long du signal audio, de sorte que nous obtenons une courbe de distances à l'issue du processus. Dans ce qui suit, nous nous intéressons à la détection des changements de locuteurs à partir de la courbe des distances.

4.2.3 Détection des changements de locuteurs à partir de la courbe des distances

Le principe de la segmentation par détection de changement de locuteurs comme vu à la section 2.1, repose sur le calcul d'une distance entre deux portions de signal paramétrisé. Selon la valeur de cette distance, soit les deux portions sont proclamées prononcées par le même locuteur, soit les deux portions ont été générées par des locuteurs différents. Nous avons

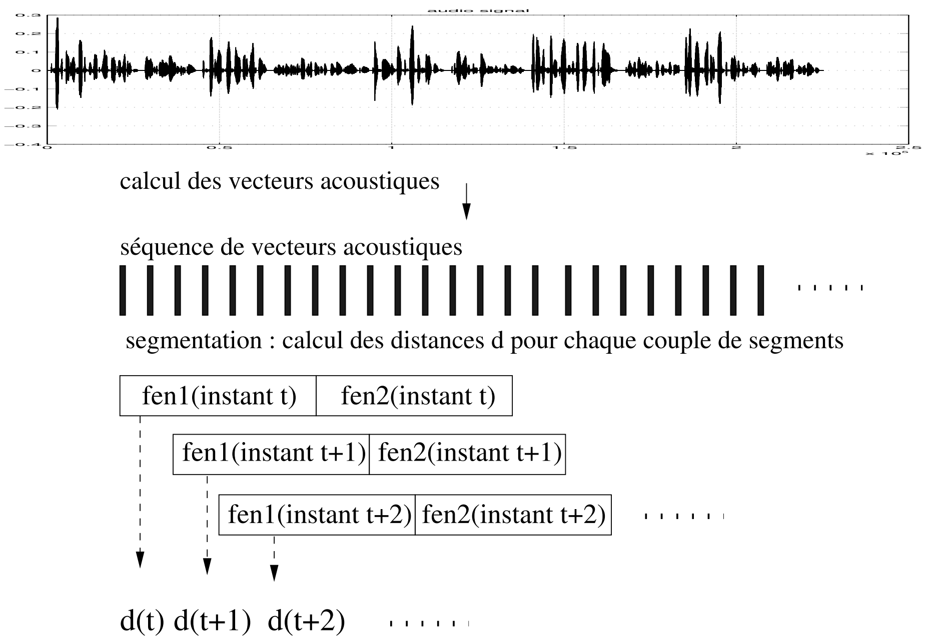

FIG. 4.5 – *Principe des fenêtres glissantes*

constaté au travers des exemples étudiés que le problème majeur de cette technique réside dans le choix du seuil de décision.

Comme expliqué au paragraphe précédent, une distance est calculée pour chaque couple de segments et ce processus est répété tout le long du signal. Nous avons donc à notre disposition la courbe des distances à la fin du processus. De plus, nous savons qu'un changement de locuteur se traduit par une forte valeur de la distance. Nous nous attachons donc à rechercher les maxima de la courbe.

Les mesures de distance que nous utilisons n'étant pas normalisées et les valeurs de ces mesures étant fort variables d'un signal audio à un autre, nous recherchons un seuil relatif. Après étude des occurrences des points de changement de locuteurs dans la courbe des distances, il apparaît que ce n'est pas l'amplitude des maxima qui importe mais plutôt leur forme. En effet, un changement de locuteur correspond dans la majorité des cas à un maximum local caractérisé par un pic significatif de la courbe des distances, i.e. un pic fortement marqué comme l'illustre la figure 4.6. Il nous reste alors à caractériser ce pic significatif.

Tout d'abord, la courbe des distances étant bruitée (présence de nombreux artéfacts), elle est lissée en remplaçant chaque valeur par la moyenne de ses voisines (filtre passe-bas moyenneur). Un pic est considéré comme significatif si la différence entre son amplitude et l'amplitude de ses minima de part et d'autre est importante. Plus formellement, soit σ la variance de la distribution des distances d. Un pic est significatif si :

$$|d(max) - d(min_d)| > \alpha\sigma \text{ et } |d(max) - d(min_g)| > \alpha\sigma \tag{4.12}$$

où α est un réel positif ($\alpha > 0$), min_d et min_g représentent respectivement le minimum à droite du pic correspondant au maximum max et le minimum à gauche du même pic. Ceci

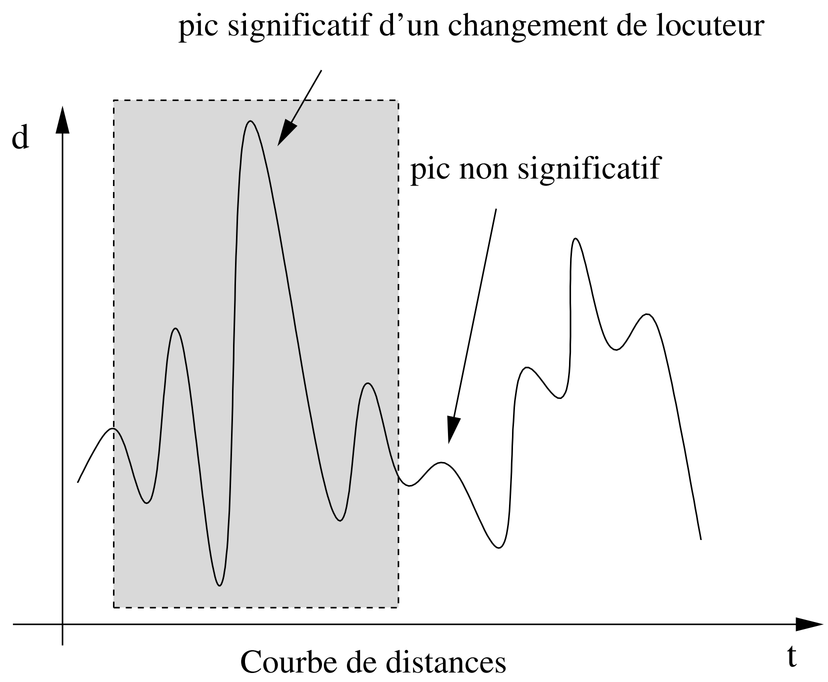

FIG. 4.6 – *Courbe des distances : pic significatif d'un changement de locuteur*

est illustré à la figure 4.7. La détection des changements de locuteurs se ramène donc à la recherche des maxima de la courbe des distances vérifiant le critère 4.12.

De plus, nous imposons une durée minimale entre deux changements de locuteurs. Si deux maxima sont trop proches (de l'ordre de quelques dixièmes de secondes) l'un de l'autre, alors nous ne conservons que le maximum des deux.

Cette méthode présente l'avantage de s'adapter à tout type de signal audio. Elle ne fait pas intervenir de seuil absolu. Par ailleurs, elle est conçue pour éviter au maximum les détections manquées, c'est-à-dire les changements de locuteurs existant et qui ne sont pas détectés. Mais ceci se fait au détriment du nombre de fausses alarmes (un changement de locuteur est détecté alors qu'il n'existe pas). Avant de passer à l'étape suivante de notre système d'indexation, nous aimerions réduire ce nombre de fausses alarmes en regroupant les segments adjacents appartenant à un même locuteur, sans pour autant augmenter le nombre de détections manquées. Ceci est fait à l'aide du Critère d'Information Bayesien qui est l'objet du paragraphe suivant.

4.2.4 Raffinement à l'aide du Critère d'Information Bayesien

L'utilisation du Critère d'Information Bayesien (BIC) pour la segmentation en locuteurs a été introduite par [Chen et al. 98c] et est longuement détaillée au paragraphe 3.2.3.

Nous faisons appel à ce critère dans le but de réduire le nombre de fausses alarmes résultant de la première passe. Il s'agit donc de valider ou non les points de changement de locuteurs $\{s_1....s_N\}$ trouvés lors de la première passe.

Pour valider le changement de locuteur en s_{i+1}, nous considérons les segments $[s_i, s_{i+1}]$ et

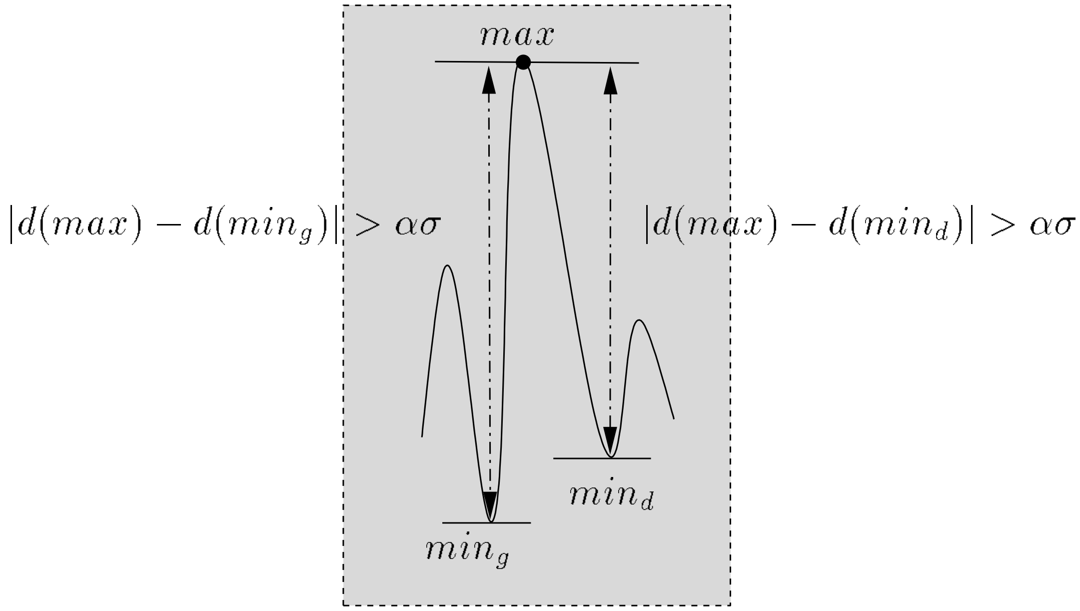

FIG. 4.7 – *Courbe des distances : caractérisation d'un changement de locuteurs*

$[s_{i+1}, s_{i+2}]$ et nous leur apliquons le critère BIC, i.e. nous comparons la modélisation de ces deux segments à l'aide de deux Gaussiennes (une Gaussienne par segment) et la modélisation à l'aide d'une seule Gaussienne, comme l'indique la figure 4.8. Cela revient respectivement à l'hypothèse H_1 (un changement de locuteur à l'interface des deux segments) et à l'hypothèse H_0 (pas de changement de locuteur) du test d'hypothèses du paragraphe 4.2.1.

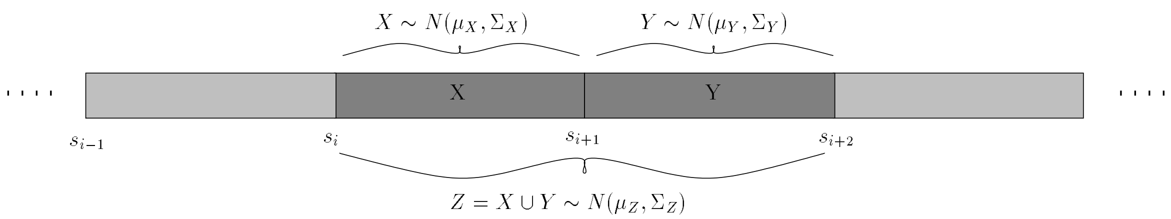

FIG. 4.8 – *Segments considérés pour le raffinement à l'aide du critère* BIC

La différence de critère BIC entre les deux modélisations va nous permettre de choisir la modélisation la plus adéquate aux données. Cette différence est donnée par :

$$\Delta\mathrm{BIC}(s_{i+1}) = -R(s_{i+1}) + \lambda P \tag{4.13}$$

où $R(i)$ désigne le rapport de maximum de vraisemblance entre l'hypothèse H_0 et l'hypothèse H_1 (cf équation 4.3), P le terme de pénalité et λ le poids de pénalité.

Si cette différence de BIC est négative alors le changement de locuteur en s_{i+1} est validé puisque la modélisation à deux Gaussiennes est préférée selon ce critère. Sinon, ce changement de locuteur est annulé et les deux segments sont réunis pour ne former qu'un à la prochaine itération, i.e. le couple de segments $[s_i, s_{i+2}]$ et $[s_{i+2}, s_{i+3}]$ sera considéré, comme illustré à la figure 4.9.

Cette deuxième passe est en fait identique à la troisième passe de la procédure BIC (cf

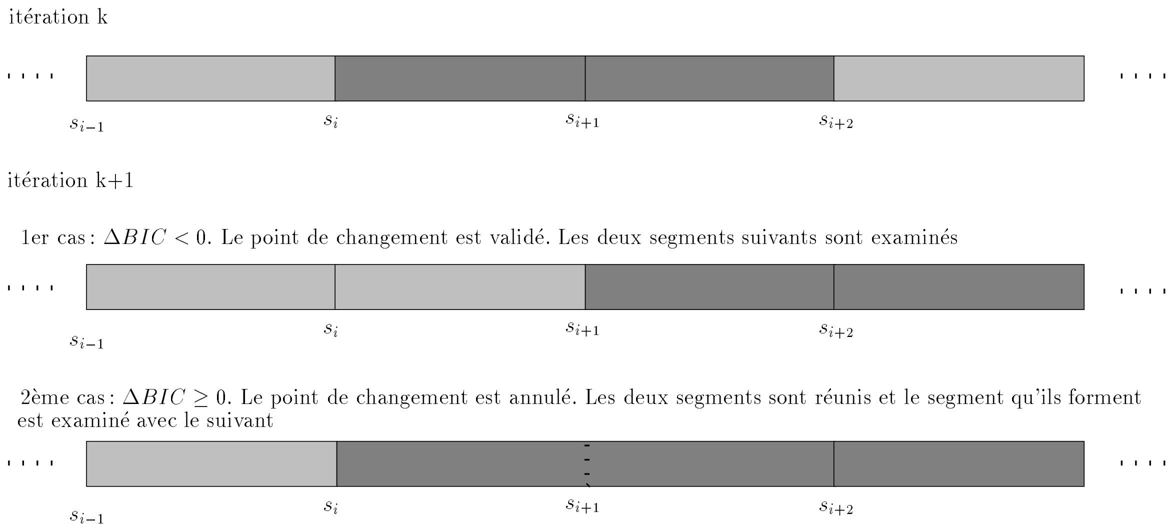

FIG. 4.9 – *Validation ou annulation des points de changement*

paragraphe 3.2.3).

Les performances des méthodes de détection de silences et de segmentation exposées dans ce chapitre, respectivement SILHYST et DISTBIC, sont évaluées au chapitre suivant.

Chapitre 5

Expériences

Dans ce chapitre, nous relatons les expériences que nous avons menées afin d'évaluer les techniques de segmentation que nous proposons. Tout d'abord, nous définissons à la section 5.1 des taux qui permettent de quantifier les performances des techniques de segmentation en locuteurs. Ensuite, nous donnons les résultats des expériences réalisées avec la technique de segmentation SILHYST par détection de silences à la section 5.2. Nous détaillons à la section 5.3 les résultats obtenus avec la technique de segmentation DISTBIC par détection de changements de locuteurs. Enfin, nous concluons sur ces techniques de segmentation.

5.1 Méthodes d'évaluation

Une bonne segmentation fournit les changements de locuteurs corrects et des segments ne contenant qu'un seul locuteur. Nous distinguons deux types d'erreurs pour la détection de changements de locuteurs. Une *fausse alarme* (FA) a lieu lorsqu'un changement de locuteur est détecté alors qu'il n'existe pas. Une *détection manquée* (DM) a lieu quand un changement de locuteur existant n'est pas détecté. Dans notre contexte, ce deuxième type d'erreur est plus grave que le premier type. En effet, un segment corrompu (i.e. contenant plusieurs locuteurs) peut détériorer l'étape de regroupement du système d'indexation par locuteurs. A l'inverse, une fausse alarme donc une sur-segmentation peut être résolue lors de cette même étape de regroupement des segments.

Nous définissons le taux de fausses alarmes (TFA) comme suit :

$$\mathrm{TFA} = 100 \times \frac{\text{nombre de FA}}{\text{nbre de changements réels} + \text{nombre de FA}}\% \tag{5.1}$$

Une valeur élevée de TFA est significative d'une sur-segmentation (*over-segmentation*). Le taux de détections manquées (TDM) est défini par :

$$\mathrm{TDM} = 100 \times \frac{\text{nombre de DM}}{\text{nbre de changements réels}}\% \tag{5.2}$$

Une valeur élevée de TDM indique une sous-segmentation (*under-segmentation*). Notre système a été paramétré de manière à obtenir de faibles valeurs de TFA et de TDM avec la contrainte supplémentaire que le TDM soit inférieur au TFA. Ceci ne correspond pas à l'objectif classique du taux d'égale erreur (*EER : equal error rate*).

Pour calculer ces taux, il est nécessaire d'avoir à disposition la segmentation de référence du document audio concerné. Cependant, la précision avec laquelle l'oreille humaine détecte les changements de locuteurs est assez faible. Cette imprécision peut être due par exemple à des respirations ou encore à des soupirs. Aussi, la segmentation de référence (quand elle existe) doit pouvoir accepter une certaine tolérance sur les limites des segments. Par exemple, si nous comparons des segmentations de référence réalisées "à la main" par différentes personnes, il y a de fortes chances pour que ces segmentations diffèrent. En ce qui concernent les signaux synthétiques, les changements de locuteurs sont évidemment connus, par construction.

Cette tolérance peut être prise en compte en définissant des intervalles de précision centrés sur les changements de locuteurs de référence ou les changements détectés. Ainsi, un point de changement détecté est une fausse alarme si dans l'intervalle de précision qui l'entoure aucun changement de référence n'est trouvé. Par ailleurs, si à l'intérieur d'un intervalle de précision centré sur un changement de référence, aucun changement de locuteur n'a été détecté, nous sommes dans ce cas en présence d'une détection manquée (cf aussi [Liu et al. 99]).

L'utilisation de tels intervalles de précision n'est pas encore tout à fait satisfaisante : en effet, la largeur de ces intervalles devrait tenir compte de la vitesse d'élocution de la personne qui parle ou encore du contenu sémantique de la conversation. Un ultime test pour juger de la qualité de la segmentation consiste à écouter individuellement chaque segment obtenu et en vérifier l'intégralité.

5.2 Evaluation de SILHYST

5.2.1 Données

L'algorithme de segmentation en locuteurs par détection de silences est évalué sur différents types de données de parole :

- 10 conversations créées artificiellement en concaténant des phrases extraites de la base de données TIMIT (parole propre, segments courts, anglais, 285 changements de locuteurs)
- 10 conversations créées artificiellement en concaténant des phrases extraites de la base de données fournie par le Centre National d'Etudes des Télécommunications (CNET) (parole propre, segments courts, français, 270 changements de locuteurs)
- 4 journaux télévisés (JT) enregistrés dans notre laboratoire (segments de toute longueur, parole spontanée et préparée, français, 160 minutes).

Les silences de fin de phrase dans les conversations créées artificiellement ne sont pas supprimés. Cela donne lieu à de longs silences inter-locuteurs, mais également à de longs silences intra-locuteurs car il arrive que plusieurs phrases d'un même locuteur s'enchaînent.

Nous abandonnons volontairement les tests sur les conversations téléphoniques de SWITCHBOARD (pour une description détaillée de cette base de données, consulter [Godfrey et al. 92]). Il est en effet illusoire de vouloir segmenter en locuteurs une conversation téléphonique par détection de silences. Nous avons vu au chapitre précédent que cette méthode ne pouvait être efficace que si les locuteurs étaient séparés par des silences significatifs en terme de durée. Or, dans une conversation téléphonique, la parole est en général spontanée et donc, les échanges sont rapides, quitte d'ailleurs à couper la parole à l'interlocuteur. Par ailleurs, les personnes ne se voyant pas, elles ont tendance naturellement à combler les "blancs".

SILHYST	α	β	T en s
TIMIT	0.7	0.9	0.3
CNET	0.7	0.9	0.15
JT	0.7	0.9	0.3

TAB. 5.1 – *Valeurs des paramètres de SILHYST pour les différents types de données de parole*

Par contre, les journaux télévisés se prêtent bien à ce type de segmentation. Il existe par exemple de longs silences entre la fin des paroles du journaliste-présentateur et le début d'un reportage. Mis à part les interviews en direct ou les interviews spontanées enregistrées dans un reportage, la parole est préparée, i.e. elle est posée et l'enchaînement des locuteurs se fait à un rythme raisonnable.

Nous travaillons sur le signal temporel : SILHYST ne nécessite pas de paramétrisation préalable.

5.2.2 Expériences

Avant d'examiner les résultats de la segmentation proprement dite, nous nous intéressons d'abord à la valeur des paramètres, puis à la qualité des silences détectés.

Valeur des paramètres

Le tableau 5.1 donne la valeur des paramètres utilisés pour les différents types de données. α et β sont respectivement les seuils haut et bas pour le seuillage par hystérésis et T représente la durée minimale d'un silence.

Des expériences d'évaluation de α et β sur différents types de données de parole montrent que les valeurs indiquées au tableau 5.1 sont indépendantes du type de données. Le caractère universel de ces valeurs provient du fait que ce sont des seuils adaptatifs par le biais du calcul de l'énergie locale moyenne : α et β représentent des pourcentages de cette énergie locale moyenne.

Par contre, T est la durée minimale des silences que l'utilisateur souhaite détecter. Si cette durée est courte, alors la majeure partie des silences intra-locuteurs et inter-locuteurs vont être détectés (les silences intra-locuteurs sont en général plus courts que les silences inter-locuteurs). A l'inverse, plus cette durée augmente, moins il y a de silences détectés, voire aucun silence si cette durée est fixée à une durée supérieure à tous les silences présents dans la conversation. Même si cette durée peut varier d'un signal à l'autre, elle présente l'avantage de pouvoir être estimée expérimentalement de manière simple et rapide. De plus, la dynamique des valeurs potentielles reste faible.

Résultats de la segmentation

Le tableau 5.2 présente les taux de fausses alarmes (TFA) et de détections manquées (TDM) à l'issue de la segmentation par détection de silences sur les différents types de données.

En ce qui concerne les conversations synthétiques, nous avons vu au paragraphe précédent qu'elles contenaient de longs silences intra- et inter-locuteurs. Nous retrouvons ici ce fait. En effet, la détection d'un silence entre deux phrases d'un même locuteur conduit tout naturellement à une fausse alarme, ce qui explique le taux élevé de fausses alarmes pour TIMIT et

SILHYST	TFA	TDM
TIMIT	80.9%	0.0%
CNET	72.7%	8.6%
JT	23.8%	69.2%

TAB. 5.2 – *Taux de fausses alarmes et de détections manquées de SILHYST avec les différents types de données de parole*

pour les données CNET. A l'inverse, de longs silences inter-locuteurs facilitent la détection des changements de locuteurs par détection de silences, d'où les faibles taux de détections manquées. Cependant, ces résultats sont à nuancer car les conversations que nous utilisons (sans suppression des silences) ressemblent peu à des conversations réelles. Les résultats obtenus sur les journaux télévisés sont à ce point de vue plus intéressants car ils correspondent à un signal réel de parole.

Le TFA obtenu pour les *JT* est de 23.8%. Ce taux est comparable au taux obtenu avec la méthode DISTBIC (cf tableau 5.8). Ces fausses alarmes sont dues à des silences intra-locuteurs significatifs. Ils ont lieu par exemple quand le journaliste-présentateur évoque les divers sujets qui seront abordés au cours du journal télévisé. En général, une pause marque le changement de sujet, de même au cours du journal et ces pauses sont comptabilisées comme fausses alarmes.

Par contre, le TDM est très élevé, ce qui montre que la détection de silences fournit une très mauvaise segmentation en locuteurs. Ceci est prévisible dans la mesure où les locuteurs ne sont pas séparés par des silences significatifs. Ces détections manquées de changement de locuteurs apparaissent surtout lors d'interviews. En effet, il est fréquent que le journaliste pose la question suivante à la personne interviewée sans attendre que cette dernière ait réellement fini de parler. Elles apparaissent également lors de traductions simultanées. Cependant, dans ce dernier cas, nous sortons de nos hypothèses de travail car il y a recouvrement de paroles de deux personnes différentes.

Qualité des silences détectés

Pour compléter cette étude, nous nous intéressons à la qualité des silences détectés. Nous voudrions savoir si ceux-ci sont purs ou au contraire, contiennent de la parole. Dans notre application, il importe en effet que seuls des silences purs soient détectés. Nous définissons donc le taux de qualité de la manière suivante :

$$\mathrm{TQ} = 100 \times \left(1 - \frac{\text{nombre de silences contenant de la parole}}{\text{nombre total de silences détectés}}\right)\% \tag{5.3}$$

Plus ce taux est élevé, plus les silences détectés sont purs.

Le tableau 5.3 donne les valeurs de ce taux pour les différents types de données. Nous constatons que dans les trois cas, les silences possèdent un haut niveau de pureté.

Les silences jugés impurs sont dans la majorité des cas des silences qui contiennent la première ou la dernière syllabe d'un mot selon qu'ils soient en début ou en fin de phrase. Les silences qui contiennent la respiration du locuteur avant de parler ne sont pas comptabilisés comme segments impurs.

SILHYST	TQ
TIMIT	98.6%
CNET	94.7%
JT	100.0 %

TAB. 5.3 – *Taux de qualité de SILHYST pour les différents types de données de parole*

5.2.3 Commentaires

De cette étude, nous pouvons conclure les choses suivantes. Notre technique de détection de silences présente l'avantage d'utiliser des paramètres qui ne varient pas ou peu d'un signal de parole à l'autre. Dans le cas où ils varient, leur valeur est facilement ajustable parce que directement liée à la durée réelle d'un silence.

Nous avons vu que la détection de silences n'est pas adaptée pour la détection de changements de locuteurs sauf si ceux-ci sont séparés par des silences significatifs.

Cependant, les longs silences peuvent venir perturber la détection de changements de locuteurs par calcul de distance. En effet, ces silences peuvent être suffisamment longs pour être en partie détectés mais pas suffisamment longs pour l'être complètement (seule une des bornes du silence est détectée), ce que nous reverrons au paragraphe 5.3.3 à propos des interjections dans les conversations téléphoniques. Par ailleurs, nous avons vu que les silences détectés étaient de bonne qualité, i.e. ils ne contiennent pas de parole. Aussi, nous préconisons l'emploi de notre détection de silences comme un pré-traitement à la segmentation DISTBIC, dont les performances sont présentées ci-après.

5.3 Evaluation de DISTBIC

5.3.1 Données

Différents types de données de parole ont été utilisés pour comparer notre algorithme de segmentation avec l'algorithme proposé par S.Chen, appelé procédure BIC (cf section 3.2.3):

- 2 conversations qui ont été créées artificiellement en concaténant des phrases de 2 secondes en moyenne extraites de la base de données TIMIT (parole propre, segments courts, anglais, 60 changements de locuteurs).

- 2 conversations créées en concaténant des phrases de 1 à 3 secondes extraites d'une base de données fournie par le CNET (parole propre, segments courts, français, 45 changements de locuteurs).

- 3 journaux télévisés extraits de la base de données de l'Institut National de l'Audiovisuel (INA) (segments de toute longueur, parole spontanée et préparée, français, 85 changements de locuteurs, 30 minutes).

- 3 conversations téléphoniques extraites de la base de données SWITCHBOARD ([Godfrey et al. 92]) (segments de toute longueur, parole spontanée, anglais, 120 changements de locuteurs, 30 minutes).

Pour les conversations synthétiques créées, les silences entre les différents locuteurs ont été réduits de manière à ressembler à des silences inter-locuteurs d'une conversation réelle. Plus précisément, les silences inter-locuteurs obtenus sont volontairements courts pour ne pas être détectés par notre algorithme. Chaque segment de locuteur est suivi par un segment d'un autre locuteur.

Afin de réaliser des tests complémentaires de notre algorithme DISTBIC, nous utilisons quatre journaux télévisés enregistrés dans notre laboratoire et nous analysons la nature des erreurs observées.

- 4 journaux télévisés français (référencés *jt*) (segments de toute longueur, parole spontanée et préparée, français, 830 changements de locuteurs, 135 minutes).

Pour la paramétrisation du signal de parole, nous utilisons des coefficients Mel-cepstraux. Nous renvoyons à [Davis et al. 80] pour la définition de ce type de coefficients acoustiques. Ces coefficients ont en effet prouvé leur efficacité en reconnaissance du locuteur ([Furui 81]). Ils sont calculés avec des fenêtres d'analyse de 32 ms espacées de 10 ms (la fréquence d'échantillonnage de nos signaux est de 8 kHz). Nous avons également testé cette paramétrisation complétée avec les Δ-coefficients (dérivées premières). D'après [Soong et al.88], l'ajout de ces derniers améliore les taux de reconnaissance de locuteur. L'utilisation de ces Δ-coefficients détériore les performances des deux passes : les maxima de la courbe de distance sont lissés, rendant ainsi la détection des pics correspondant à un changement de locuteur plus délicate. Quant au raffinement avec le BIC, il est sensible à la dimension des vecteurs acoustiques. Par ailleurs, la charge de calculs est augmentée de manière non négligeable. Aussi, nous n'utilisons pas ces $\Delta-$coefficients.

5.3.2 Choix de la mesure de distance pour la première passe

La figure 5.1 montre les courbes de distance obtenues avec les différentes distances détaillées à la section 4.2.1. Nous avons produit ces courbes à partir de données TIMIT. Les lignes verticales indiquent la localisation exacte des changements de locuteurs réels. Et les étoiles ($*$) représentent les changements de locuteurs trouvés par la première passe de notre technique de segmentation DISTBIC. Nous voudrions mettre tout d'abord en évidence la capacité de notre technique à détecter des changements de locuteurs proches les uns des autres. Bien que la distance de Kullbach-Leibler (4.2.1) et le rapport de vraisemblance généralisé (4.2.1) sont les plus couteux en termes de charge de calcul, ils donnent les meilleurs résultats : nous distinguons aisément les maxima correspondant aux changements de locuteurs réels. La mesure de similarité μ_G (4.2.1) semble être un bon compromis puisque la charge de calculs est assez faible et les résultats similaires à ceux donnés par la distance de Kullbach-Leibler et le rapport de vraisemblance généralisé.

Pour les documents audio contenant de la parole spontanée, **nous recommandons l'utilisation du rapport de vraisemblance généralisé**, comme cela est fait par la suite. En effet, cette distance a pour avantages de produire des pics étroits et élevés correspondant aux changements de locuteurs et de faibles variations d'amplitude pour un même locuteur.

BIC	λ	*fen1*	*rés1*	*fen2*	*rés2*
TIMIT	1.3	3 s	0.6 s	2 s	0.2 s
CNET	1.3	3 s	0.6 s	2 s	0.2 s
INA	2.0	3 s	0.6 s	1.8 s	0.15 s
SWITCHBOARD	2.0	3 s	0.6 s	1.8 s	0.15 s

TAB. 5.4 – *Valeurs des paramètres de la procédure* BIC *pour les différents types de données de parole*

5.3.3 Comparaison des techniques de segmentation BIC et DISTBIC

Valeur des paramètres

Les deux techniques de segmentation DISTBIC et BIC étant basées sur des propriétés du signal local, il n'est pas surprenant que le choix des paramètres joue un rôle essentiel. Les paramètres sont ici choisis de manière à satisfaire :

$$\text{TDM} < \text{TFA} \tag{5.4}$$

avec de faibles valeurs de TFA et de TDM.

Les valeurs des paramètres ont été déterminées expérimentalement sur des données de même type que les données de test. Ces valeurs sont confirmées par les résultats obtenus avec les données de test. Le tableau 5.4 donne les valeurs de paramètres pour l'algorithme BIC :

- λ est le poids de pénalisation du critère BIC (cf. équation 3.5)
- *fen1* est la durée en secondes de la fenêtre $d(a, c)$ et *rés1* est la résolution utilisés lors de la première passe de la procédure BIC (cf. figure 3.1)
- de même, *fen2* est la durée en secondes de la fenêtre $d(a, c)$ et *rés2* est la résolution de la deuxième passe (cf. figure 3.1)

Le tableau 5.5 rend compte des valeurs des paramètres utilisées pour l'algorithme DISTBIC :

- λ est le critère de pénalisation du critère BIC (cf équation 3.5)
- α est le coefficient défini à l'équation 4.12 (cf. également la figure 4.7)
- *fen* est la durée en secondes d'une fenêtre et *décal* est le décalage en secondes entre deux fenêtres d'une itération à l'autre (cf. paragraphe 4.2.2 et figure 4.5)

Les longueurs des fenêtres (*fen*, *fen1* et *fen2*) résultent d'un compromis entre les contraintes suivantes :

- une courte durée pour faire l'hypothèse que la fenêtre ne contient les paroles que d'un seul locuteur
- une longue durée pour avoir une bonne estimation des modèles de locuteurs (processus Gaussiens multi-dimensionnels)

DISTBIC	λ	*fen*	*décal*	α
TIMIT	1.2	1.96 s	0.7 s	15%
CNET	1.0	1.96 s	0.7 s	15%
INA	1.8	2 s	0.1 s	50%
SWITCHBOARD	1.5	2 s	0.1 s	50%

TAB. 5.5 – *Valeurs des paramètres de la méthode* DISTBIC *pour les différents types de données de parole*

	BIC	
	TFA	TDM
TIMIT	31.5	30.5
CNET	14.3	50.0
INA	18.3	15.7
SWITCHBOARD	20.3	30.6

TAB. 5.6 – TFA *et* TDM *avec la procédure* BIC

Les paramètres *res1*, *res2* et *décal* déterminent également la précision sur la localisation des changements de locuteurs.

Plus la valeur de λ est élevée, plus la valeur de Δ-BIC a de chances d'être positive, P étant positif (cf équation 3.8). Alors moins de changements de locuteurs seront détectés. De même, plus la valeur de α est élevée, moins il y aura de changements de locuteurs détectés (cf équation 4.12).

Nous pouvons également remarquer que les paramètres ne semblent pas être influencés par la langue. En effet, les valeurs des paramètres diffèrent peu d'une méthode de segmentation à l'autre pour les conversations synthétiques en anglais et en français (TIMIT et CNET). Ceci est également vrai pour les conversations réelles (cf. tableaux 5.4 et 5.5). Les petites différences observées proviennent probablement des conditions d'enregistrement.

Coût calculatoire

Concernant le coût calculatoire, DISTBIC procède en deux passes, il est donc impossible de faire un traitement temps réel (Il en est de même de BIC qui procède en 3 passes). Cependant, le temps de traitement requis pour la détection des changements de locuteurs est largement inférieur au temps d'écoute (quelques minutes comparés à 45 minutes d'enregistrement par exemple). BIC est plus rapide que DISTBIC mais la différence n'est pas significative dans le cadre de l'indexation par locuteurs.

$$\text{BIC} < \text{DISTBIC} << \text{temps d'écoute} \tag{5.5}$$

Comparaison des performances

Pour évaluer les performances de notre technique de segmentation, nous la comparons à la procédure BIC décrite dans [Delacourt et al. 99c]. Pour les deux techniques, nous indiquons le

	$1^{ère}$ passe		2^{nde} passe	
	TFA	TDM	TFA	TDM
TIMIT	40.3	14.3	28.2	15.6
CNET	18.2	16.7	16.9	21.4
INA	37.4	9.03	18.5	13.5
SWITCHBOARD	39.0	29.1	25.9	29.1

TAB. 5.7 – TFA *et* TDM *respectivement avec la première et la seconde passes de notre technique de segmentation*

taux de fausses alarmes (TFA) et le taux de détections manquées (TDM). En ce qui concerne notre technique de segmentation, nous distinguons la segmentation basée sur la distance d_{R} (première passe) et le raffinement à l'aide du critère BIC (seconde passe). Le tableau 5.6 présente les résultats obtenus pour la procédure BIC appliquée à différents types de données décrits au paragraphe 5.3.1. Le tableau 5.7 présente les performances des deux passes de notre technique de segmentation appliquée aux mêmes données.

Le TDM et le TFA de la procédure BIC (respectivement 15.7% et 18.3%) et de la deuxième passe de notre algorithme (respectivement 13.5% et 18.5%), obtenus à partir des journaux télévisés de l'INA sont quasiment égaux. Cela signifie que les deux techniques de segmentation sont équivalentes avec des conversations contenant de longs segments de locuteurs. Nous pouvons également remarquer la baisse sensible du TFA entre la première et la seconde passe de notre algorithme : de 37.4% à 18.5%. La segmentation basée sur la distance d_{R} est en fait sensible aux changements d'environnement sonore ou d'intonation du locuteur.

Les conversations téléphoniques (SWITCHBOARD dans les tableaux 5.6 et 5.7) contiennent également de longs segments mais de parole spontanée. En particulier, les conversations téléphoniques sont "clairsemées" de petits mots comme "Yeah" ou "Hum-hum". Quand ces mots sont prononcés alors que l'autre personne parle, notre hypothèse que les personnes ne parlent pas simultanément n'est pas respectée. Le processus de segmentation se trouve détérioré par ces petits mots : ils ne sont en effet pas correctement détectés. Selon l'application, la détection de ces petits mots peut s'avérer non pertinente. A l'inverse, si le niveau de précision requis pour une tâche de transcription automatique est élevé, il devient alors nécessaire de les détecter correctement. Dans le contexte de l'indexation par locuteurs, nous décidons de ne pas les prendre en compte.

La segmentation basée sur la distance (première passe) étant sensible aux changements d'environnements sonores, elle détecte dans la plupart des cas une des bornes de ces petits mots. C'est ce qui explique la valeur élevée du TFA de la première passe : 39.0%. Le TFA reste également plus élevé avec la seconde passe de notre algorithme qu'avec la procédure BIC (respectivement 25.9% contre 20.3%). Par ailleurs, les TDM des deux techniques sont comparables : 29.1% pour DISTBIC et 30.6% pour BIC.

Quant aux conversations contenant de courts segments (TIMIT et CNET dans les tableaux), notre technique de segmentation fournit de meilleurs résultats que la procédure BIC : pour ces conversations, le TDM est deux fois plus faible avec notre technique (15.6% pour TIMIT et 21.4% pour CNET) qu'avec la procédure BIC (30.5% pour TIMIT et 50.0% pour CNET) pour des valeurs de TFA comparables (28.2% pour TIMIT et 16.9% pour CNET with DISTBIC contre 31.5% pour TIMIT et 14.3% pour CNET avec BIC). Les conversations

CNET sont faites de segments plus courts que les conversations TIMIT : cela explique le taux élevé de détections manquées.

Les différences de performances constatées entre les conversations synthétiques (TIMIT et CNET) et les conversations réelles (SWITCHBOARD et INA) avec les deux algorithmes de segmentation s'expliquent par la différence de durée des segments réels de locuteurs. Les conversations synthétiques sont ici constituées de courts segments alors que les conversations réelles sont composées en moyenne de segments plus longs. Dans le cas d'une segmentation idéale, la dernière passe des deux algorithmes estime les modèles de locuteurs (des Gaussiennes multi-dimensionnelles) sur la longueur réelle des segments. Et, plus les segments sont longs, meilleure est l'estimation des modèles (plus fiable et plus robuste) et meilleure est la segmentation obtenue. Dans la pratique, la longueur effective des segments, sur laquelle les modèles de locuteurs sont estimés lors de la dernière passe, est plus importante pour des conversations contenant de longs segments que pour des conversations composées de courts segments. Par conséquent, la modélisation et, de ce fait, la qualité de la segmentation sont meilleures pour les conversations avec de longs segments (nos conversations rélles) que pour les conversations avec de courts segments (nos conversations synthétiques).

Les différences de performances entre les deux types de conversations peuvent aussi s'expliquer par les conditions d'enregistrement. En effet, quand les locuteurs utilisent des canaux de communication différents, la détection des changements de locuteurs est rendue plus aisée car les différences entre locuteurs se trouvent renforcées par les différences de canaux. A l'inverse, dans nos conversations synthétiques (comme TIMIT et CNET), la détection des changements de locuteurs repose uniquement sur les différences de caractéristiques entre locuteurs. En d'autres termes, pour les conversations réelles, les deux algorithmes de segmentation détectent les ruptures entre locuteurs avec leurs conditions d'enregistrement respectives et pour les conversations synthétiques, les algorithmes détectent uniquement les ruptures entre les locuteurs.

Nos expériences montrent que notre technique de segmentation est plus précise que la procédure BIC en présence de segments courts, bien que les deux techniques aient les mêmes performances en présence de segments longs.

5.3.4 Etude qualitative des erreurs de DISTBIC

Nous avons mené d'autres expériences sur des journaux télévisés enregistrés dans notre laboratoire afin d'étudier les occurrences des erreurs. Les résultats sont présentés dans le tableau 5.8. Pour évaluer plus finement notre technique de segmentation, nous définissons le taux de décalages (TD) :

$$\mathrm{TD} = 100 \times \frac{\text{nombre de décalages}}{\text{nbre de changements réels}}\% \tag{5.6}$$

Un décalage est un changement de locuteur qui a été détecté à un instant décalé par rapport à sa position temporelle réelle. Un décalage correspond en fait à une fausse alarme et une détection manquée proches l'une de l'autre et qui ne devrait pas affecter le processus de regroupement. A la suite d'un décalage d'un changement de locuteur, l'un des segments contient les paroles de deux locuteurs. Cependant, la proportion de données d'un des locuteurs (de l'ordre de quelques dixièmes de seconde) est négligeable comparé au volume de données de l'autre locuteur (quelques secondes). L'un des locuteurs reste majoritaire à l'intérieur du segment considéré.

	$1^{ère}$ passe			2^{nde} passe		
	TFA	TDM	TD	TFA	TDM	TD
jt	59.0	8.9	8.4	23.7	9.4	8. 4

TAB. 5.8 – *Journaux télévisés :* TFA, TDM *et* TD *respectivement avec la première et la seconde passes*

La plupart des détections manquées sont dues à de très courtes phrases, surtout durant les interviews : les questions des journalistes sont en général très brèves et elles ne sont pas détectées ou alors partiellement. En fait, les paramètres ont été ajustés pour détecter de longs segments de locuteurs, aussi les segments très courts ne sont pas toujours correctement détectés. Quant au TFA, sa valeur élevée s'explique par deux raisons principales. Tout d'abord, quand une personne de langue étrangère est interviewée et que ses paroles sont traduites simultanément ou plus exactement avec un léger décalage, cela créée des FA. (dans ce cas, notre hypothèse de non-recouvrement entre les différents locuteurs n'est pas respectée). La deuxième raison est liée à la façon dont sont construits les reportages de journaux télévisés : les événements sont commentés mais la bande son correspondant à ces événements reste en fond sonore. Aussi, quand un changement d'environnement sonore intervient dans cette bande son (par exemple, le passage d'une voiture derrière le commentateur), cela provoque bien souvent une fausse alarme.

5.3.5 Conclusion

En résumé, DISTBIC présente l'avantage de détecter des changements de locuteurs proches les uns des autres pour des résultats comparables à la procédure BIC pour des changements plus espacés les uns des autres.

Les paramètres qui interviennent dans DISTBIC ne sont pas encore déterminés de manière systématique. Cependant, ils dépendent essentiellement de la longueur réelle des segments présents dans la conversation et non pas de la nature du signal. En particulier, le critère BIC est désormais utilisé par différents laboratoires sous des formes légèrement différentes, par exemple chez [Liu et al. 99, Gauvain et al. 98]. Néanmoins, ces laboratoires sont également confrontés au problème de la détermination du paramètre λ intervenant dans le critère BIC (chez [Liu et al. 99], ce paramètre s'appelle Θ et chez [Gauvain et al. 99], ce paramètre est transformé en deux autres paramètres α et β).

Par ailleurs, dans la plupart des applications, une sur-segmentation est préférable à une sous-segmentation (dans ce dernier cas, les paroles de deux locuteurs ou plus peuvent être réunies au sein d'un même segment). Aussi, en fonction de l'application choisie, il est peut-être préférable de ne pas effectuer la deuxième passe, qu ivise à réduire le taux de fausses alarmes.

C'est ce que nous faisons dans le contexte des évaluations NIST pour la tâche de poursuite de locuteur (cf Annexe A et *http://www.itl.nist.gov/iaui/894.01/spk_2000/index.html*). Le but est d'obtenir des segments ne contenant les paroles que d'un seul locuteur et d'une durée suffisamment longue pour prendre une décision de vérification du locuteur cible fiable et robuste. La sur-segmentation ne pose donc pas de problème dans ce contexte à condition néanmoins d'obtenir des segments d'une longueur raisonnable.

Il est aussi difficile de comparer les résultats de notre technique de segmentation avec les techniques proposées dans la littérature (à moins bien-sûr de toutes les implémenter). D'une part, parce que les bases de données de parole ne sont pas les mêmes et les résultats sont variables d'une base de données à une autre. D'autre part, les segmentations proposées dans la littérature sont pour la majorité des segmentation en classes acoustiques. Aussi, si deux locuteurs ont des voix proches et des conditions acoustiques comparables alors ils appartiennent à la même classe acoustique. Dans le cadre de la transcription automatique de nouvelles radio- ou télé-diffusées (évaluations DARPA), les données des deux locuteurs réunies serviront à adapter les modèles de parole et le taux de reconnaissance de la parole sera amélioré pour les deux personnes. Le fait de ne pas distinguer les deux locuteurs ne sera alors pas considéré comme une erreur et la segmentation sera jugée de bonne qualité puisqu'elle améliore le taux global d'erreur sur les mots (*WER* ou *Word Error Rate*). Par contre, dans le contexte de l'indexation par locuteurs, cette segmentation ne sera pas jugée d'aussi bonne qualité.

Enfin, une autre source de difficulté pour comparer des techniques de segmentation en locuteurs est qu'il n'existe pas à l'heure actuelle de méthode d'évaluation commune à toutes les équipes de recherche dans ce domaine. Ceci est lié à l'émergence de la discipline et il y a fort à croire que d'ici peu des méthodes standard d'évaluations existeront. Peut-être même qu'elles utiliseront le taux de fausses alarmes et le taux de détections manquées, associés à une tolérance sur les changements détectés, comme le font déjà [Liu et al. 99].

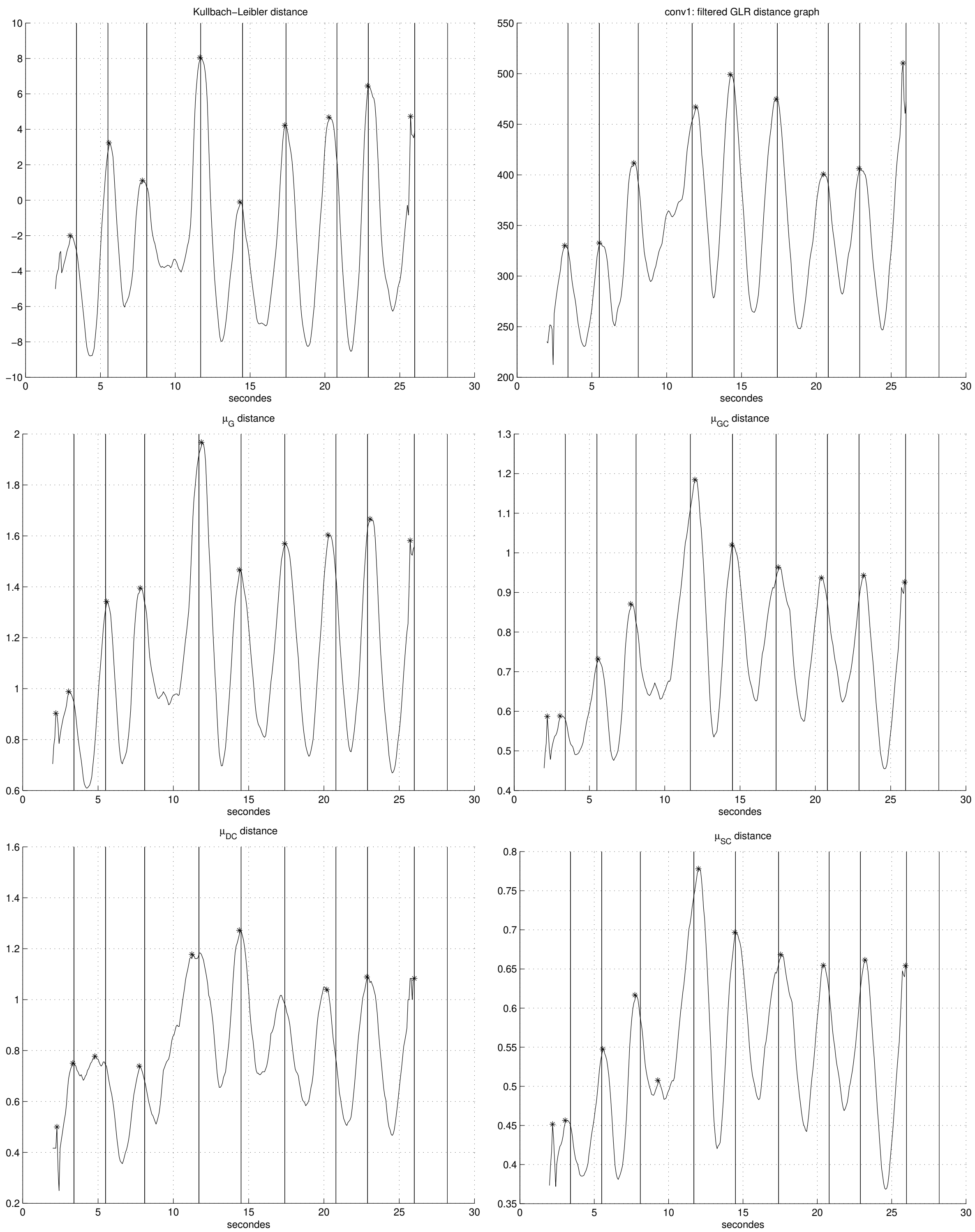

FIG. 5.1 – *Segmentation basée sur le calcul d'une distance. De gauche à droite et de haut en bas : La distance de Kullbach-Leibler, le rapport de vraisemblance généralisé, les mesures de similarité* μ_G, μ_{GC}, μ_{DC} *et* μ_{SC}.

Deuxième partie

Regroupement

Introduction

Après avoir obtenu des segments ne contenant les paroles que d'un seul locuteur, l'étape suivante consiste à regrouper les différents segments appartenant à un même locuteur (cf 2.3). Cette étape de regroupement peut être utile à d'autres applications. Par exemple, elle peut servir à extraire les messages téléphoniques déposés par une même personne sur une messagerie vocale. Dans ce dernier cas, une segmentation en amont n'est pas nécessaire car les messages de différents locuteurs sont distincts les uns des autres (signal sonore entre les messages ou alors longs silences, etc...) Par contre, une difficulté supplémentaire apparaît. Les enregistrements d'une personne, réalisés à partir de différents terminaux (téléphone fixe, téléphone mobile, etc....), doivent être reconnus comme provenant d'un seul et même locuteur. Il faut donc s'affranchir des variations entre les différents canaux de transmission. Nous revenons sur ce point à la section 8.1.1.

Notre problème de regroupement des segments appartenant à un même locuteur est un problème de **classification non-supervisée**. Nous sommes en présence d'une collection d'objets (en l'occurrence des segments ne contenant les paroles que d'un seul locuteur) et nous devons regrouper ces objets par classes, i.e. les locuteurs. Comme nous ne connaissons ni la nature des classes (pas de connaissance a priori sur les locuteurs), ni le nombre de classes (nombre de locuteurs inconnu), nous parlons de classification non-supervisée.

Les problèmes de classification non-supervisés ont été étudiés dans de nombreux domaines, notamment en traitement d'images [Duda et al. 73]. Nous allons nous intéresser aux techniques de **regroupement hiérarchique** (*hierarchical clustering*). Les aspects théoriques de ces techniques sont décrits ci-après. Il peut être intéressant dans le cas de conversations de tenir compte des relations “temporelles” entre les segments. En effet, en cas de sur-segmentation, deux segments contigüs ont de fortes chances d'appartenir à un même locuteur. Les relations de “voisinage” entre segments doivent alors être prises en compte, ce qui n'est pas le cas dans le regroupement hiérarchique. Nous faisons donc appel à des techniques de **regroupement** que nous qualifions de **séquentiel**. Ces techniques sont présentées suite au regroupement hiérarchique. Un autre aspect du regroupement qu'il est important de définir est la manière de déterminer les relations inter-classes. C'est l'objet de la fin de cette introduction.

Regroupement hiérarchique

Dans cette section, nous expliquons le principe du regroupement hiérarchique et définissons le vocabulaire associé, utile pour la suite.

Le but est de classifier un ensemble d'éléments de manière itérative. Nous distinguons deux approches : le **regroupement par agglomération** (*agglomerative* ou *top-bottom clustering*) et le **regroupement par division** (*divisive* ou *top-down clustering*). La première approche

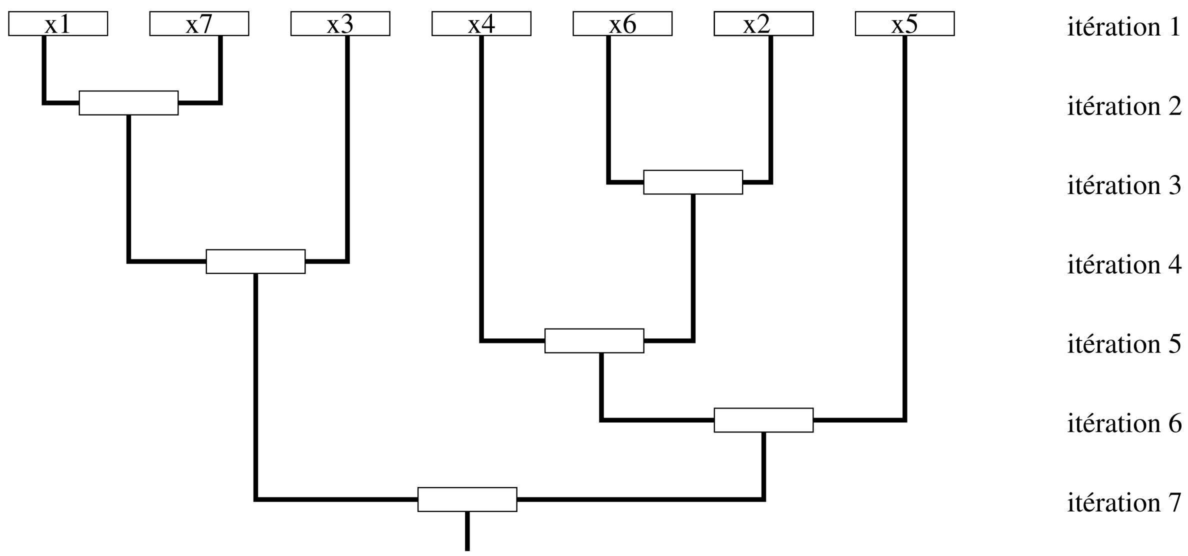

FIG. 6.2 – *Exemple de regroupement par agglomération : exemple de dendrogramme*

considère au début chaque élément comme un groupe ou une classe (*cluster*) et réunit à chaque itération les deux groupes les plus proches au sens d'un critère, appelé **critère de regroupement** (*merging criterion*). Ce processus est répété jusqu'à ce qu'un **critère d'arrêt** (*stopping criterion*) soit satisfait. A l'inverse, le regroupement par division considère au début l'ensemble des éléments comme ne formant qu'un seul groupe et à chaque itération, divise un des groupes selon un critère, appelé **critère de division** (*splitting criterion*). De même, le processus est réitéré jusqu'à ce qu'un critère d'arrêt soit atteint. Le critère d'agglomération étant dans notre contexte plus intuitif que le critère de division, nous considérons, par la suite, uniquement le regroupement par agglomération, qui est illustré à la figure (6.2).

D'après ce que nous venons de voir, deux choses sont importantes à définir pour le regroupement hiérarchique : le critère de regroupement et le critère d'arrêt. Il va de soi que ces critères sont dépendants de l'application envisagée. Ainsi, dans notre contexte, le critère de regroupement est choisi de manière à regrouper des segments ou des groupes de segments appartenant à un seul et même locuteur. Un critère d'arrêt trivial est le nombre final de classes. Or, nous supposons le nombre de locuteurs, donc de classes, inconnu. Deux alternatives se présentent alors à nous.

La première consiste à réitérer l'algorithme jusqu'à obtenir une classe unique. Nous obtenons à l'issue du regroupement un arbre de classification, appelé **dendrogramme**, comme le montre la figure (6.2). C'est la manière de parcourir l'arbre qui définit alors la partition finale.

La deuxième alternative consiste à imposer une contrainte :

– soit sur le critère de regroupement. Par exemple, si le critère de regroupement est une distance et qu'à chaque itération, les deux groupes de segments les plus proches au sens de cette distance sont réunis, nous pouvons imposer la contrainte supplémentaire que cette distance ne doit pas dépasser un certain seuil. Le processus de regroupement s'arrête alors quand les deux groupes de segments les plus proches sont séparés d'une distance ne satisfaisant plus cette contrainte.

– soit sur le groupe de segments résultant du dernier regroupement. Si ce groupe de segments n'est pas jugé suffisamment homogène alors le processus de regroupement est stoppé. Le critère d'homogénéité reste à définir.

Un autre aspect du regroupement hiérarchique qu'il est important de définir est la manière d'actualiser le critère de regroupement au fur et à mesure du regroupement des éléments. L'actualisation du critère de regroupement est lui aussi dépendant de l'application.

Ces aspects seront vus plus en détails dans les chapitres qui suivent.

Regroupement séquentiel

Les méthodes de regroupement hiérarchique ne prennent pas en compte les relations de "voisinage" qui peuvent exister entre les différents éléments à classifier (ou alors par le biais du critère de regroupement). Il peut être intéressant dans certaines applications de tenir compte de ces relations. Par exemple, si nous considérons une conversation téléphonique, il est fort probable que l'intra-variabilité de chaque locuteur soit assez forte. En effet, une conversation téléphonique est en général constituée de paroles spontanées. Supposons que le regroupement hiérarchique compare des segments appartenant à un même locuteur mais que l'un des segments se situe au début de la conversation et l'autre à la fin de la conversation. Le regroupement de ces deux segments risque alors d'échouer à cause de la forte intra-variabilité. Il est intéressant dans ce cas de considérer les segments séquentiellement (par ordre temporel) car la variabilité entre deux segments proches temporellement, appartenant à un même locuteur, est moins forte qu'entre deux segments éloignés temporellement, appartenant à ce même locuteur.

Par ailleurs, pour une application temps réel, il est important de pouvoir traiter les segments au fur et à mesure et non d'effectuer le regroupement une fois tous les segments collectés. C'est donc là une autre source d'application des algorithmes de regroupement séquentiel.

Le principe du regroupement séquentiel est expliqué à la figure (6.3). A la première itération, le premier élément va former la première classe. A chaque itération, l'élément suivant (au sens du voisinage) est examiné. S'il satisfait le critère de regroupement avec les classes (groupes d'éléments) existantes alors il est assigné au groupe d'éléments le plus proche au sens du critère de regroupement. A l'opposé, si aucun des groupes d'éléments ne convient, alors une nouvelle classe est créée. Ceci implique que le critère de regroupement soit contraint pour que de nouvelles classes puissent être créées. Le critère d'arrêt est ici très simple : le processus s'arrête quand l'ensemble des éléments a été examiné. Enfin, l'actualisation du critère de regroupement au fur et à mesure du regroupement est aussi un point crucial pour le regroupement séquentiel.

Détermination des relations inter-classes

A l'origine du regroupement agglomératif ou séquentiel, le critère de regroupement est estimé pour des couples d'éléments. Puis, au fur et à mesure du regroupement, des classes d'éléments se forment. La question est alors de savoir comment le critère de regroupement est estimé entre deux classes.

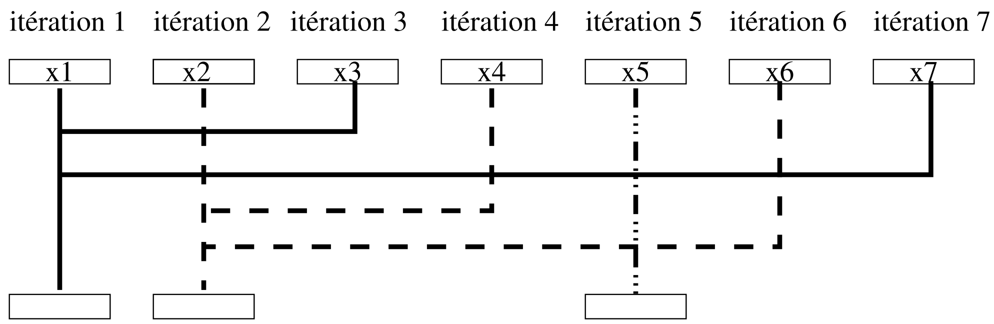

FIG. 6.3 – *Exemple de regroupement séquentiel*

Plusieurs alternatives sont possibles. Elles sont exposées dans [Solomonoff et al. 98]. Il faut donc définir le critère c entre deux classes d'éléments, respectivement G_k et G_l. Voici quelques unes des possibilités :

- **estimation par paire minimale** (*minimum pair/single linkage*) : le critère de regroupement entre deux classes d'éléments est défini comme le critère minimal séparant deux éléments, chacun appartenant respectivement à une des classes

$$c(G_k, G_l) = \min_{i \in G_k, j \in G_l} c(i, j)$$

- **estimation par paire maximale** (*maximum pair/complete linkage*) : le critère de regroupement entre deux classes d'éléments est défini comme le critère maximal séparant deux éléments, chacun appartenant respectivement à une des classes

$$c(G_k, G_l) = \max_{i \in G_k, j \in G_l} c(i, j)$$

- **estimation par paire moyenne** (*average pair/linkage*) : le critère de regroupement entre deux classes d'éléments est défini comme le critère moyen séparant deux éléments, chacun appartenant respectivement à une des classes. N_k et N_l désignent le nombre d'éléments respectivement de G_k et de G_l dans la formule ci-dessous.

$$c(G_k, G_l) = \frac{\sum_{i \in G_k, j \in G_l} c(i, j)}{N_k N_l}$$

- **estimation "complète"** (*full linkage*) : une autre solution consiste à considérer une classe d'éléments comme ne formant qu'un seul et même élément (dans notre cas, un groupe de segments sera considéré comme un seul segment obtenu en concaténant tous les segments du groupe). Les caractéristiques de cette classe, vue comme un élément, sont alors recalculées à chaque nouvel ajout d'élément, ainsi que les critères de regroupement impliquant cette classe. Cette solution implique un coût calculatoire loin d'être négligeable.

Après avoir présenté les deux types de regroupement que nous allons considérer par la suite, nous allons tout d'abord passer en revue les travaux existant dans la littérature sur l'application de ces algorithmes au regroupement de segments au chapitre 7. A partir de cet état de l'art, nous présentons au chapitre 8 l'algorithme de regroupement hiérarchique retenu. Le chapitre 9 présente tout d'abord les méthodes d'évaluation qui nous permettent de juger des performances du regroupement hiérarchique testé. La méthode de regroupement retenue est ensuite testée et les performances obtenues sur différents types de données sont exposées. Une étude des paramètres intervenant dans ce regroupement hiérarchique est également réalisée. Enfin, nous concluons la partie consacrée au regroupement des segments par locuteurs.

Chapitre 7

Etat de l'art

Munis du formalisme exposé au chapitre précédent, nous présentons dans ce chapitre des méthodes de regroupement des segments appartenant à un seul et même locuteur existant dans la littérature. Comme au précédent chapitre, nous nous intéressons aux techniques de regroupement hiérarchique par agglomération (section 7.1) puis aux algorithmes de regroupement séquentiel (section 7.2).

De même que pour la segmentation en locuteurs (cf chapitre 3), l'essentiel des travaux présentés ici ont été réalisés dans le cadre des évaluations DARPA de systèmes de transcription automatique de journaux radio- ou télé-diffusés. Nous avons vu que l'adaptation au locuteur améliore significativement les performances des systèmes de reconnaissance de la parole sur de grands vocabulaires. En général, peu de données du locuteur suffisent pour l'adaptation. Cependant, dans le cas d'une adaptation non supervisée, il est préférable d'utiliser une quantité de données plus importante. Dans ce contexte, l'objectif est ici de regrouper les segments (issus de la segmentation par exemple) en groupes homogènes, c'est-à-dire des groupes ne contenant qu'un seul et même locuteur dans les mêmes conditions d'enregistrement, afin d'avoir suffisamment de données de ce locuteur. Ces données servent ensuite à adapter au locuteur considéré des modèles de parole pré-entraînés dans le but d'améliorer les taux de reconnaissance. Plus il y a de données disponibles pour un locuteur donné, meilleure est l'adaptation. Il faut bien souvent faire un compromis entre l'homogénéité au sein des groupes de segments et leur taille, qui doit être suffisamment importante pour une meilleure robustesse des techniques d'adaptation. Les évaluations DARPA pour les systèmes de transcription automatique utilisent la base de données HUB-4 qui regroupe des journaux radio- et télé-diffusés américains (cf introduction de la partie I).

7.1 Regroupement hiérarchique par agglomération

Dans cette section, nous présentons divers critères de regroupement possibles, ainsi que des crières d'arrêt qui leur sont associés au travers de différents travaux.

7.1.1 Utilisation du rapport de vraisemblance généralisé et nombre de classes connu

Ce paragraphe présente un regroupement hiérarchique avec comme critère de regroupement la distance basée sur le rapport de vraisemblance défini à l'équation (2.3). Comme vu précédemment, plus cette distance est petite, plus il est probable que les segments aient été générés par le même locuteur. Le principe du regroupement est le suivant : une matrice de distances D est formée où chaque élément de la matrice $d(i,j)$ représente la distance entre les segments i et j (i et j variant de 1 à N, avec N le nombre total de segments). A chaque itération, les deux segments ou groupes de segments les plus proches au sens de cette distance sont réunis.

[Gish et al. 91] utilisent ce regroupement pour séparer pilotes et contrôleurs aériens. Ces travaux ont été déjà évoqués au chapitre 2 à la section 2.1.4.

Pour réévaluer la distance entre deux groupes de segments, [Gish et al. 91] font le choix de la distance par paire maximale. Par ailleurs, le critère d'arrêt est trivial : comme les segments doivent être regroupés en deux classes, le processus s'arrête quand le nombre de groupes de segments est égal à deux. Les communications du contrôleur étant prépondérantes (en termes de temps de parole), le groupe contenant le plus de segments est alors identifié comme celui du contrôleur.

Des tests sur 423 segments, dont 220 de contrôleurs et 203 de pilotes ont abouti à 6 erreurs de classification avec cette méthode de regroupement. Cependant, il ne faut pas oublier que dans le cadre de cette application, les pilotes et le contrôleur aérien n'utilisent pas le même canal de transmission et de ce fait se distinguent assez facilement.

Le rapport de vraisemblance généralisé nous semble être un bon critère de regroupement, les résultats viennent d'ailleurs le confirmer. Néanmoins, dans cette application, le critère d'arrêt est simple puisque le nombre final de classes est connu, ce qui n'est pas notre cas. Nous ne pouvons donc appliquer cet algorithme tel quel.

7.1.2 Utilisation du rapport de vraisemblance et du critère de dispersion

Le regroupement hiérarchique traité dans ce paragraphe utilise également comme critère de regroupement entre deux segments ou deux groupes de segments le rapport de vraisemblance généralisé (cf 7.1.1). De la même manière, une matrice de distances est formée. De plus, en remarquant que des segments consécutifs ont de fortes chances d'appartenir à un même locuteur si le taux de fausses alarmes est élevé, la distance entre deux segments est alors pondérée par un paramètre α qui tend à favoriser le regroupement des segments consécutifs.

Le nombre de locuteurs n'étant pas a priori connu, le regroupement hiérarchique est effectué jusqu'à n'obtenir plus qu'un seul groupe de segments. Il s'agit alors de choisir une partition de segments à partir du dendrogramme obtenu. Pour ce faire, un critère de dispersion est défini. Une bonne partition sera caractérisée par une faible dispersion au sein de chaque classe de segments (i.e. chaque locuteur). La matrice de dispersion est définie par :

$$W_{\alpha,k} = \sum_{j=1}^{k} N_j \times \Sigma_j \tag{7.1}$$

où Σ_j est la matrice de covariance des N_j vecteurs acoutiques contenus dans les segments du groupe G_j. k est le nombre de groupes de segments dans la partition. L'index α dans l'équation (7.1) rappelle que le dendrogramme obtenu dépend de ce paramètre. Enfin, la dispersion est définie comme suit :

$$\text{dispersion} \;\; = |W_{\alpha,k}| \tag{7.2}$$

où $|\quad|$ désigne le déterminant.

Pour un nombre k donné de groupes de segments, ce dendrogramme est élagué de manière à obtenir k feuilles (i.e groupes de segments). Pour une liste (non exhaustive) de combinaisons (α, k), la procédure d'élagage fournit une liste de partitions potentielles parmi lesquelles l'une d'elles est choisie en minimisant le critère de dispersion $|W_{\alpha,k}|$.

Cependant, ce critère conduit en pratique à la solution extrême d'un segment par locuteur. Pour éviter cela, une pénalité est introduite contre un trop grand nombre de groupes de segments dans la partition. Le critère avec pénalité devient alors la minimisation de :

$$|W_{k,\alpha}| \times \sqrt{k} \tag{7.3}$$

Le but du regroupement hiérarchique dans les travaux de ([Jin et al. 97]) est de regrouper les segments appartenant à un même locuteur. Les auteurs entendent par locuteur un locuteur avec les mêmes conditions de transmission (même canal) et d'enregistrement. Cependant, ce regroupement étant destiné à l'adaptation des systèmes aux locuteurs, il n'est pas gênant de regrouper des locuteurs ayant les mêmes caractéristiques acoustiques. L'algorithme de regroupement hiérarchique améliore les performances du système de transcription automatique en termes de reconnaissance de mots. Il est testé sur des données provenant de la base de données HUB-4. Par contre, le nombre de groupes de segments, et donc de locuteurs, est souvent inférieur au nombre réel de locuteurs.

En effet, l'algorithme a tendance à regrouper les locuteurs qui ont les mêmes caractéristiques acoustiques, ce qui n'est pas gênant pour l'adaptation au locuteur mais peut constituer un obstacle dans le cadre de l'indexation par locuteurs. Le paramètre α joue un rôle essentiel selon les auteurs pour le regroupement. Cependant, ils n'en précisent ni la valeur, ni la façon dont il est choisi.

Pour d'autres applications où la distinction entre locuteurs est indispensable, [Jin et al. 97] suggèrent d'examiner d'autres critères de sélection d'une partition et pénalités, parmi lesquels :

$$|W_{k,\alpha}| + C \times \sqrt{k}$$

ou

$$|W_{k,\alpha}| + C \times \log k$$

où C est une constante.

Cependant ces critères ne semblent pas avoir de fondements théoriques. Par contre, ces critères ne sont pas dépendants du critère de regroupement utilisé donc ils peuvent s'appliquer avec d'autres distances.

7.1.3 Utilisation du rapport de vraisemblance et du critère de pureté

Ce paragraphe présente un algorithme de regroupement hiérarchique analogue au précédent : il est basé sur le rapport de vraisemblance généralisé comme critère de regroupement. L'algorithme s'arrête quand il n'y a plus qu'un seul groupe de segments. C'est la sélection de la partition dans le dendrogramme obtenu qui diffère. Cette sélection repose sur un critère de pureté défini ci-après. Ce même critère de pureté peut également servir à l'évaluation de la partition obtenue.

Une partition est parfaite si chaque groupe de segments ne contient les segments que d'un seul locuteur et si tous les segments de ce locuteur sont tous réunis dans le même groupe de segments. La méthode d'évaluation doit donc pénaliser la réunion des paroles de différents locuteurs ou à l'inverse, la séparation des paroles d'un même locuteur en plusieurs groupes de segments.

Soit n_{ij} le nombre de segments de parole du locuteur j dans le groupe de segments i. Soit N_L le nombre total de locuteurs, N_G le nombre de groupes de segments et N_S le nombre de segments de parole. Enfin, $n_{.j} = \sum_{i=1}^{N_G} n_{ij}$ désigne le nombre total de segments du locuteur j et $n_{i.} = \sum_{j=1}^{N_L} n_{ij}$ la taille du groupe de segments en nombre de segments.

La première mesure d'évaluation proposée est le *Rand Index*:

$$I_{\text{RAND}} = \frac{1}{2}\left\{\sum_i n_{i.}^2 + \sum_j n_{.j}^2\right\} - \sum_i \sum_j n_{ij}^2 \tag{7.4}$$

I_{RAND} mesure le nombre de paires de segments qui appartiennent à un même locuteur mais qui ne sont pas dans le même groupe ou les paires de segments qui sont dans le même groupe mais qui n'appartiennent pas au même locuteur. Plus I_{RAND} est faible, meilleure est la partition.

La deuxième mesure d'évaluation proposée est développée par BBN et est appelée métrique BBN:

$$I_{\text{BBN}} = \sum_i \sum_j \frac{n_{ij}^2}{n_{i.}} - Q.N_G \tag{7.5}$$

où Q est un paramètre à ajuster. Il permet de favoriser ou non de larges groupes quitte à réunir des segments n'appartenant pas à un même locuteur. Plus la valeur de I_{BBN} est élevée, meilleure est la partition.

La pureté p_i d'un groupe de segments i est définie comme suit :

$$p_i = \sum_j \frac{n_{ij}^2}{n_i^2} \tag{7.6}$$

La pureté d'un groupe de segments représente le degré d'appartenance des segments à un même locuteur. La pureté p_i représente aussi la probabilité que deux segments choisis aléatoirement dans le groupe de segments i proviennent du même locuteur. En introduisant la notion de pureté, les formules (7.4) et (7.5) deviennent :

$$I_{\text{RAND}} = \sum_i n_{i.}^2 (\frac{1}{2} - p_i) + \frac{1}{2}\sum_j n_{.j}^2 \tag{7.7}$$

et

$$I_{\text{BBN}} = \sum_i n_{i.} p_i - Q \times N_G \tag{7.8}$$

Les partitions contenant de larges groupes de segments impurs obtiennent de meilleurs scores de I_{RAND} et I_{BBN} que celles contenant de petits groupes de segments purs (ne contenant que les segments d'un même locuteur).

En connaissant la pureté et la taille (i.e. le nombre de segments) de chaque groupe de segments, il est facile d'évaluer une partition. Cependant, autant la taille des groupes de segments est connue, autant la connaissance de la pureté nécessite de connaître pour chaque segment l'identité du locuteur auquel il se rapporte. Une estimation de la pureté est donc proposée. Elle repose sur les plus proches voisins (*nearest neighbors*). Cette pureté est d'abord estimée pour chaque segment du groupe et la pureté du groupe de segments est définie comme la moyenne des puretés des segments qu'il contient. La pureté d'un segment k appartenant à un groupe de segments G_i est estimée comme suit :

1. trier tous les segments par ordre croissant de distance par rapport au segment k
2. prendre les $n_{i.}$ plus proches segments et compter parmi ces segments ceux appartenant effectivement au groupe G_i. Soit n_{group} ce nombre.
3. définir la pureté ρ_k du segment k, comme la fraction des premiers $n_{i.}$ plus proches voisins qui sont dans le groupe G_i :
$$\rho_k = \frac{n_{group}}{n_{i.}} \tag{7.9}$$

La pureté du groupe de segments est alors définie comme :

$$\hat{p}_i = \frac{1}{n_{i.}} \sum_{k \in G_i} \rho_k \tag{7.10}$$

Plus la partition obtenue se rapproche de la partition idéale (i.e. la partition réelle), plus cet estimateur de pureté tend vers la pureté théorique définie à l'équation (7.6).

La pureté ainsi estimée peut également servir à opérer un regroupement hiérarchique contraint : la pureté est alors utilisée comme critère d'arrêt.

Le rapport de vraisemblance généralisé comme critère de regroupement et la pureté comme critère de sélection à partir du dendrogramme sont proposés par [Solomonoff et al. 98] pour le regroupement hiérarchique. L'utilisation de la distance de Kullbach-Leibler (3.1) comme critère de regroupement est également envisageable car elle ne nécessite pas la modélisation de la réunion des deux segments ou groupes de segments considérés contrairement au rapport de vraisemblance généralisé. Pour calculer des distances et former la matrice de distances, chaque segment est modélisé par une mixture de Gaussiennes avec des matrices de covariance diagonales, entraînées par l'algorithme EM (Expectation-Maximisation [Dempster et al. 77]). Le regroupement hiérarchique est réalisé jusqu'à n'obtenir plus qu'un seul groupe (la distance entre deux groupes de segments est réestimée à chaque itération).

Pour sélectionner une partition à partir du dendrogramme obtenu, [Solomonoff et al. 98] procèdent de la manière suivante. Ils coupent le dendrogramme horizontalement de manière à obtenir que quelques groupes de segments, genre 5 ou 10. Chacun de ces groupes de segments correspond à un sous-arbre du dendrogramme. Un découpage de ce sous-arbre qui optimise la pureté peut être trouvé. Ainsi la partition finale est la réunion de toutes les partitions des sous-arbres. Cependant, cette méthode ne garantit pas l'optimisation de la pureté globale.

Les auteurs ont mené des expériences sur des données de SWITCHBOARD. Les 60 segments durent une minute chacun et émanent de 20 locuteurs (3 segments par locuteurs) moitié hommes, moitié femmes. Pour chaque locuteur, deux des enregistrements sont réalisés sur le même canal, le troisième sur un canal différent. Les résultats montrent que l'estimateur n'est mauvais que pour peu de groupes de segments, et que plus le nombre de groupes augmente, meilleur devient l'estimateur. La partition choisie avec le meilleur score estimé (i.e. la pureté estimée) contient 29 groupes de locuteurs au lieu des 20 locuteurs attendus. Parmi les 29 groupes, seulement 2 sont contaminés, i.e. contiennent des segments générés par des locuteurs différents.

Cependant, il faut quand même noter que ces expériences ont lieu sur une collection de 60 segments, ce qui est peu pour une conversation réelle, et surtout que chaque segment dure une minute, ce qui permet une bonne modélisation pour mesurer les distances entre groupes de segments. Par ailleurs, [Solomonoff et al. 98] signalent eux-mêmes que leur méthode de sélection de la partition à partir du dendrogramme ne garantit pas d'avoir la partition optimale.

Par ailleurs, comme au paragraphe précédent, la pureté est indépendant du critère de regroupement utilisé. Il peut également s'appliquer comme critère d'arrêt et non plus comme critère de sélection à partir du dendrogramme.

7.1.4 Utilisation du rapport de vraisemblance croisé et d'un critère d'efficacité

Le critère de regroupement considéré dans ce paragraphe est le rapport de vraisemblance croisé. Chaque segment de parole m_j est modélisé par un GMM λ_j (*Gaussian Mixture Model* [Reynolds 95]). Ce GMM est obtenu à partir d'un GMM du monde λ_B indépendant du locuteur (*speaker-independant background model*, cf [Reynolds 97]) pré-entraîné à l'aide d'autres données de parole. Le rapport de vraisemblance croisé d_{ij} entre les messages m_i et m_j est alors défini par :

$$d_{ij} = \log \frac{l(m_i|\lambda_B)}{l(m_i|\lambda_j)} + \log \frac{l(m_j|\lambda_B)}{l(m_j|\lambda_i)} \tag{7.11}$$

où $l(m_i|\lambda_j)$ représente la vraisemblance du segment m_i selon le modèle λ_j. Une matrice de distances D dont les éléments sont les d_{ij} est formée. A chaque itération, les groupes de segments les plus proches au sens de cette distance sont réunis. L'algorithme s'achève quand il n'y a plus qu'un seul groupe de segments. Il faut alors définir un critère de sélection d'une partition à partir du dendrogramme obtenu.

Le critère de sélection s'appuie sur la mesure I_{BBN} définie à l'équation (7.5). Soient $I_{BNN}(C)$ la mesure d'évaluation pour la partition sélectionnée, $I_{BNN}(P)$ la mesure d'évaluation pour la partition parfaite et enfin $I_{BNN}(S)$ la mesure d'évaluation pour la partition ne comprenant que des singletons (en d'autres termes, aucun regroupement n'est effectué). L'efficacité η du regroupement est alors définie comme suit :

$$\eta = \frac{I_{BNN}(C) - I_{BNN}(S)}{I_{BNN}(P) - I_{BNN}(S)} \tag{7.12}$$

Une partition parfaite obtient un score de 1.0 alors que la partition ne contenant que des singletons obtient un score de 0.0. Cependant, un regroupement hiérarchique aboutissant à un seul groupe de segments peut obtenir un score négatif selon la valeur de Q intervenant dans la définition de I_{BNN}. La pureté des groupes de segments intervenant dans la mesure I_{BBN} est estimée par la méthode proposée par [Solomonoff et al. 98] et détaillée au paragraphe 7.1.3 équation 7.10.

Le dendrogramme résultant d'un regroupement hiérarchique de N objets offre $2N - 1$ groupes d'objets. Deux méthodes sont proposées pour sélectionner la meilleure partition. La première méthode, la plus basique, consiste à couper le dendrogramme horizontalement et la partition retenue est celle dont la mesure I_{BBN} est maximale. Cependant, en général, la partition parfaite (réelle) ne correspond pas à un découpage horizontal du dendrogramme. Cependant, examiner toutes les combinaisons possibles à partir d'un dendrogramme est une tâche plus que coûteuse en temps de calculs !

La deuxième méthode proposée est itérative. Elle consiste à sélectionner dans le dendrogramme le nœud qui réalise le meilleur score $\hat{p}_i - \frac{Q}{n_i}$. Les nœuds-fils et parents sont alors supprimés de l'arbre et le processus recommence jusqu'à avoir traité tous les nœuds du dendrogramme. De même, cette méthode conduit à un temps de calculs important.

Une dernière méthode, appelée d^*, sélectionne une partition non plus à partir du dendrogramme mais à partir de l'ensemble des segments. Il ne s'agit plus de regroupement hiérarchique à proprement parler. Le principe de l'algorithme est le suivant : chaque segment est considéré comme le centroïde d'une hypersphère de rayon d. Cette hypersphère forme un groupe de segments pour lequel le score $\hat{p}_i - \frac{Q}{n_i}$ est calculé. $\hat{p}_i$ représente la pureté estimée du groupe de segments par la méthode des plus proches voisins (cf section 7.1.3 équation (7.10)). Le segment qui obtient le meilleur score et les segments qui sont inclus dans l'hypersphère correspondante sont retenus pour former un groupe de segments de la partition finale. Tous ces segments ne sont plus considérés par la suite. Le processus est réitéré jusqu'à ce que tous les segments aient été traités et appartiennent à un groupe de segments de la partition finale.

Une première version de l'algorithme d^* repose sur une valeur fixe optimale de d choisie a priori. Une deuxième version de l'algorithme utilise plusieurs valeurs de d : c'est-à-dire que le score maximum est désormais recherché pour tous les segments et toutes les valeurs de d considérées. d prend ses valeurs dans l'intervalle $[d_{\min}, d_{\max}]$, avec $d_{\min} = 0.1\%$ et $d_{\max} = 10\%$ de la distribution des distances et dix autres valeurs de rayon possibles, régulièrement espacées dans l'intervalle.

[Reynolds et al. 98] utilisent ces méthodes pour regrouper des segments de parole selon l'identité du locuteur. Les expériences sont menées sur des données issues de SWITCHBOARD. Elles consistent en 1369 messages de 30 secondes prononcés par 225 hommes et 172 femmes. Le nombre de messages par locuteur va de 1 à 27. La deuxième version de l'algorithme d^* présente de meilleures performances ($\eta = 0.611$) que le regroupement hiérarchique utilisant un découpage horizontal du dendrogramme ($\eta = 0.509$) et que le regroupement hiérarchique avec une recherche séquentielle du meilleur score ($\eta = 0.574$).

L'utilisation du rapport de vraisemblance croisé suppose un modèle du monde, qui doit être entraîné sur un gros volume de données de parole (de l'ordre de quelques heures). Cette distance n'est a priori pas envisageable dans notre contexte puisque nous faisons l'hypothèse que nous n'avons aucune connaissance a priori. Cependant, en supposant que les données de parole à indexer représente quelques heures, nous pouvons envisager l'apprentissage de ce modèle du monde (cf [Matsui et al. 95]).

Par ailleurs, nous remarquons comme au paragraphe précédent que l'emploi de *GMMs* pour modéliser chaque segment n'est possible que si ces segments sont suffisamment longs. C'est le cas dans les expériences menées ci-dessus car chaque segment est d'une durée de 30 secondes en moyenne. Par contre, ces modèles ne sont pas recommandés avec des segments de quelques secondes car ils ne sont pas estimés de manière fiable étant donné le faible volume de données de parole.

Enfin, les auteurs présentent leurs résultats en terme d'efficacité mais ne détaillent pas le nombre de groupes de segments obtenus et la pureté des groupes. A notre avis, l'efficacité ne suffit pas à déterminer la qualité de la partition obtenue. De plus, l'efficacité fait intervenir le paramètre Q par le biais de I_{BBN} (cf équation 7.5). Ce paramètre Q sert à favoriser ou non de larges groupes de segments. Aucune information n'est donnée sur la valeur optimale de ce paramètre, s'il est variable en fonction des bases de données, etc...

7.1.5 Utilisation de la distance de Mahalanobis ou de Kullbach-Leibler et d'un seuil

Dans ce paragraphe, la distance de Mahalanobis ([Duda et al. 73] page 24) et la distance de Kullbach-Leibler (cf 3.1) sont testées comme critères de regroupement pour le regroupement hiérarchique. Ces distances sont contraintes par un seuil, i.e. le regroupement hiérarchique s'arrête quand la distance entre les deux groupes de segments les plus proches dépasse ce seuil.

[Siegler et al. 97] utilisent cet algorithme de regroupement dans le contexte de la transcription automatique de journaux radio- ou téle-́diffusés. Le regroupement hiérarchique vise à regrouper les segments d'un même locuteur enregistrés dans les mêmes conditions acoustiques pour ensuite adapter à ce locuteur des modèles de parole pré-entraînés. Le seuil utilisé pour contraindre le regroupement de deux groupes de segments doit être suffisamment bas pour que les groupes de segments résultant ne contiennent qu'un seul et même locuteur. Dans le cas contraire (erreur de fausse alarme), cela peut contribuer à la dégradation des performances de l'adaptation consécutive si les locuteurs réunis ont des caractéristiques acoustiques éloignées. A l'inverse, ce seuil doit être suffisamment élevé pour ne pas obtenir trop de groupes de segments d'un même locuteur. Le premier cas d'erreur (fausse alarme) étant plus préjudiciable à l'adaptation, le seuil est choisi de manière à minimiser ce type d'erreur.

La distance de Kullbach-Leibler fournit de meilleurs résultats que la distance de Mahalanobis. Le seuil est apparemment choisi fixe : les valeurs varient de 0.020 à 0.066 pour la distance de Kullbach-leibler. En parallèle, la probabilité de fausses alarmes évoluerait respectivement de 0.1% à 1.7% ([Siegler et al. 97] ne définissent pas clairement ce taux) pour des tailles de groupes de segments variant respectivement de 32.6s à 36.6s. Les performances de ce regroupement hiérarchique sont jugées par rapport au taux d'erreurs pour la reconnaissance automatique des mots. En l'occurrence, elles sont jugées assez bonnes car elles contribuent à l'amélioration du taux de reconnaissance sur 4 journaux issus de la base de données HUB-4.

Ce regroupement hiérarchique est également utilisé par les chercheurs de la société Philips [Harris et al. 99].

La distance de Kullbach-Leibler nous semble être un critère de regroupement prometteur. Elle est d'ailleurs utilisée par d'autres laboratoires (éventuellement sous des noms différents, ce que nous verrons dans la suite de ce chapitre). Par contre, utiliser un seuil fixe pour contraindre ce critère de regroupement ou tout autre nous paraît plus criticable. En effet, il n'est pas dit que la valeur de ce seuil ne varie pas d'une base de données de parole à une autre, et même au sein d'un même enregistrement. Utiliser un seuil adaptatif nous paraît plus judicieux.

7.1.6 Utilisation de l'entropie relative et de l'entropie

Le critère de regroupement utilisé dans ce paragraphe est l'entropie relative entre deux segments $X = \{x_1, ..., x_{N_1}\}$ et $Y = \{y_1, ..., y_{N_2}\}$ qui est définie par :

$$D(X,Y) = \frac{1}{N_1} \sum_{i=1}^{N_1} \log \frac{p(x_i, M_1)}{p(x_i, M_2)} + \frac{1}{N_2} \sum_{j=1}^{N_2} \log \frac{p(y_j, M_2)}{p(y_j, M_1)} \tag{7.13}$$

où M_1 et M_2 sont les modèles statistiques sous-jacents de X et de Y. L'entropie relative n'est autre que la distance de Kullbach-Leibler symétrisée (cf équation 4.6). Ce résultat se retrouve dans [Cover et al. 91].

Les modèles utilisés sont des GMMs (*Gaussian Mixture Models*). La distance entre deux groupes de segments est par paire maximum. Le critère d'arrêt choisi est basée sur cette distance : si elle est inférieure à un certain seuil, déterminé empiriquement, alors les deux groupes de segments sont fusionnés.

L'homogénéité d'un groupe de segments, peut être évaluée à l'aide de l'entropie qui est définie comme suit :

$$H = -\frac{1}{N_C} \sum_{i=1}^{N_C} \sum_{j=1}^{N_{S_i}} P(S_j|C_i)\ log P(S_j|C_i) \tag{7.14}$$

où N_C est le nombre de groupes de segments, N_{S_i} le nombre de locuteurs dans le groupe de segments i, $P(S_j|C_i)$ la probabilité conditionnelle du locuteur j dans le groupe de segments i et $-P(S_j|C_i)\ log P(S_j|C_i)$ représente l'entropie du groupe de segments i.

En comparant les regroupements hiérarchiques obtenus respectivement pour une distance par paire minimum entre groupes de segments et pour une distance par paire maximum, le deuxième fournit des groupes de segments plus homogènes en termes d'entropie.

[Heck et al. 97] évalue cet algorithme de regroupement hiérarchique décrit dans le contexte des évaluations DARPA des systèmes de transcription automatique de journaux radio- ou télé-diffusés. Le but est de regrouper les segments fournis par la segmentation de référence en ensembles homogènes (même environnement d'enregistrement, même locuteur) de manière à avoir un volume de données plus important pour l'adaptation des modèles. Pour évaluer l'efficacité du regroupement hiérarchique, les auteurs comparent les résultats de reconnaissance après adaptation des modèles obtenus d'une part avec la segmentation de référence fournie dans le cadre des évaluations DARPA et d'autre part avec les groupes de segments issus du regroupement hiérarchique. L'amélioration due au regroupement hiérarchique est de 6.3%.

A nouveau, nous mettons en cause l'utilisation d'un seuil (fixe ?) déterminé empiriquement. Par ailleurs, le regroupement étant effectué dans le cadre des évaluations DARPA, seule l'amélioration du taux de reconnaissance par mot importe. Rien ne garantit que les locuteurs soient correctement séparés.

7.1.7 Utilisation de la mesure de divergence et de la configuration des groupes de segments

Le critère de regroupement évalué dans ce paragraphe est la mesure de divergence Gaussienne :

$$d(X,Y) = \frac{1}{2}\mathrm{tr}(\Sigma_x^{-1}\Sigma_y + \Sigma_y^{-1}\Sigma_x - 2I) + \frac{1}{2}(\mu_x - \mu_y)^T(\Sigma_x^{-1} + \Sigma_y^{-1})(\mu_x - \mu_y) \tag{7.15}$$

où μ_x et Σ_x représentent respectivement le vecteur moyen et la matrice de covariance supposée diagonale du segment X. I est la matrice identité. Nous montrons en annexe B que la mesure de divergence Gaussienne n'est autre que la distance de Kullbach-Leibler pour des modèles Gaussiens.

Le critère d'arrêt porte sur la configuration des groupes de segments résultants : l'algorithme de regroupement s'arrête quand le plus petit des groupes de segments contient un nombre de données de parole préalablement fixé.

[Hain et al. 98a] proposent cette méthode de regroupement hiérarchique et la compare à la méthode de [Siegler et al. 97] (cf 7.1.5). Les seuils des deux méthodes sont ajustés de manière à fournir le même nombre final de groupes de segments. Pour évaluer les performances de chaque méthode, les performances réalisées par le système après adaptation des modèles sont examinées. Les deux méthodes se révèlent efficaces d'une part pour fournir des segments homogènes et d'autre part pour augmenter le taux de reconnaissance par rapport à la segmentation de référence fournie dans le contexte des évaluations DARPA.

Ce dernier résultat peut sans doute s'expliquer par le fait que les auteurs ajustent les paramètres des deux méthodes de manière à obtenir le même nombre de groupes de segments. Par ailleurs, les deux critères de regroupement s'appuyant tous sur la matrice de covariance et finalement mesurant la (dis)similarité entre les matrices de covariance, il y a de fortes chances pour que les groupes de segments obtenus dans les deux cas soient semblables.

Le critère d'arrêt nous paraît peu adapté pour le regroupement en locuteurs. En effet, ce critère ne permet pas de partition avec des singletons. Or, c'est ce qui devrait se produire si dans une conversation, un locuteur n'intervient qu'une seule fois.

[Hain et al. 98a] développent une troisième méthode de regroupement par fusion/division que nous ne détaillons pas. D'autres méthodes de fusion/division se trouvent dans [Johnson et al. 98, Johnson 99].

7.1.8 Utilisation du Critère d'Information Bayésien (BIC)

Le critère BIC a été étudié aux sections 3.2.3 et 4.2.4 pour la segmentation en locuteurs par détection de changements de locuteurs.

Soit $\mathcal{G}_k = \{G_i : i = 1, ..., k\}$ une partition contenant k groupes de segments. Chaque groupe est modélisé par une distribution Gaussienne multi-dimensionnelle $N(\mu_i; \Sigma_i)$ où μ_i est le vecteur moyen et Σ_i est la matrice de covariance des vecteurs acoustiques contenus dans les segments de G_i. Le nombre de paramètres à estimer pour chaque groupe est $p+\frac{1}{2}p(p+1)$ où p représente la dimension des vecteurs acoustiques. Soit n_{v_i} le nombre de vecteurs acoustiques dans le groupe de segments G_i. Le critère d'Information Bayésien de la partition $\mathcal{G}_k$ est alors donné par la formule suivante :

$$\mathrm{BIC}(\mathcal{G}_k) = \sum_{i=1}^{k}\{-\frac{1}{2}n_{v_i}\log|\Sigma_i|\} - \lambda P \tag{7.16}$$

avec la pénalité :

$$P = \frac{1}{2}(p + \frac{1}{2}p(p+1))\log N \tag{7.17}$$

où $N = \sum_i n_{v_i}$ et le poids de pénalité λ est égal à 1 en théorie (cf [Rissanen 89, Hayes 96]). Parmi les partitions potentielles, la partition qui maximise le critère BIC défini à l'équation (7.16) est sélectionnée. Cependant, cette sélection peut s'avérer coûteuse en temps de calculs.

Dans le cas d'algorithmes de regroupement hiérarchique, l'application du critère BIC peut être sensiblement améliorée. Soit $\mathcal{G} = \{G_1, G_2, ..., G_k\}$ la partition courante. G_1 et G_2 sont les groupes de segments candidats au regroupement et le nouveau groupe est G. Le principe consiste à comparer la partition courante $\mathcal{G}$ avec la nouvelle partition $\mathcal{G}' = \{G, G_3, ..., G_k\}$. Chaque groupe de segments est modélisé par une distribution Gaussienne multi-dimensionnelle $N(\mu_i; \Sigma_i)$. D'après (7.16), la différence de BIC engendrée par le regroupement de G_1 et de G_2 est donnée par :

$$dBIC = -\frac{n_v}{2}\log|\Sigma| + \frac{n_{v_1}}{2}\log|\Sigma_1| + \frac{n_{v_2}}{2}\log|\Sigma_2| + \lambda P \tag{7.18}$$

où $n_v = n_{v_1} + n_{v_2}$ est la taille (en nombre de vecteurs acoustiques) du nouveau groupe de segments, Σ sa matrice de covariance, la pénalité est donnée par :

$$P = \frac{1}{2}(p + \frac{1}{2}p(p+1)) \log n_v$$

et le poids de pénalité λ est égal à 1 en théorie. La pénalité s'applique à la différence de BIC et non au BIC de chacune des partitions.

Le critère BIC, comme critère d'arrêt, consiste à ne pas réunir deux groupes de segments si (7.18) est négative. A chaque regroupement, la valeur de BIC de la partition résultante augmente. De cette manière, un arbre de classification "optimal" est construit au fur et à mesure du regroupement. Cette utilisation du critère BIC comme critère d'arrêt revient à utiliser un critère de regroupement contraint par la pénalité P.

Une autre possibilité consiste à utiliser comme critère de regroupement une distance comme celles précédemment mentionnées (rapport de vraisemblance généralisé ou distance de Kullbach-Leibler) et d'utiliser le critère BIC comme critère d'arrêt.

Cette méthode de regroupement hiérarchique est proposée par [Chen et al. 98c]. Elle est également exposée dans [Chen et al. 98b]. Ils utilisent en fait la dernière possibilité mentionnée : le rapport de vraisemblance généralisé (cf 2.3) comme critère de regroupement et le BIC comme critère d'arrêt. La réévaluation du critère de regroupement se fait par paire maximum. Le nombre réel de locuteurs est de 28 dans les 824 segments issus des données d'évaluation HUB-4 de 1996. Ces segments durent en moyenne 2 à 3 secondes. Le critère BIC comme critère d'arrêt fournit 31 groupes de segments, donc un nombre assez proche de la partition réelle. Par ailleurs, [Chen et al. 98c] définissent la pureté d'un groupe de segments G_i comme le rapport entre le nombre N_{maj} de segments du locuteur majoritaire et le nombre total N_i de segments contenus dans G_i :

$$\text{pureté } = \frac{N_{maj}}{N_i} \tag{7.19}$$

Cette définition est différente de celle proposée par [Solomonoff et al. 98] (cf equation 7.6).

Parmi les 31 groupes de segments obtenus, 21 d'entre eux ont une pureté égale à 100% (pureté idéale). Les 10 groupes restants ont une pureté qui varie de 67% à 99%. Donc globalement, une pureté assez élevée pour chacun des groupes de segments obtenus. Par ailleurs, [Chen et al. 98c] évaluent les performances de l'adaptation suite au regroupement hiérarchique. Les taux d'erreurs obtenus avec leur partition sont quasi-identiques à ceux obtenus avec la partition réelle.

Cet algorithme de regroupement présente l'avantage d'avoir un critère d'arrêt quasi-systématique, si nous ne tenons pas compte du paramètre λ intervenant dans l'équation (7.18). En effet, la valeur de λ est en théorie égale à 1. Mais en pratique, ce critère donne de meilleurs résultats pour des valeurs de λ différentes de 1, comme nous avons pu le voir dans la partie consacrée à la segmentation (cf section 5.3.3). Et rien ne permet encore de déterminer a priori la valeur de λ.

7.2 Regroupement séquentiel

Dans cette section, nous présentons deux méthodes de regroupement séquentiel. Nous en rappelons brièvement le principe (cf introduction de la partie II) : le premier segment (temporellement parlant) forme une première classe de locuteur. Les segments suivants sont examinés au fur et à mesure qu'ils sont détectés. Pour chaque segment, un critère de regroupement contraint est calculé par rapport à chaque classe de locuteur déjà existante. Si le critère est vérifié pour l'une des classes et qu'il est optimum par rapport à l'ensemble des classes alors le segment est ajouté au groupe de segment correspondant. A l'inverse, si le critère n'est vérifié pour aucune des classes alors un nouveau groupe de segments est créé avec ce segment.

7.2.1 Utilisation de sous-espaces propres du locuteur

Dans ce paragraphe, le critère de regroupement s'appuie sur des techniques de vérification du locuteur. Chaque groupe de segments créé est modélisé par un sous-espace propre (SEP). Soit $\{x_t^{(i)}\}$ la séquence de vecteurs acoustiques correspondant aux segments contenus dans la classe du locuteur i ($\{x_t^{(i)}\}$ correspond à la concaténation des séquences de vecteurs acoustiques formant chaque segment du groupe i). Le SEP du locuteur i est calculé de la manière suivante : soit la matrice $X^{(i)}$ dont les lignes sont formées par les vecteurs $x_t^{(i)} - \mu^{(i)}$ où $\mu^{(i)}$ est le vecteur moyen des vecteurs acoustiques $x_t^{(i)}$ $(1 \leq t \leq M)$. La matrice $X^{(i)}$ est décomposées en valeurs singulières (DVS : cf [Golub et al. 96, Pea 92]) :

$$X^{(i)} = U^{(i)} \Sigma^{(i)} V^{(i)T} \tag{7.20}$$

$U^{(i)}$ et $V^{(i)}$ sont les matrices dont les colonnes sont respectivement les vecteurs propres de $X^{(i)} X^{(i)T}$ et $X^{(i)T} X^{(i)}$. $\Sigma^{(i)}$ est la matrice des valeurs singulières de $X^{(i)}$. Les vecteurs propres de la matrice de corrélation de $X^{(i)T} X^{(i)}$ sont les vecteurs de base des données de parole $X^{(i)}$. $V^{(i)}$ est une base orthonormée de l'espace du locuteur. Si les r valeurs singulières les plus élevées sont sélectionnées dans $\Sigma^{(i)}$ alors $V^{(i)}$ devient une matrice $N \times r$ formée avec les vecteurs de base orthonormaux $\{v_1^{(i)} ... v_r^{(i)}\}$ et caractérise le SEP du locuteur i.

Pour définir le critère de regroupement associé au locuteur i, la distance d'un vecteur acoustique x_t au SEP i est définie comme suit :

$$dist(V^{(i)}, x_t) = \| x_t - \{\sum_{j=1}^{r} ((x_t - \mu^{(i)})^T v_j^{(i)}) v_j^{(i)} + \mu^{(i)}\} \|^2 \tag{7.21}$$

Cette distance est la norme du résidu orthogonal d'une projection de $x_t - \mu^{(i)}$ sur le SEP i. Le critère de regroupement du SEP i pour un segment $X^{(k)}$ composé de la séquence de vecteurs acoustiques $\{x_t^{(k)}\}$ $(1 \leq t \leq N)$ est alors défini comme la moyenne des distances des vecteurs acoustiques au SEP :

$$dist(V^{(i)}, X^{(k)}) = \frac{1}{N} \sum_t dist(V^{(i)}, x_t^{(k)}) \tag{7.22}$$

Ce critère de regroupement est contraint, c'est-à-dire qu'un seuil est imposé sur la distance de l'équation (7.22) au-delà duquel le segment n'est pas assigné au SEP considéré. Ce seuil θ est défini par :

$$\theta = \mu + \frac{\sigma}{3} \tag{7.23}$$

où μ et σ désignent la moyenne et la déviation standard de la distribution des distances entre les vecteurs acoustiques à partir desquels le SEP a été construit et le SEP.

Quand un nouveau segment est ajouté au SEP alors le SEP est mis à jour : une nouvelle DVS est effectuée sur tous les vecteurs acoustiques composant le groupe de segments correspondant et le seuil θ est également mis à jour.

Cette méthode est présentée par [Nishida et al. 98, Nishida et al. 99]. Elle permet de segmenter et indexer un journal télévisé par locuteurs en temps réel. Comme vu à la section 3.1.1, la segmentation repose sur une détection de silences. Les segments de parole correspondent alors à une section de parole entre deux silences. L'hypothèse que les locuteurs sont séparés par des silences significatifs (i.e. facilement détectables) est implicitement faite.

Les expériences sont menées sur 150 minutes de journaux télévisés NHK. Le but est d'extraire les paroles du présentateur. Pour évaluer le processus d'indexation, les auteurs définissent deux taux :

$$\text{taux d'extraction} = \frac{\#\text{ de segments du présentateur correctement identifiés}}{\#\text{ total de segments du présentateur}} \tag{7.24}$$

$$\text{taux de précision} = \frac{\#\text{ de segments du présentateur correctement identifiés}}{\#\text{ de segments identifiés comme présentateur}} \tag{7.25}$$

Le taux d'extraction est de 93.4% et le taux de précision est de 98.7%. Les auteurs signalent cependant qu'il est préférable d'avoir des segments assez longs, en particulier le premier, sinon l'indexation échoue. Ceci est un des points faibles de la méthode : les SEP ne sont pas représentatifs, étant donné le faible nombre de vecteurs utilisés. Par ailleurs, le seuil d'acceptabilité θ d'un nouveau segment dans un SEP, défini à l'équation 7.23, est sensible à la longueur des segments.

D'autre expériences sont menés sur des débats télévisés ou des téléfilms et, dans les deux cas, les résultats se dégradent de manière significative. De plus, les seuils sont choisis de manière complètement empiriques et semblent peu robustes.

7.2.2 Utilisation du BIC

Ces travaux font suite aux travaux sur le regroupement hiérarchique avec utilisation du critère BIC exposés à la section 7.1.8. Dans cette précédente section, le regroupement hiérarchique est réalisé *off-line*, i.e. tous les segments à partitionner sont à disposition. Dans la présente section, le regroupement se fait *on-line*, i.e. les segments sont traités au fur et à mesure de leur formation pour répondre à des contraintes de temps réel.

Soient $G_1, .., G_k$ les groupes de segments déjà formés et $S_1, .., S_M$ les segments restant à partitionner. A ce point, les gains de regroupement $\mathfrak{G}(G_i, S_j)$ et $\mathfrak{G}(S_i, S_j)$ pour tous les couples (i, j) possibles, sont calculés de la manière suivante :

$$\mathfrak{G}(X_i, X_j) = \text{BIC}(\text{regrouper } X_i \text{ et } X_j) - \text{BIC}(\text{garder } X_i \text{ et } X_j \text{ séparés}) \tag{7.26}$$

où X_i représente indifféremment un groupe de segments ou un segment. Ce gain est une différence de critère d'information Bayésien. Parmi tous les gains calculés, le maximum d'entre-eux est recherché $\mathfrak{G}(X_{i_0}, X_{j_0}) = \Delta\text{BIC}_{MAX}$. Si $\Delta\text{BIC}_{MAX} > 0$ alors le couple d'objets correspondant (G_{i_0}, S_{j_0}) ou (S_{i_0}, S_{j_0}) sont regroupés. Si le second couple d'objets est concerné, cela correspond à la création d'un nouveau groupe de segments. Par contre, si $\Delta\text{BIC}_{MAX} < 0$, alors les deux objets ne sont pas réunis. S'il s'agit du couple d'objets (G_{i_0}, S_{j_0}), il en résulte la création d'un nouveau groupe de segments. S'il s'agit du couple (S_{i_0}, S_{j_0}), il en résulte la création de deux nouveaux groupes de segments. Ce processus est répété jusqu'à ce que les M segments soient traités.

En théorie, cet algorithme *on-line* est sous-optimal comparé à l'algorithme *off-line*, présenté au paragraphe 7.1.8. En effet, les maxima dans l'algorithme *on-line* sont locaux alors qu'ils sont globaux dans l'autre algorithme. Cependant, les réunions optimales de segments concernent des segments proches temporellement parlant. En pratique, il se trouve que l'algorithme *on-line* prend mieux en compte ces relations que l'algorithme *off-line*. Par ailleurs, les segments trop petits ne sont pas considérés ici : ils sont collectés dans un groupe de segments "poubelle" (*garbage*).

[Tritschler et al. 99] comparent les résultats obtenus sur les données extraites de la base de données HUB-4 1997. Même si les deux algorithmes fournissent de bons résultats, l'algorithme *on-line* se montre plus performant que l'algorithme *off-line* en terme de rapidité, de pureté des groupes de segments (respectivement 98.58% contre 96.7%) et de nombre de groupes de segments.

Les mêmes remarques qu'au paragraphe 7.1.8 peuvent être faites concernant la valeur du paramètre λ qui intervient dans la formule 7.26.

7.3 Conclusions

Dans ce chapitre, plusieurs méthodes de regroupement hiérarchique ou séquentiel sont exposées. Parmi les critères de regroupement décrits, le rapport de vraisemblance généralisé ou la distance de Kullbach-Leibler nous semblent les plus prometteurs pour le regroupement en locuteurs. Leur capacité à distinguer des locuteurs différents a déjà été prouvée dans la partie I consacrée à la segmentation en locuteurs. En ce qui concernent les critères de regroupement, nous retiendrons surtout le critère d'information Bayésien , le critère de pureté défini par [Solomonoff et al. 98] ou encore l'entropie définie à l'équation 7.14. Les critères de sélection d'une partition à partir du dendrogramme ne sont pas encore arrivées suffisamment à maturité pour être utilisées telles quelles, d'autant que le problème se complique singulièrement quand le nombre de segments augmente.

Chapitre 8

Méthodes proposées

Dans ce chapitre, nous décrivons les techniques de regroupement hiérarchique que nous mettons en œuvre pour regrouper les segments par locuteurs. La section 8.1 explique les éventuels pré-traitements que nous réalisons avant d'effectuer le regroupement, afin d'en améliorer les performances. Puis la section 8.2 présente les techniques de regroupement utilisées proprement dites.

8.1 Pré-traitements

Le premier pré-traitement consiste en la soustraction de la moyenne cepstrale pour réduire les effets du canal de transmission (cf paragraphe 8.1.1). Le deuxième pré-traitement concerne les segments jugés trop courts (cf paragraphe 8.1.2).

8.1.1 Paramétrisation et soustraction de la moyenne cepstrale

Pour l'étape de segmentation, nous travaillons sur le signal paramétrisé par des coefficients Mel-cepstraux. Nous utilisons la même paramétrisation pour l'étape de regroupement.

[Furui 81] montre que l'utilisation des coefficients cepstraux, moyennant une transformation, permet d'annuler les distortions introduites par les canaux de transmission. Elle permet également de réduire l'intra-variabilité à long terme d'un locuteur. Cette transformation est simple : elle consiste à soustraire à chaque vecteur cepstral de la séquence relative au locuteur considéré, le vecteur cepstral moyen estimé sur l'ensemble de la séquence.

Soit $X = \{x_1, \ldots, x_N\}$ une séquence de vecteurs acoustiques générés par un locuteur L_X. Chaque vecteur est composé de coefficients cepstraux $x[i]$ pour i variant de 1 à d, d étant la dimension de l'espace cepstral. Alors la moyenne cepstrale est calculée pour chaque coefficient :

$$\mu_c[i] = \frac{1}{N}\sum_{t=1}^{N} x_t[i] \qquad \forall i \in \{1 \ldots d\}$$

Après transformation, les coefficients acoustiques sont de la forme :

$$x_t[i] - \mu_c[i] \qquad \forall i \in \{1 \ldots d\}$$

Nous n'utilisons pas cette technique pour l'étape de segmentation pour deux raisons. La première raison est que nous ne connaissons pas les segments de locuteurs. La deuxième raison que nous invoquons est que cette technique n'est efficace que si la moyenne cepstrale est calculée sur les données du locuteur considéré. En effet, retirer la moyenne cepstrale calculée sur l'ensemble des données audio n'apporte rien (mais ne dégrade pas non plus les performances de la segmentation puisque cela revient à soustraire la même constante à l'ensemble des données). La deuxième raison est liée à la nature de notre méthode de segmentation. En effet, DISTBIC s'apparente à un algorithme de détection de ruptures dans le signal. Aussi, il est plus facile de détecter un changement entre deux locuteurs qui utilisent des canaux différents qu'entre deux locuteurs qui utilisent des canaux ayant les mêmes caractéristiques cepstrales.

L'information de canal dans le contexte de la segmentation en locuteurs est un atout supplémentaire. Dans le cadre du regroupement hiérarchique, elle peut constituer tantôt un atout, tantôt un inconvénient. Si nous considérons un document audio dans lequel les locuteurs utilisent le même canal de transmission durant toute la conversation et que ce canal n'est pas sujet à de fortes variations (par exemple, les conditions d'enregitrement du journaliste-présentateur d'un journal télévisé) alors cette information de canal peut permettre de distinguer les locuteurs entre-eux et donc de mieux les séparer. Par contre, si au sein d'un même document audio, les locuteurs changent de canal de transmission (par exemple, les messages déposés sur une boîte vocale par un même locuteur appelant de différents endroits avec différents terminaux) ou si leur canal de transmission subit des variations importantes (par exemple, avec l'emploi d'un téléphone mobile) alors il est intéressant de réduire les effets du canal. Ces variations viendront en effet gêner le regroupement des segments d'un locuteur donné.

C'est pourquoi nous soustrayons à chaque segment de locuteurs issu de la segmentation la moyenne cepstrale calculée sur la séquence de vecteurs acoustiques qu'il contient.

8.1.2 Traitement des segments courts

Avant de commencer le regroupement des segments, nous effectuons un deuxième prétraitement sur les segments courts.

Lors de la segmentation, nous avons vu que les segments courts posaient des problèmes de modélisation. Ils contiennent en effet trop peu de données pour évaluer de manière fiable et robuste les paramètres d'un modèle de locuteur. Nous entendons par segment court un segment dont la durée n'excède pas deux secondes.

Pour éviter qu'ils soient mal modélisés et en conséquence, qu'ils perturbent le processus de regroupement, nous éliminons les segments courts dans un premier temps. Une fois le regroupement des autres segments effectué, des groupes de locuteurs sont formés. Nous pouvons alors réintégrer ces segments courts.

Pour les réintégrer, nous procédons de la manière suivante. Pour chaque segment court, nous calculons le critère de regroupement utilisé lors du processus de regroupement, entre ce segment et les groupes de segments issus du regroupement. Si le segment court satisfait le critère pour un des groupes de segments et que ce critère est optimal au regard des autres groupes de segments, alors le segment court est ajouté au groupe de segments correspondant. A l'inverse, si le segment ne satisfait pas le critère quel que soit le groupe de segments considéré alors le segment court n'est pas pris en compte.

Nous ne formons pas un nouveau groupe de segments, i.e. un nouveau locuteur, avec ce segment court. Dans le cas contraire, nous aboutirions à une mauvaise modélisation du locuteur, toujours en raison du faible volume de données, et ce locuteur pourrait perturber l'étape de reconnaissance, étape finale du système d'indexation.

Par ailleurs, en rejetant un segment court, nous pouvons également ne pas détecter une personne qui n'intervient qu'une seule fois et rapidement dans la conversation. Nous reviendrons sur ce point à la section III.

8.2 Regroupement hiérarchique

Le chapitre précédent nous amène à penser que le rapport de vraisemblance généralisé (cf équation 2.3) constitue un bon critère de regroupement pour les segments appartenant à un même locuteur. Cette idée est renforcée par le fait que cette distance a déjà prouvé son efficacité lors de l'étape de segmentation. Nous la choisissons donc comme critère de regroupement. Cette distance n'étant pas contrainte, nous choisissons le critère BIC comme critère d'arrêt (cf équation7.18). L'association de ces deux critères pour le regroupement hiérarchique revient au regroupement hiérarchique finalement utilisé par [Chen et al. 98c]. Cet algorithme ayant été décrit en détails auparavant, nous ne nous étendons guère plus sur celui-ci et nous renvoyons le lecteur au paragraphe 7.1.8.

Par contre, n'étant pas soumis à des contraintes de temps réel, nous réestimons à chaque regroupement les distances inter-groupes susceptibles d'être modifiées (*estimation complète* , cf II).

Chapitre 9

Expériences et Résultats

Ce chapitre est consacré aux expériences menées sur les techniques de regroupement de segments par locuteurs. La section 9.1 décrit tout d'abord les méthodes employées pour tester les performances de ces algorithmes. La section suivante présente les résultats obtenus avec l'algorithme de regroupement hiérarchique d'une part appliqué à des segments purs (section 9.2.1) et d'autre part, à des segments résultant de la segmentation précédente (section 9.2.2). La section consacrée au regroupement hiérarchique appliqué à des segments purs étudie également l'influence des paramètres intervenant dans cet algorithme. Enfin, nous concluons sur cette technique de regroupement hiérarchique.

9.1 Méthodes d'évaluation

Pour qu'un regroupement des segments par locuteur soit correct dans le contexte de l'indexation, il doit satisfaire aux conditions suivantes :

- il doit y avoir autant de groupes de segments N_G que de locuteurs N_L présents dans la conversation
- chaque groupe de segments doit ne contenir que les segments relatifs à un même locuteur et tous les segments de ce locuteur doivent se trouver dans ce groupe de segments

La première condition est facile à évaluer. Il suffit de compter le nombre de groupes de segments obtenus et de comparer ce nombre au nombre de locuteurs effectivement engagés dans la conversation.

La deuxième condition peut être évaluée en considérant d'une part le nombre de groupes de segments obtenus et d'autre part la pureté de chaque groupe de segments. Soit $n_{i,j}$ le nombre de segments du groupe i prononcés par le locuteur j et n_i le nombre total de segments contenu dans le groupe i. La pureté $p_{\infty,i}$ du groupe de segments i peut se définir comme suit :

$$\begin{aligned} p_{\infty,i} &= 100 \times \frac{\text{nombre de segments du locuteur majoritaire } k}{\text{nombre total de segments contenus dans le groupe } i} \% \\ &= 100 \times \frac{n_{i,k}}{n_i} \% \end{aligned} \tag{9.1}$$

Cette définition est celle donnée par [Chen et al. 98b] (cf section 7.1.8). Nous préférons cette définition à celle proposée par [Solomonoff et al. 98] à l'équation 7.6 (cf section 7.1.3). En utilisant les mêmes notations que précédemment, nous en rappelons ici l'expression :

$$p_{2,i} = 100 \times \sum_j \frac{n_{ij}^2}{n_i^2}\%$$

En effet, la pureté $p_{\infty,i}$ nous renseigne sur la proportion qu'occupe le locuteur majoritaire au sein du groupe de segments. La pureté $p_{2,i}$ représente la probabilité que deux segments choisis aléatoirement dans le groupe i proviennent du même locuteur.

Il est intéressant d'observer que nous pouvons définir une famille de puretés de façon plus générale de type L^l de la forme :

$$p_{l,i} = 100 \times \sum_j \left(\frac{n_{ij}}{n_i}\right)^l \% \tag{9.2}$$

$p_{l,i}$ représente la probabilité que l segments pris aléatoirement dans le groupe i proviennent du même locuteur. A partir de l'équation 9.2 et de la définition de la norme L^∞, nous avons :

$$\lim_{l \mapsto \infty} (p_{l,i})^{\frac{1}{l}} = p_{\infty,i}$$

ce qui justifie la notation utilisée à l'équation 9.1.

Nous établissons aussi la relation suivante entre les puretés $p_{2,i}$ et $p_{\infty,i}$:

$$\begin{aligned} p_{2,i} &= \sum_j \frac{n_{ij}^2}{n_i^2} \\ &= \frac{n_{ik}^2}{n_i^2} + \sum_{j \neq k} \frac{n_{ij}^2}{n_i^2} \\ &= p_{\infty,i}^2 + \sum_{j \neq k} \frac{n_{ij}^2}{n_i^2} \end{aligned}$$

$$p_{2,i} = p_{\infty,i}^2 + \sum_{j \neq k} \frac{n_{ij}^2}{n_i^2} \tag{9.3}$$

Nous pouvons facilement démontrer que ces puretés sont égales, et de valeur 1, quand il n'y a qu'un seul locuteur dans le groupe. Quand les locuteurs sont équiprobables au sein d'un même groupe, ces deux puretés sont aussi égales et leur valeur est égale à la proportion qu'occupe chaque segment dans le groupe.

Nous démontrons à l'annexe D de ce document que $p_{2,i}$ nous renseigne sur la répartition des segments appartenant aux locuteurs non majoritaires dans le groupe i. En effet, $p_{2,i}$ prend une valeur minimale si tous les autres locuteurs potentiels apparaissent avec la même fréquence dans le groupe i.

Quoiqu'il en soit, il nous semble peu opportun de calculer ces puretés en termes de nombres de segments. En effet, si la pureté $p_{\infty,i}$ est égale à 95%, nous pourrions en conclure que le groupe de segments est plutôt homogène. Mais si les segments du locuteur majoritaire sont de l'ordre de quelques secondes et l'un ou plusieurs des segments contaminants durent plusieurs minutes, alors il y a de fortes chances pour que le groupe de segments ne soit pas vraiment homogène et surtout amène à la construction d'un modèle de locuteur de mauvaise qualité.

Nous proposons donc de remplacer dans les deux définitions précédentes, le nombre de segments par le nombre de secondes ou le nombre de trames d'analyse :

$$p_{\infty,i} = 100 \times \frac{\text{nombre de trames du locuteur majoritaire } k}{\text{nombre total de trames contenues dans le groupe } i}\% \tag{9.4}$$

$$p_{2,i} = 100 \times \sum_j \frac{(\text{nombre de trames du groupe } i \text{ prononcées par le locuteur } j)^2}{(\text{nombre total de trames contenues dans le groupe } i)^2}\% \tag{9.5}$$

Nous avons ainsi une idée plus juste de la pureté des groupes de segments.

Enfin, ces deux mesures nécessitent la connaissance de l'indexation de référence. Une solution consiste à les estimer par la méthode proposée par [Solomonoff et al. 98] (cf section 7.1.3), estimation qui se révèle fausse si la partition obtenue ne possède pas une configuration proche de la partition idéale. Dans ce chapitre, nous calculons la pureté a posteriori, en nous servant de la partition de référence.

Concernant le nombre de groupes de segments, nous ajoutons qu'il est préférable d'avoir plus de groupes de segments que de locuteurs, si toutefois ces groupes sont purs. Si le nombre de groupes de segments est inférieur au nombre de locuteurs, cela implique que les paroles provenant de locuteurs différents aient été réunies au sein d'un même groupe de segments. En d'autres termes, les groupes de segments obtenus sont impurs, entraînant à l'étape suivante des erreurs de modélisation des locuteurs.

La situation extrême inverse, i.e. une partition ne contenant que des singletons, risque d'aboutir également à des erreurs de modélisation des locuteurs. En effet, la faible quantité de données présente dans chaque singleton ne permet pas de construire des modèles de locuteurs suffisamment fiables.

9.2 Expériences

Nous menons les expériences en deux étapes. La première étape consiste à étudier le comportement de l'algorithme de groupement décrit au chapitre précédent sur différents types de données. Ces données sont générées à partir de la segmentation de référence quand celle-ci est disponible. Ceci signifie que les segments en entrée du système sont parfaitement purs : ils ne contiennent les paroles que d'un seul locuteur. La deuxième étape consiste à tester cet algorithme de regroupement sur les mêmes données que précédemment, mais qui résultent cette fois-ci de notre méthode de segmentation. Il se peut donc que certains segments soient impurs.

9.2.1 Evaluation avec des segments de référence

Le but de cette évaluation, réalisée avec des segments de référence, est de déterminer les paramètres optimaux et d'analyser l'évolution des résultats en fonction de la valeur des paramètres.

Description des données

Nous testons l'algorithme de regroupement hiérarchique sur différents types de données :

- 10 conversations créées artificiellement en concaténant des phrases extraites de la base de données TIMIT (parole propre, segments courts de 2 à 4 secondes, anglais, 52 minutes)

- 10 conversations créées artificiellement en concaténant des phrases extraites de la base de données fournie par le Centre National d'Etudes des Télécommunications (CNET) (parole propre, segments courts de 1 à 3 secondes, français, 34 minutes)

- 2 dialogues *dial1* et *dial2* (impliquant deux personnes) créés artificiellement en concaténant des phrases extraites d'une autre base de données fournie par le Centre National d'Etudes des Télécommunications (parole propre, segments longs, supérieurs à 2 secondes, français, 13 minutes). Deux autres dialogues *dial1_sans_sil* et *dial2_sans_sil* sont créés en utilisant les dialogues précédents et en supprimant tous les longs silences. Ces dialogues sont référencés par la suite par DIAL. L'un des dialogues est entre deux femmes et l'autre entre deux hommes.

- 2 conversations *conv1* et *conv2* (impliquant dix personnes) créées artificiellement en concaténant des phrases extraites d'une autre base de données fournie par le Centre National d'Etudes des Télécommunications (parole propre, segments longs, supérieurs à 2 secondes, français, 32 minutes). Deux autres conversations *conv1_sans_sil* et *conv2_sans_sil* sont créées en utilisant les conversations précédentes et en supprimant tous les longs silences. Ces conversations sont par la suite appelées CONV. Elles contiennent chacune autant d'hommes que de femmes.

Pour chaque expérience, nous présentons 3 graphes de résultats. Le graphe de gauche est consacré au nombre de locuteurs. Il mentionne le **nombre réel de locuteurs** présents dans le document audio et le **nombre de locuteurs effectivement trouvés** suite au regroupement hiérarchique. Ces nombres sont des moyennes sur l'ensemble des documents audio évalués. Le graphe du milieu présente **les puretés** $p_{\infty,i}$ et $p_{2,i}$ définies aux équations (9.4) et (9.5) exprimées en pourcentage. Enfin, le graphe de droite montre la **durée moyenne en secondes** des groupes de locuteurs résultant.

Comme annoncé à la section 9.1, un regroupement hiérarchique est jugé correct si le nombre de locuteurs trouvés est proche du nombre réel, si les valeurs des puretés sont élevées et si les groupes de segments ont une durée moyenne permettant une modélisation plus fiable du locuteur correspondant.

Nous mentionnons les deux puretés définies précédemment mais nous nous basons essentiellement sur la pureté $p_{\infty,i}$ qui à nos yeux est plus facilement interprétable dans le cadre de notre application. Ces deux puretés évoluent dans le même sens mais pas forcément dans les même proportions.

Influence de la paramétrisation

Pour étudier l'influence de la paramétrisation, ou plus exactement l'influence de la taille de l'espace acoustique, nous testons le regroupement hiérarchique avec plusieurs dimensions de vecteurs acoustiques, les autres paramètres étant fixés, et nous comparons les partitions obtenues. Les coefficients acoustiques sont des coefficients Mel-cepstraux comme pour l'étape de segmentation. Les graphes 9.1, 9.2, 9.3 et 9.4 (respectivement les tableaux E.1, E.2, E.3 et E.4) donnent les résultats pour les données de type CNET, TIMIT, DIAL et CONV respectivement pour des dimensions de vecteurs acoustiques de 12, 16 et 24 (nous ne mentionnons pas les résultats sur la dimension 24 avec les données CONV).

Pour les quatre types de données, nous constatons que plus la dimension augmente, plus il y a de regroupements de segments. Par exemple, pour les données CNET, le nombre de locuteurs effectivement trouvés passe de 14.7% en dimension 12 à 8.6% en dimension 24, pour les données TIMIT, de 31.5% en dimension 12 à 15.1% en dimension 24 et enfin pour les données DIAL de 7.8% en dimension 12 à 2.8% en dimension 24 et pour les données CONV de 29.3% en dimension 12 à 21.5% en dimension 16.

La dimension de l'espace acoustique intervient à deux endroits : d'une part, dans le calcul du critère de regroupement (rapport de vraisemblance généralisé) et d'autre part, dans le calcul du critère d'arrêt (critère BIC). La distance permet de choisir les deux groupes de segments à réunir. Une étude plus détaillée des résultats montre que ce sont pratiquement les mêmes couples de groupes de segments qui sont regroupés et dans le même ordre. Ceci signifie que le calcul de la distance est sensible à la dimension de l'espace acoustique mais sans incidence notable sur le regroupement. Par contre, le regroupement des deux groupes de segments se fait tant que le critère d'arrêt n'est pas vérifié. Ceci montre que le critère BIC est particulièrement sensible à la dimension de l'espace acoustique.

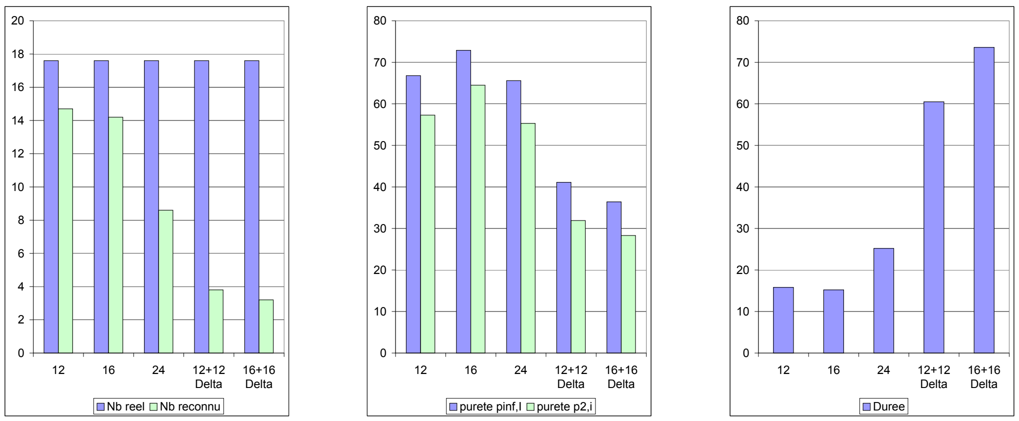

FIG. 9.1 – *Données CNET : influence de la dimension de l'espace acoustique et de l'apport des coefficients* Δ *(*$\lambda = 1.0$*)*

Par ailleurs, nous évaluons également l'influence des informations dynamiques en utilisant des vecteurs acoustiques composés de coefficients et de leurs dérivées premières (coefficients Δ). L'utilisation de ces coefficients Δ aboutit à une forte dégradation des résultats en termes de pureté. La pureté $p_{\infty,i}$ pour les données CNET est de 66.8% pour la dimension 12 et de 65.6% pour la dimension 24 et seulement de 41.1% pour la dimension 24 formées de 12 coefficients et de leur dérivée première. Cette tendance est encore plus forte pour la dimension 16 et la dimension 32 composée de 16 coefficients et 16 Δ-coefficients : la pureté passe de 72.9% à 36.4%. Dans ce dernier cas, la dégradation est à la fois due à l'emploi des dérivées premières mais aussi à l'augmentation de la dimension des vecteurs acoustiques. Les mêmes baisses de performances sont constatées pour les données TIMIT (cf E.2) et pour les données DIAL (cf E.3). Par la suite, nous n'utilisons plus ces Δ-coefficients.

Il nous reste à dire que des dimensions de 12 ou 16, au vu des résultats, sont les dimensions à recommander (pour les valeurs des autres paramètres que nous utilisons) quel que soit le type de données. Ces dimensions présentent aussi l'avantage de ne pas être trop élevées et,

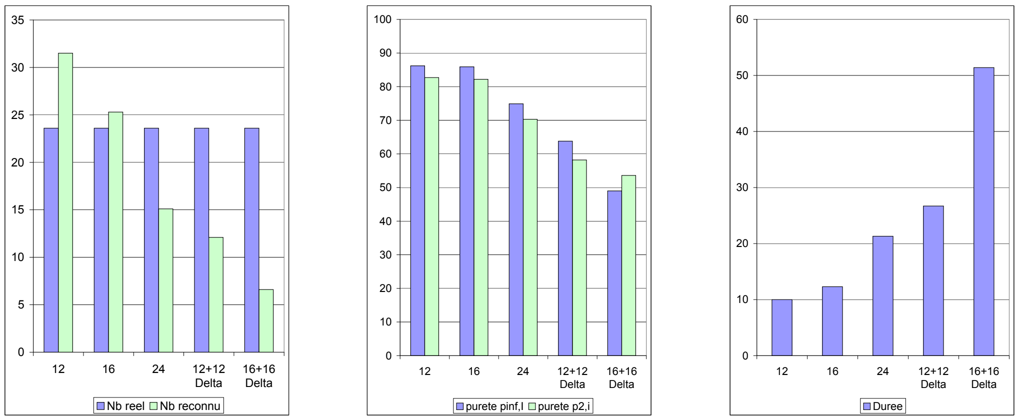

FIG. 9.2 – *Données TIMIT : influence de la dimension de l'espace acoustique et de l'apport des coefficients* Δ *(*$\lambda = 1.0$*)*

par conséquent, de conserver une complexité de calcul raisonnable.

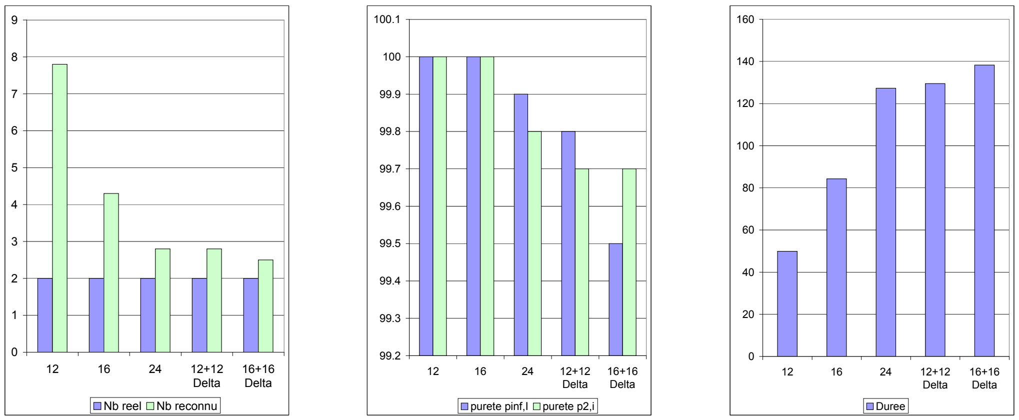

FIG. 9.3 – *Données DIAL : influence de la dimension de l'espace acoustique et de l'apport des coefficients* Δ $(\lambda = 1.2)$

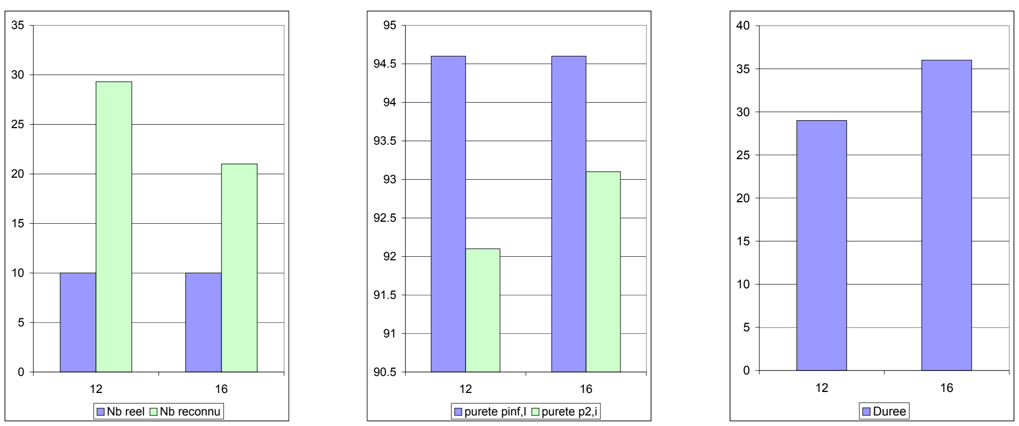

FIG. 9.4 – *Données CONV : influence de la dimension de l'espace acoustique* $(\lambda = 1.2)$

Influence de la pénalité λ intervenant dans le critère d'arrêt

Le facteur λ intervient dans le Critère d'Information Bayésien (cf équation 7.18) qui nous sert de critère d'arrêt. Pour rappel, λ représente la pondération du terme de pénalité. Lors de l'évaluation de la méthode de segmentation (chapitre 5), nous avons vu que plus la valeur de λ était élevée, moins le signal était segmenté et donc plus il y avait de regroupements de segments. Nous retrouvons le même résultat pour le regroupement hiérarchique.

Nous testons différentes valeurs de λ pour chaque type de données. Pour les données CNET et TIMIT (cf graphes 9.5 et 9.6 et tableaux E.5 et E.6), nous prenons les valeurs de 0.8, 1.0 et 1.2. Ces valeurs nous sont suggérées par les résultats obtenus pour la segmentation. En effet, CNET et TIMIT sont des documents audio contenant de courts segments (de l'ordre de 1 à 4 secondes) et d'après les résultats sur la segmentation, la valeur du paramètre λ dépend essentiellement de la longueur réelle des segments de locuteurs. La valeur recommandée pour les segments courts dans le cadre de la segmentation est de 1.2.

Pour une valeur de λ de 0.8, le nombre de locuteurs reconnu est de 21.7 pour les données CNET et de 36.5 pour les données TIMIT. Ce nombre n'est plus que de 9.1 pour les données CNET et de 17.6 pour les données TIMIT pour une valeur de λ de 1.2. Cela confirme donc que plus la valeur de λ est élevée et plus il y a de regroupements effectués.

Pour les données CNET, il est difficile de déterminer pour quelle valeur de λ la partition obtenue est la meilleure. Une valeur de 0.8 aboutit à une partition contenant plus de locuteurs qu'en réalité avec une pureté de 79.0% mais une durée moyenne de seulement 10.1 s. Une valeur de λ de 1.0 fournit une partition contenant moins de locuteurs qu'en réalité avec une pureté moindre (72.9%) mais par contre, une durée moyenne plus longue : 15.2s. Le tout est de savoir ce qu'il vaut mieux privilégier pour la modélisation qui constitue l'étape suivante : des groupes de segments plus longs et moins purs ou des segments plus courts et plus purs? Seule la mise en œuvre de l'étape suivante (la modélisation des locuteurs et la reconnaissance de la séquence de locuteurs à l'aide de ces modèles) nous permettra de répondre.

Pour les données TIMIT, la question se pose moins. En effet, une valeur de λ de 1.0 fournit un nombre de locuteurs proche du nombre réel (25.3 pour 23.6) avec une pureté de 85.9% et une durée de 12.3.

Le graphe 9.7 (tableau E.7) présente les résultats pour les données DIAL qui sont des documents audio dont la longueur réelle des segments est plus grande que celles de CNET ou de TIMIT. Aussi, nous menons des expériences avec des valeurs plus élevées de λ, variant de 1.0 à 1.5. Les meilleurs résultats sont obtenus avec λ égal à 1.5 : la pureté est de 100% avec un nombre de locuteurs de 3.8 (il est en réalité de 2) et une durée moyenne de 106.6 s. Les valeurs de λ de 1.0 et 1.2 ne permettent pas de regrouper suffisamment les segments d'un même locuteur. Ce résultat se retrouve au graphe 9.8 (tableau E.8) avec les données CONV, conversations composées également de segments longs. La valeur de λ fixée à 1.5 fournit le meilleur compromis entre le nombre de locuteurs obtenus comparé au nombre réel, entre la valeur de pureté de 92.4% et entre la durée moyenne des groupes de segments de 56.6 s.

En résumé, les résultats expérimentaux mettent en évidence deux points :

- plus la valeur du paramètre λ intervenant dans le critère d'arrêt BIC est élevée, plus il y a de regroupements de groupes de segments
- la valeur de λ est dictée par la longueur réelle des segments.

Ces résultats confirment ceux que nous avions obtenus pour la segmentation.

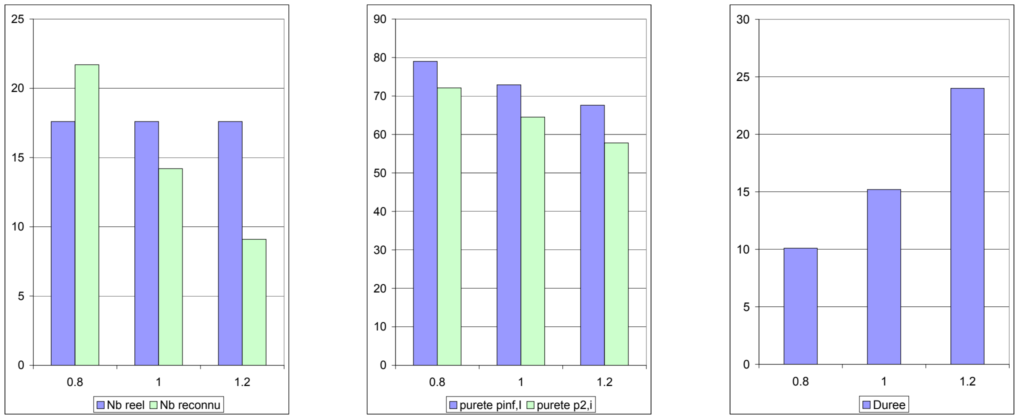

Fig. 9.5 – *Données CNET: influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

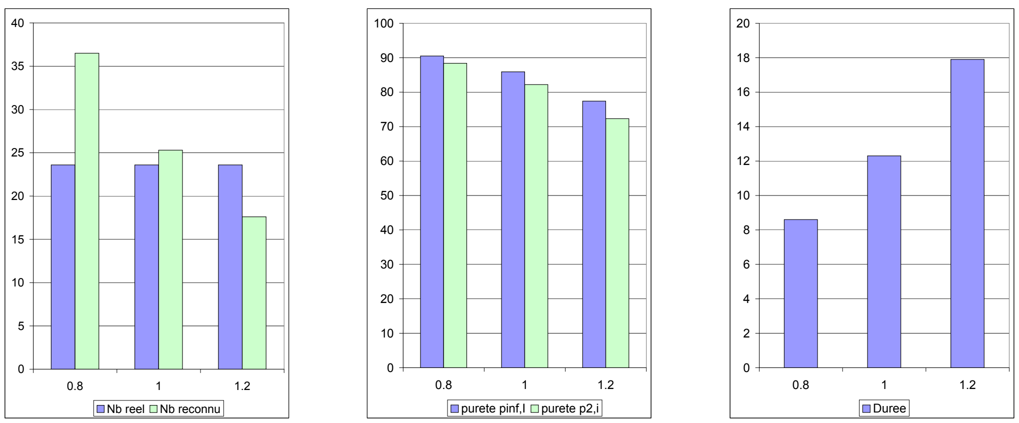

Fig. 9.6 – *Données TIMIT: influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

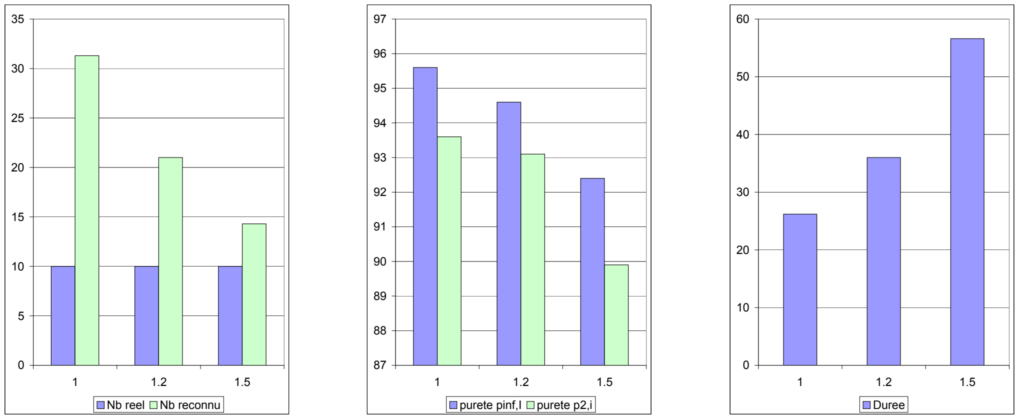

FIG. 9.7 – *Données DIAL : influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

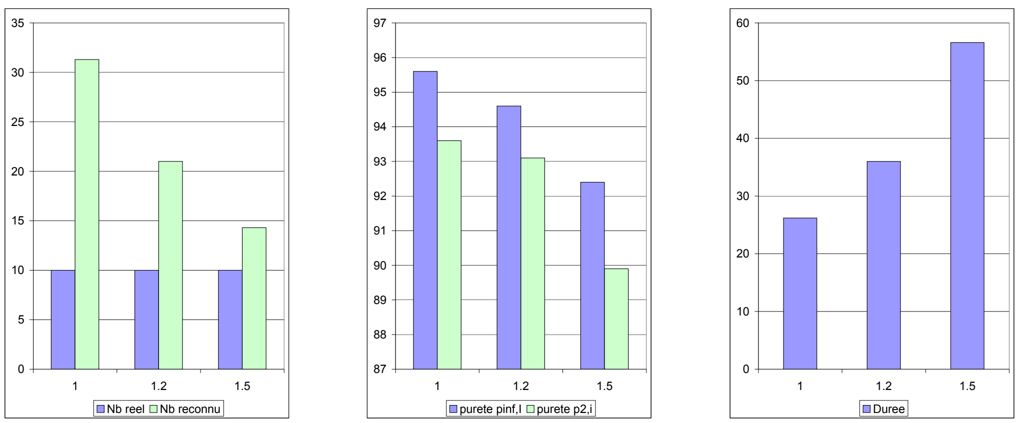

FIG. 9.8 – *Données CONV : influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

Influence du pré-traitement et du post-traitement pour les segments courts

Dans cette section, nous testons l'influence des segments courts, i.e. dont la durée est inférieure à 2 secondes dans nos expériences. En effet, nous mettons en place des traitements particuliers pour ces segments comme expliqué au paragraphe 8.1.2. Nous rappelons brièvement en quoi consiste ces traitements. Le pré-traitement vise à écarter les segments courts du regroupement hiérarchique. le post-traitement permet ou non, une fois le regroupement hiérarchique des autres segments terminé, d'attribuer les segments courts aux groupes de segments obtenus. Nous testons trois versions de l'algorithme :

- *avec pré/avec post* : les segments courts sont écartés du regroupement hiérarchique et sont éventuellement agglomérés aux groupes de segments résultants.
- *avec pré/sans post* : les segments courts sont éliminés définitivement du regroupement hiérarchique.
- *sans pré/sans post* : les segments courts sont considérés comme tout autre segment.

Notons que, par construction, le nombre de locuteurs obtenus entre les versions *avec pré/avec post* et *avec pré/sans post* ne varie pas. En effet, le post-traitement ne crée pas de nouveau groupe de segments donc pas de nouveau locuteur.

Les graphes 9.9 et 9.10 (tableaux E.9 et E.10) présentent les résultats obtenus avec les conversations contenant des segments courts CNET et TIMIT respectivement. Pour les données CNET, les résultats se dégradent sensiblement entre les regroupements *avec pré/sans post* et *avec pré/avec post*. La pureté perd 20% : sa valeur passe de 93.2% à 79.0%. En parallèle, la durée moyenne des groupes de segments passe de 6.8s à 10.1s. Le post-traitement des segments courts ne semble donc pas être d'une grande efficacité. Son emploi est même à proscrire pour les données CNET. Cette baisse de performances est beaucoup moins sensible pour les données TIMIT : la pureté passe de 90.9% à 90.5% avec en parallèle une augmentation de la durée moyenne de 8.4s à 8.6s. L'effet du post-traitement, quel qu'il soit, n'est donc pas flagrant dans le cas présent. Cette différence de performances entre les données CNET et TIMIT peuvent s'expliquer pour deux raisons. La première raison est que la durée des groupes de segments obtenus est beaucoup plus faible dans le cas des données CNET (6.8s) que dans le cas des données TIMIT (8.4s). Ceci aboutit donc à une meilleure modélisation des groupes de segments dans le cas des données TIMIT. La deuxième raison, qui sans doute contribue majoritairement à cette différence de performances, est que les segments CNET sont en moyenne beaucoup plus courts que les segments de TIMIT. Certains font à peine 0.3s. Aussi, ils sont très mal modélisés et facilement attribués à des groupes de segments qui ne leur correspondent pas.

La suppression du pré-traitement aboutit à une augmentation du nombre de locuteurs trouvés : pour les données CNET, ce nombre passe de 21.7 à 47.2 et pour les données TIMIT, ce nombre augmente dans une moindre mesure : il passe de 36.5 à 38.6. Pour les données CNET, le pré-traitement améliore les résultats. Ceci est moins flagrant pour les données TIMIT : cette faible augmentation du nombre de locuteurs trouvés s'accompagne d'une légère amélioration de la pureté : sa valeur passe de 90.5% à 91.6%. Il est donc difficile de conclure sur l'efficacité du pré-traitement pour les données TIMIT.

Les graphes 9.11 et 9.12 (tableaux E.11 et E.12) donnent les résultats du regroupement hiérarchique appliqué à des conversations contenant une majorité de longs segments : DIAL et CONV respectivement. Pour les dialogues, seule la durée moyenne des groupes de segments

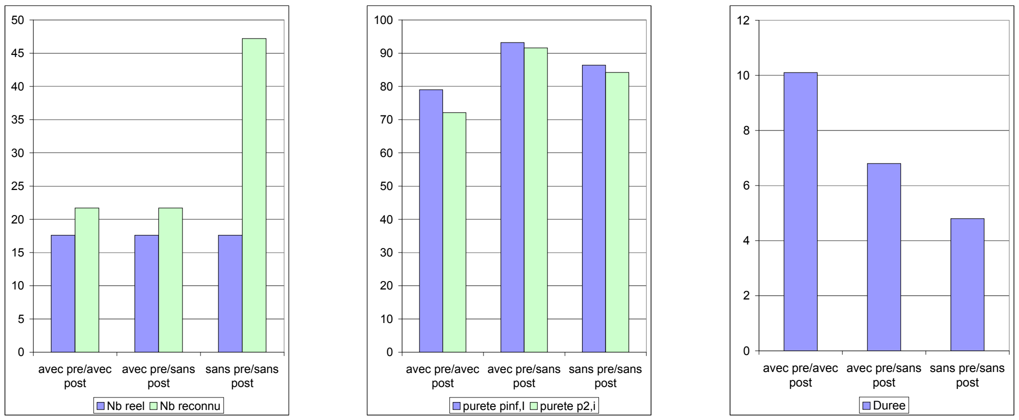

FIG. 9.9 – *Données CNET : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt $\lambda = 0.8$)*

est modifiée : elle passe de 100.4s à 106.6s entre les regroupements *avec pré/sans post* et *avec pré/avec post* pour une même valeur de pureté. Il n'est pas sûr que le gain de 2 s sur une durée moyenne d'une centaine de secondes justifie l'emploi du post-traitement. Les regroupements *avec pré/sans post* et *avec pré/avec post* aboutissent à des puretés respectives de 93.4% et de 89.9% sur les données CONV, avec en parallèle des durées moyennes respectives de 34.3s et de 56.6s. Il est à nouveau difficile de conclure ici : est-ce qu'une perte de 3.5% de pureté accompagnée d'un gain de 20s pour la durée moyenne est préférable pour la modélisation des locuteurs et la reconnaissance de la séquence de locuteurs, à une pureté plus élevée maisassociée à une durée moyenne moindre pour les groupes de segments ?

La suppression du pré-traitement amène, comme pour les données CNET et TIMIT à une augmentation du nombre de locuteurs : il passe de 3.8 à 4.5 pour les données DIAL et de 14.3 à 28.0 pour les données CONV. Dans le cas des données DIAL, cette augmentation du nombre de locuteurs trouvés est accompagnée d'une baisse de la pureté (de 100% à 99.8%) et d'une baisse de la durée moyenne (de 106.6s à 81.9s). L'emploi du pré-traitement est donc conseillé pour les données DIAL. Pour les données CONV, nous retombons sur le même problème que pour le post-traitement car en parallèle de l'augmentation du nombre de locuteurs trouvés, il y a une hausse de la pureté (de 92.4% à 94.3%) et une baisse de la durée moyenne de plus de 20s (de 56.6s à 34.2s).

En résumé, le post-traitement ne prouve pas son efficacité pour les conversations contenant de courts segments. Il peut même être à l'origine de fortes dégradations. Pour les conversations contenant de longs segments, il est difficile de conclure. Seule la mise en place de l'étape suivante du système d'indexation permettra de conclure.

Quant au pré-traitement, il améliore les résultats pour les données CNET et DIAL. Pour les données TIMIT et CONV, son efficacité reste également à prouver. De même que pour le post-traitement, seule l'étape suivante nous permettra de conclure.

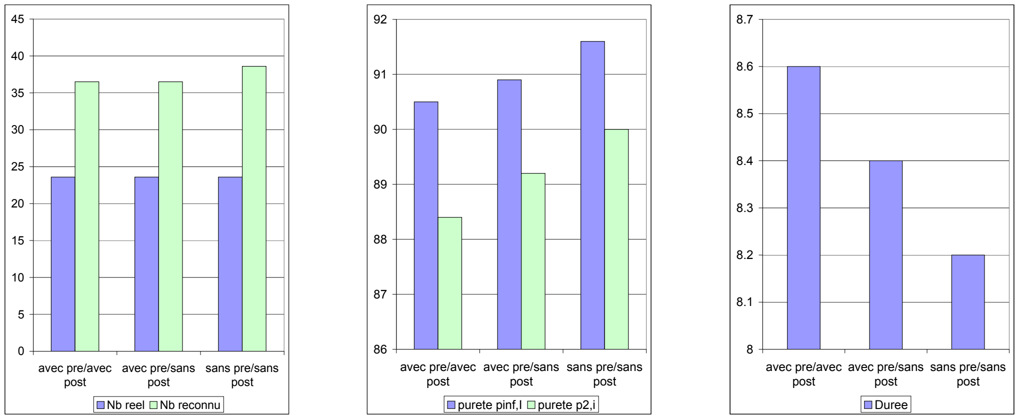

FIG. 9.10 – *Données TIMIT : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt* $\lambda = 0.8$*)*

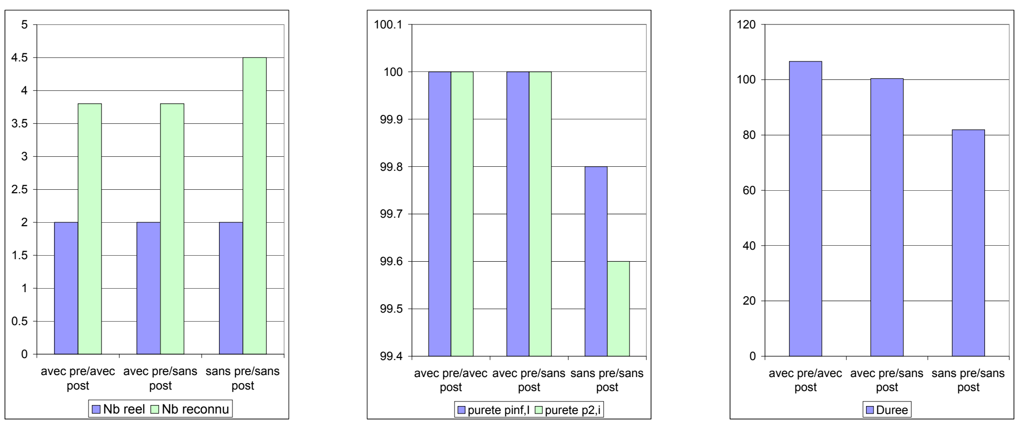

FIG. 9.11 – *Données DIAL : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

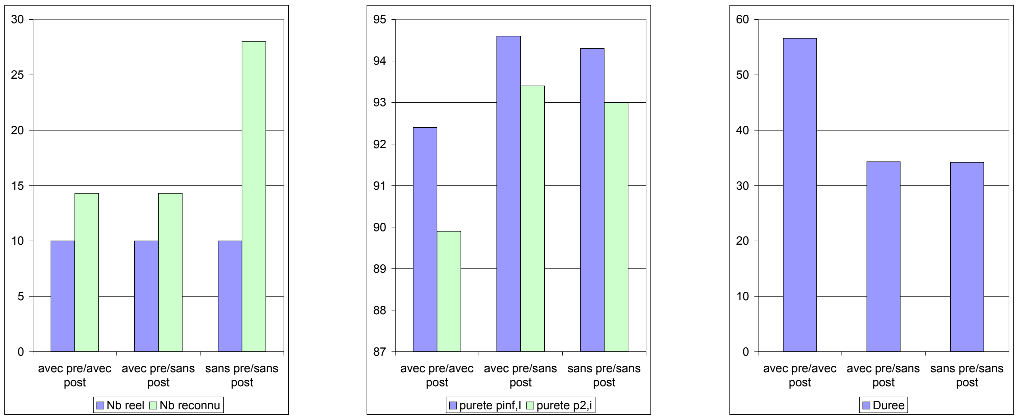

FIG. 9.12 – *Données CONV: influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

Réévaluation des distances après regroupement

Au chapitre d'introduction de cette partie consacrée au regroupement des segments par locuteurs, nous avons traité des relations inter-classes, à savoir comment est réestimé le critère de regroupement entre groupes de segments après chaque regroupement. Nous testons les quatre possibilités évoquées dans l'introduction :

- *min* : estimation par paire minimale
- *max* : estimation par paire maximale
- *moyenne* : estimation par paire moyenne
- *mise à jour* : estimation "complète"

Dans le graphe 9.13 (tableau E.13) qui fournit les résultats de ces tests pour les données CNET, nous constatons que c'est la méthode *mise à jour* qui fournit les meilleurs résultats en termes de nombre de locuteurs obtenus (21.7 pour 17.6), la méthode *max* qui fournit les meilleurs résultats en termes de pureté (80%) et la méthode *min* qui fournit les meilleurs résultats en termes de durée moyenne (14.1 s). Les différences de résultats entre les quatre méthodes ne sont pas énormes : le nombre de locuteurs évolue de 21.7 à 28.5, la pureté de 78.3% à 80.0% et la durée de 8.7 s à 14.1 s. Néanmoins, la méthode *mise à jour* est le meilleur compromis avec un nombre de locuteurs de 21.7, une pureté de 79% et une durée moyenne de 10.1 s.

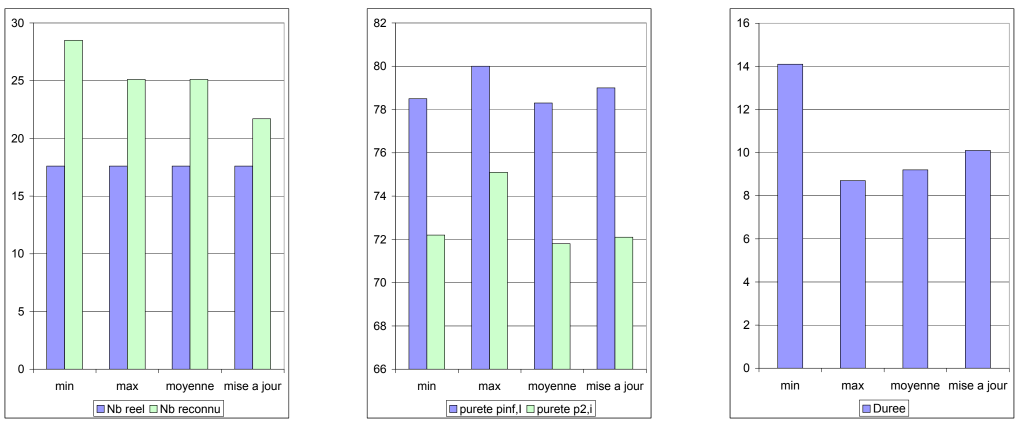

FIG. 9.13 – *Données CNET : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 0.8$)*

Pour les données TIMIT (cf graphe 9.14 et tableau E.14), la méthode *mise à jour* donne les meilleurs résultats en termes de nombre de locuteurs reconnus (36.5 pour 23.6 réels), et en termes de durée moyenne (8.6s). c'est par contre la méthode *max* qui obtient la pureté maximale avec 98%. Les disparités entre les différentes méthodes sont plus fortes que pour les données CNET : le nombre de locuteurs trouvés varie de 36.5 à 65.5 et la pureté de 90.5% à 98%. D'ailleurs, la plus faible pureté (90% quand même) correspond au nombre de locuteurs détectés le plus faible et, à l'inverse, la pureté la plus élevée correspond au nombre de locuteurs trouvés le plus élevé. La méthode *mise à jour* est à nouveau le meilleur compromis avec un nombre de locuteurs trouvés de 36.5 (pour 23.6), une pureté de 90% et une durée moyenne de 8.6 s.

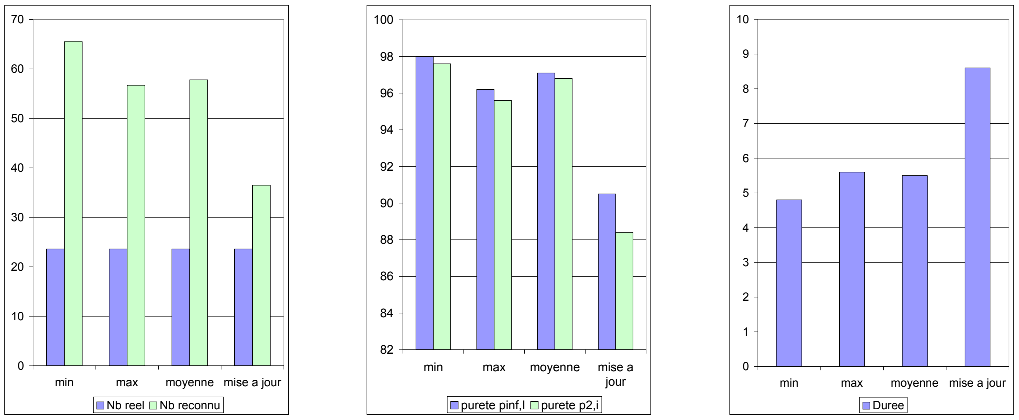

FIG. 9.14 – *Données TIMIT : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 0.8$)*

Les données DIAL constituent des données un peu particulières pour tester la réévaluation du critère de regroupement. Ces dialogues ne mettant en jeu que deux personnes, les distances minimale, maximale, moyenne et mise à jour entre les deux locuteurs devraient présenter moins de variations qu'entre plusieurs locuteurs. D'ailleurs, le nombre de locuteurs varie peu : il est de 3 pour la méthode *min* et de 11 pour la méthode *moyenne*. La pureté évolue de 99.4% pour la méthode *min* à 100% pour la méthode *mise à jour*. Quant à la durée moyenne, elle passe de 104.4 s (méthode *mise à jour*) à 122.1 s (méthode *min*). Cette fois, la méthode *min* fournit les meilleurs résultats pour les données DIAL.

Le graphe 9.16 (tableau E.16) expose les résultats obtenus par ces 4 méthodes de réevaluation du critère de regroupement sur les données CONV. Les meilleurs résultats sont fournis par la méthode *mise à jour* en termes de nombre reconnu de locuteurs (14.3 pour 10) et de durée moyenne (56.6 s). La méthode *min* atteint la plus forte pureté avec 97.5%. Ce résultat est cependant à nuancer par le grand nombre de locuteurs obtenus en parallèle : 98.8 pour 10 réels. Comme pour les données TIMIT et CNET, la méthode *mise à jour* correspond au meilleur compromis avec un nombre de locuteurs trouvés de 14.3, une pureté de 92.4% et une durée moyenne de 56.6s.

En résumé, la méthode *mise à jour* fournit les meilleurs résultats pour les données TIMIT, CNET et DIAL. Par contre, la méthode *min* est la plus performante pour les données DIAL. Cependant, il se peut que ce dernier résultat soit biaisé par le fait que seuls deux locuteurs interviennent dans ces dialogues.

Par ailleurs, nous n'avons pas fait de tests sur les temps d'exécution de ces différentes méthodes puisque nous n'avons pas de contrainte de temps réel. Cependant, nous pouvons affirmer que, bien que la méthode *mise à jour* soit la plus performante dans la majorité des cas, elle est aussi la plus coûteuse en temps de calculs étant donné le grand nombre d'opérations

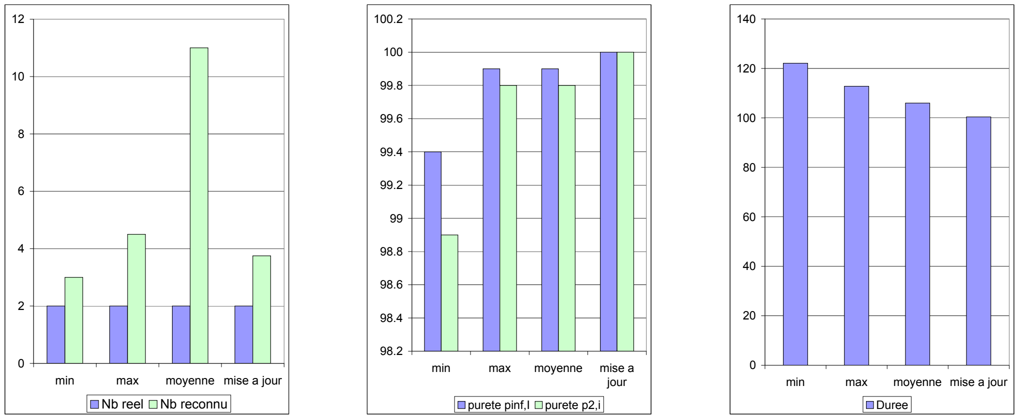

FIG. 9.15 – *Données DIAL : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

à réaliser par rapport aux autres méthodes.

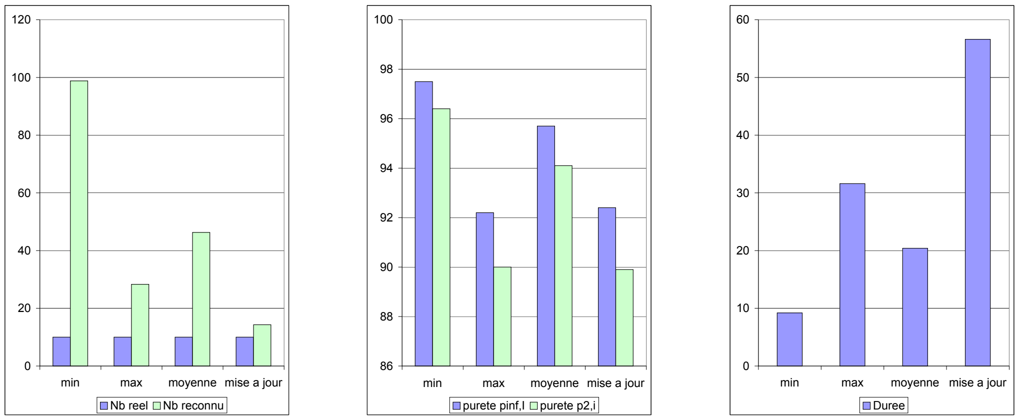

FIG. 9.16 – *Données CONV : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

Influence de la présence de silences

Dans ce paragraphe, nous analysons l'influence de la présence de silences significatifs (i.e. d'une longueur allant de quelques dixièmes de seconde à quelques secondes) sur les performances de l'algorithme de regroupement hiérarchique testé. Les graphes 9.17 et 9.18 (tableaux E.17 et E.18) détaillent les performances du regroupement hiérarchique sur, respectivement, les dialogues DIAL avec et sans silences, et sur les conversations avec et sans silences, pour une dimension de vecteurs acoustiques de 16 et une valeur de λ de 1.5.

Concernant les dialogues DIAL, la partition résultante du regroupement hiérarchique est quasi-parfaite (6 et 5 locuteurs reconnus pour 2 réels avec des puretés de 100%) pour les dialogues contenant des silences. Ces partitions sont parfaites lorsque les silences sont supprimés. Ces excellents résultats s'expliquent tout d'abord par le fait qu'il n'y a que deux locuteurs : le regroupement se trouve donc simplifié. Les groupes de locuteurs en surplus, obtenus avec des conversations contenant des silences, sont en fait des groupes de segments de silence.

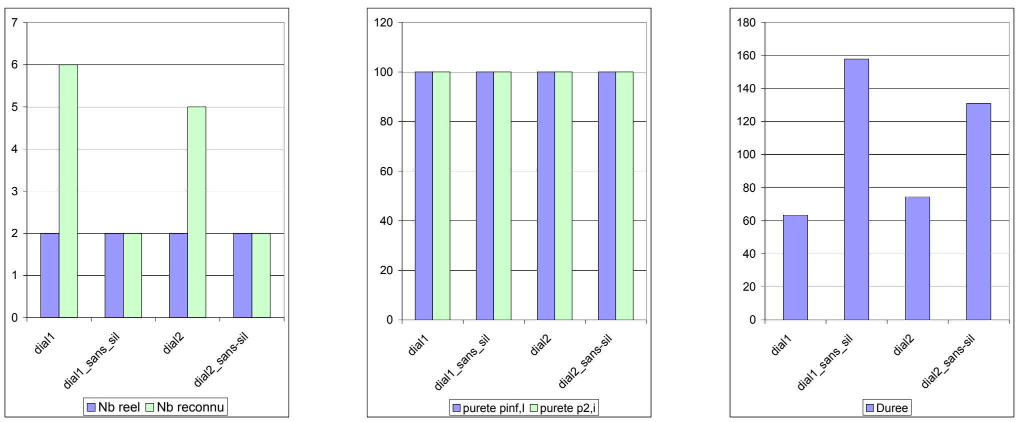

FIG. 9.17 – *Données DIAL : influence de la présence de silences (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

Pour les conversations (cf graphe 9.18 et tableau E.18), le nombre de locuteurs trouvé diminue également entre les conversations avec silences et les conversations sans silences. Une étude plus détaillée des résultats montre que le surplus de locuteurs pour les conversations avec silences correspond également à des groupes de segments de silences. Nous constatons par contre pour *conv1* une baisse de la pureté : celle-ci est de 89.5% pour *conv1* avec silences et de 86.7% pour *conv1* sans silences. Cela signifie que les groupes de silences obtenus dans le premier cas ont une pureté supérieure à la pureté moyenne. Ce n'est pas le cas pour *conv2* car la pureté passe d'une valeur de 95.7% avec silences à 97.7% sans silences.

En conclusion, la présence de silences - significatifs en termes de durée - ne perturbe pas le regroupement hiérarchique. En effet, les segments de silences se regroupent. Par contre, ils ne se regroupent pas au sein d'un seul et même groupe, ce qui aboutit à un nombre de groupes de segments plus important que le nombre réel de locuteurs. Ceci amènerait dans l'étape suivante à construire plusieurs modèles de silence. Cela impliquerait une surcharge de calculs d'une part et cela aboutirait peut-être à une mauvaise modélisation du fait de la

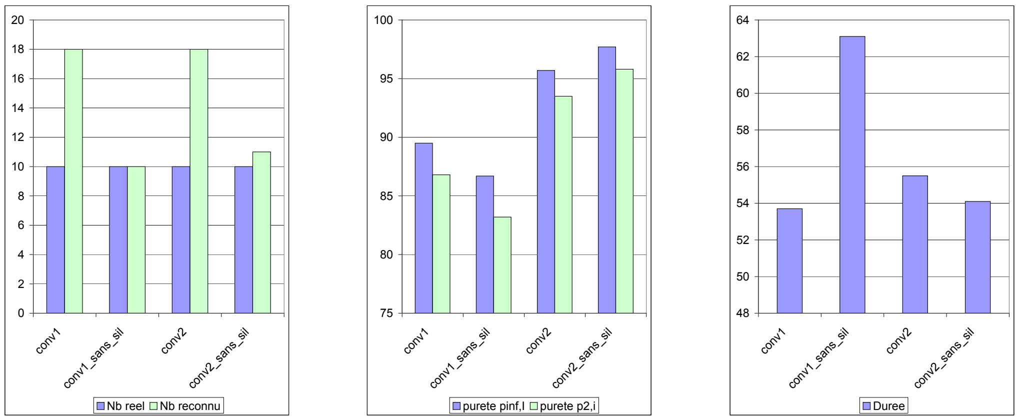

FIG. 9.18 – *Données CONV : influence de la présence de silences (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

dispersion des données de silence. Il est donc plus intéressant de les supprimer par exemple à l'aide de la méthode de détection de silences SILHYST que nous proposons au chapitre 4.

Conclusions

En guise de conclusion sur le regroupement hiérarchique appliqué à des segments purs et sur l'étude des paramètres intervenant dans cette technique de regroupement, nous rappelons les résultats obtenus :

- une dimension de 16 sans utilisation des coefficients Δ pour l'espace acoustique est recommandée quel que soit le type de données
- la valeur de λ intervenant dans le critère BIC est en relation directe avec la longueur réelle des segments
- l'efficacité du pré-traitement et du post-traitement des segments courts ne pourra être réellement prouvée que lorsque l'étape de modélisation des locuteurs et de la reconnaissance de la séquence de locuteurs sera mise ne place
- la meilleure méthode pour évaluer la distance entre les groupes de segments est la méthode par mise à jour
- enfin, l'élimination préalable des silences significatifs tend à améliorer les résultats du regroupement

Forts de ces enseignements, nous étudions à la section suivante les performances de ce même algorithme appliqué à des segments issus d'une segmentation préalable.

9.2.2 Evaluation avec des segments résultant de la segmentation

Dans cette section, nous présentons les résultats du même regroupement hiérarchique (cf chapitre 8) appliqué à des segments qui résultent d'une segmentation préalable. Nous segmentons de deux manières :

- soit en appliquant notre méthode de segmentation DISTBIC (cf section 4.2)
- soit en appliquant notre méthode de segmentation DISTBIC, combinée avec une détection de silences préliminaire à l'aide de notre algorithme SILHYST (cf section 4.1)

Comme nous l'avons vu au chapitre 5, les segments issus de la segmentation ne sont pas aussi purs (i.e. ne contiennent pas les paroles que d'un seul locuteur) que les segments de référence. Il faut donc probablement s'attendre à de moins bons résultats du regroupement hiérarchique.

Description des données

Nous testons l'algorithme de regroupement hiérarchique, et indirectement la segmentation, sur les même données qu'à la section 9.2.1 : 10 conversations TIMIT, 10 conversations CNET, 2 dialogues DIAL et 2 conversations CONV. Pour ces deux derniers types de données, nous n'utilisons plus les conversations sans silences significatifs.

Ayant évalué le regroupement hiérarchique sur ces mêmes données mais sur les segments de référence, les tests qui suivent nous permettent de mettre en évidence les baisses de performances directement liées aux erreurs de segmentation. Les conversations précédentes étant synthétiques (rappelons qu'elles sont créées en concaténant des phrases de différents locuteurs), nous testons également le regroupement hiérarchique sur des données réelles :

- 3 journaux télévisés français enregistrés dans notre laboratoire (parole préparée et spontanée, 126 minutes). Ces journaux télévisés sont référencés JT par la suite.
- 49 conversations téléphoniques issues de la base de données SWITCHBOARD (une description de la base de données se trouve dans [Godfrey et al. 92]) (américain, parole spontanée, durée : de 5 à 10 minutes par conversation). Ces conversations sont référencées par la suite SWB.

Conformément aux conclusions de la section précédente, nous utilisons comme paramétrisation pour le regroupement hiérarchique des vecteurs acoustiques composés de 16 coefficients Mel-cepstraux. De même, conformément aux résultats obtenus sur la segmentation, des vecteurs acoustiques composés de 12 coefficients Mel-cepstraux sont utilisés pour la segmentation. Ces dimensions ne sont plus précisées par la suite.

Les graphes de résultats sont formés comme précédemment (cf description page 91). Dans les légendes des graphes, nous mentionnons les valeurs des paramètres qui interviennent dans les différents traitements et qui sont susceptibles d'être modifiés. Ces paramètres sont les suivants :

- pour SILHYST, la durée minimale des silences (cf section 4.1)
- pour DISTBIC, la valeur du paramètre λ intervenant dans le critère BIC à la seconde passe (cf section 4.2)
- pour le regroupement hiérarchique, la valeur du paramètre λ intervenant dans le critère d'arrêt BIC

Comparaison de la segmentation de référence, de la segmentation DISTBIC et de la combinaison des segmentations SILHYST et DISTBIC

Dans ce paragraphe, nous comparons les résultats du regroupement hiérarchique précédé de différents types de segmentations : REFERENCE (segmentation de référence), DISTBIC et la combinaison SILHYST+DISTBIC. Nous utilisons pour les méthodes de segmentation en locuteurs les valeurs de paramètres recommandées à l'issue du chapitre 5. De même, pour le regroupement, nous prenons les valeurs des paramètres qui ont fourni les meilleurs résultats pour le regroupement des segments de référence.

En ce qui concerne les données CNET (cf graphe 9.19 et tableau E.19), nous constatons, comme nous pouvions nous y attendre, une baisse des performances entre la segmentation de référence et les deux autres segmentations. Le nombre reconnu de locuteurs est en hausse (par rapport au nombre réel des locuteurs) et la pureté et la durée moyenne sont à l'inverse en baisse. Ces baisses de performances se retrouvent également pour les données TIMIT (graphe 9.20 et tableau E.20), DIAL (graphe 9.21 et tableau E.21) et CONV (graphe 9.22 et tableau E.22).

Pour en revenir aux données CNET, cette baisse de performances se ressent surtout pour le nombre reconnu de locuteurs : celui-ci est de 21.7 pour la segmentation de référence (pour un nombre réel de 17.6) et de 34.7 pour les segmentations SILHYST+DISTBIC et DISTBIC seule, pour des puretés pratiquement égales : 79%, 77.6% et 78.2% respectivement. Le nombre reconnu de locuteurs étant plus important, la durée moyenne baisse : elle passe de 10.1 s pour REFERENCE à 6.6 s pour les deux autres segmentations.

Les résultats obtenus avec SILHYST+DISTBIC sont légèrement moins bons qu'avec DISTBIC seule. Le nombre reconnu de locuteurs et la durée sont égaux mais la pureté perd 0.6%.

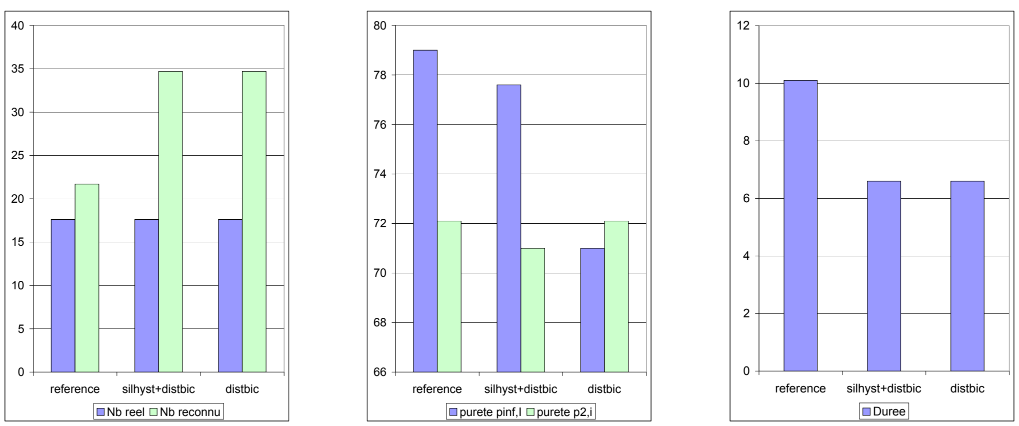

FIG. 9.19 – *Données CNET : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.15 s, segmentation :* $\lambda = 1.0$*, regroupement :* $\lambda = 0.8$*)*

Le graphe 9.20 (tableau E.20) présente les résultats des trois types de segmentations suivies du regroupement hiérarchique pour les données TIMIT. Contrairement aux données CNET, le nombre de locuteurs reconnus et la durée moyenne restent relativement stables entre les 3 types de segmentation : le premier varie de 36.0 à 38.1 pour un nombre réel de 23.6 et la deuxième de 8.3s à 8.7s. Par contre, la pureté se dégrade entre la segmentation de référence (88.4%) et les deux autres types de segmentation (81.7%). Cette baisse de pureté est liée au fait que les segments résultant des deux derniers types de segmentation sont moins purs que la segmentation de référence.

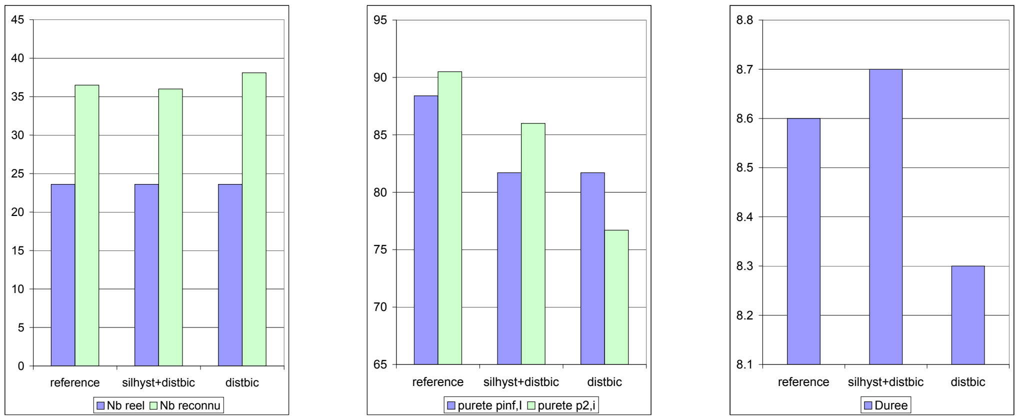

FIG. 9.20 – *Données TIMIT : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation :* $\lambda = 1.2$*, regroupement :* $\lambda = 0.8$*)*

Les résultats du regroupement précédé des 3 types de segmentation sur les données DIAL (graphe 9.21 et tableau E.21) et sur les données CONV (graphe 9.22 et tableau E.22) évoluent de la même manière. Entre la segmentation de référence et les deux autres segmentations, la pureté diminue et la durée moyenne augmente. Par contre, le nombre reconnu de locuteurs baisse dans le cas des données DIAL et augmente dans le cas des données CONV.

Pour les données DIAL, la baisse des performances des deux derniers types de segmentation peut s'expliquer par un trop fort regroupement (bien que le nombre de locuteurs reconnus reste supérieur au nombre réel) impliquant ainsi une baisse de la pureté. Les segments issus de la segmentation sont sans doute plus courts et les valeurs des paramètres ne sont plus adaptées à cette longueur de segments. Nous avons vu en effet que la valeur du poids de pénalité λ (que ce soit pour la segmentation ou pour le regroupement) dépendait essentiellement de la longueur réelle des segments.

Les différences de performances existant entre SILHYST+DISTBIC et DISTBIC pour les données DIAL et CONV sont dues à l'absence des silences dans le premier cas. Ces silences sont supprimés après avoir été détectés par SILHYST. Or, nous avions vu que dans le regroupement des segments de référence, la suppression de ces silences pouvait amener à une baisse de la pureté.

Mises à part les données CNET, le nombre de locuteurs obtenus suite au regroupement avec la segmentation de référence et avec la segmentation DISTBIC est quasiment égal pour les autres types de données. Dans les 3 cas, nous pouvons également remarquer que pour les données TIMIT, la baisse de pureté est d'environ 7%, pour les données DIAL d'environ 8% et pour les données CONV de 7%. C'est donc dans les mêmes proportions qu'a lieu la diminution de la pureté pour ces trois types de données.

Pour comparer le regroupement hiérarchique précédé des différentes méthodes de seg-

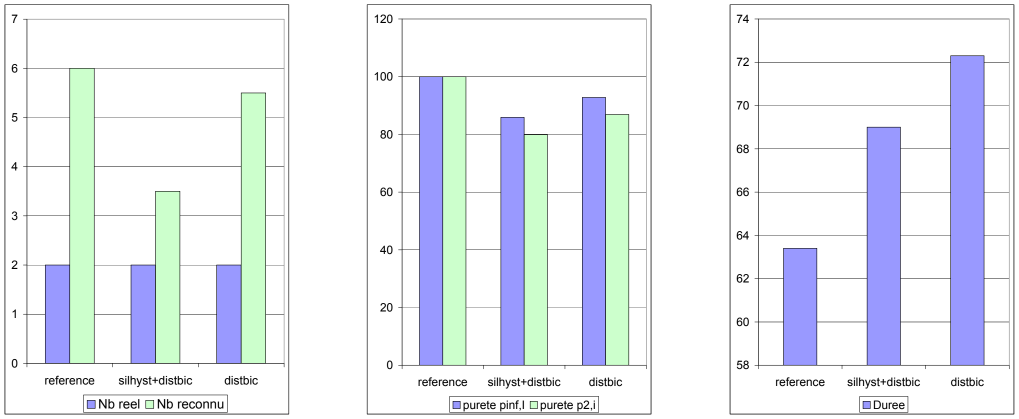

FIG. 9.21 – *Données DIAL : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.5$, regroupement : $\lambda = 1.5$)*

mentation, nous avons choisi les valeurs des paramètres en fonction des caractéristiques des données (notamment la longueur réelle des segments), valeurs déduites de nos précédentes expériences. Ces valeurs ont été évaluées sur des expériences séparées : la segmentation d'une part et le regroupement d'autre part. Il se peut que l'enchaînement des deux processus entraîne une modification des valeurs optimales des paramètres. Nous y voyons au moins deux raisons. D'une part, les valeurs sélectionnées correspondent aux valeurs pour la longueur réelle des segments dans la conversation et non la longueur des segments résultant de la segmentation. D'autre part, pour la segmentation, nous avons fait un compromis sur le taux de détections manquées et le taux de fausses alarmes (cf section 5.1). Une détection manquée étant plus préjudiciable qu'une fausse alarme pour le regroupement, peut-être faut-il choisir un autre "point de fonctionnement" ?

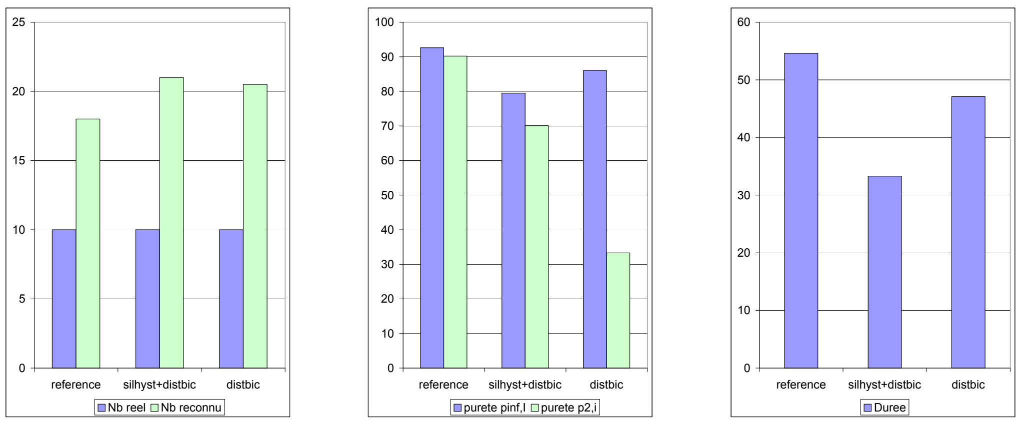

FIG. 9.22 – *Données CONV : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation :* $\lambda = 1.5$*, regroupement :* $\lambda = 1.5$*)*

Influence du poids de pénalité de la segmentation

Dans ce paragraphe, nous nous intéressons à la valeur du poids de pénalité de la segmentation. En particulier, nous voudrions savoir s'il y a lieu de modifier celle-ci quand la segmentation est suivie du regroupement. La valeur optimale trouvée à l'issue de la segmentation résulte d'un compromis entre le taux de détections manquées (TDM) et le taux de fausses alarmes (TFA). Nous avons fait le choix d'avoir un TDM aussi faible que possible, tout en préservant un TFA à un niveau faible. Peut-être est-il nécessaire pour obtenir de meilleurs résultats pour le regroupement de choisir un taux de détections manquées encore plus faible avec un taux de fausses alarmes plus fort, quitte à obtenir de très courts segments qui ne sont pas pris en compte lors du regroupement.

Les données TIMIT étant composées de courts segments, la valeur recommandée pour le poids de pénalité de la segmentation est de 1.2. Nous testons une valeur plus faible, i.e. $\lambda = 1.0$, pour obtenir éventuellement de plus courts segments plus purs. Le graphe 9.23 (tableau E.23) relate ces expériences. Nous constatons que diminuer la valeur de λ de 1.5 à 1.2 aboutit au contraire à une dégradation des résultats : pour des nombres de locuteurs obtenus et des durées moyennes quasiment égaux, la pureté perd 5% : elle passe de 86.0% à 80.9%. Nous ne testons pas une valeur plus élevée de λ car nous savons d'après les résultats sur la segmentation que cela aboutirait à trop de regroupements et donc à des segments, certes plus longs, mais aussi plus impurs. La valeur de $\lambda = 1.2$ pour la segmentation semble se confirmer pour les conversations contenant de courts segments.

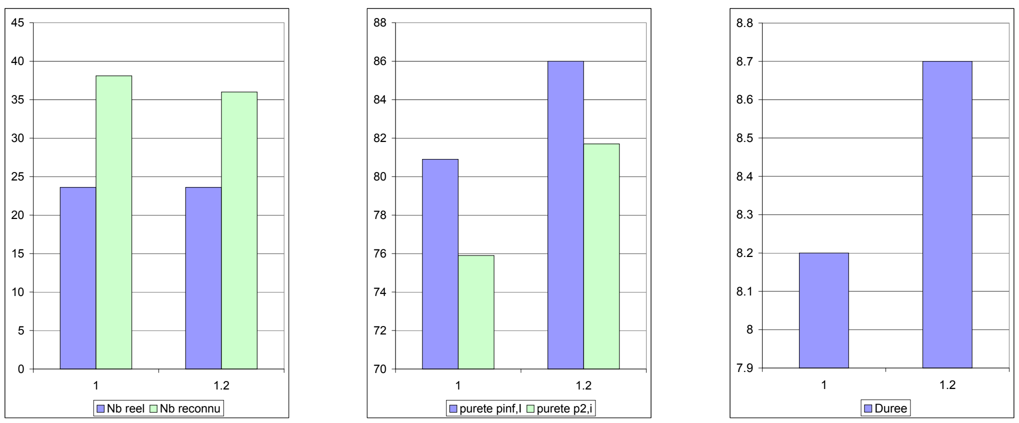

FIG. 9.23 – *Données TIMIT : influence du poids de pénalité de la segmentation SILHYST + DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement : $\lambda = 0.8$)*

Les données DIAL et CONV sont des conversations contenant de longs segments. La valeur recommandée pour le poids de pénalité de la segmentation est de 1.5 (cf chapitre 5). Comme pour les données TIMIT, nous testons une valeur plus faible de λ par rapport à la valeur "recommandée" pour obtenir éventuellement des segments plus courts et plus purs à l'issue de la segmentation et aboutir à une meilleure partition suite au regroupement.

Le graphe 9.24 (tableau E.24) mentionne les résultats du regroupement hiérarchique obtenus avec des valeurs de λ de 1.2 et de 1.5 respectivement pour la segmentation combinée SILHYST+DISTBIC. Les partitions obtenues sont très proches : le nombre de locuteurs obtenus est légèrement plus grand pour $\lambda = 1.2$ (4.0 au lieu de 3.5 pour $\lambda = 1.5$). Par contre, la pureté et la durée moyenne s'améliorent, également dans une faible mesure : la pureté passe de 85.9% pour $\lambda = 1.5$ à 87.4% pour $\lambda = 1.2$ et la durée moyenne de 69.0s à 71.4s. Il nous est donc difficile de conclure sur le meilleur choix de la valeur du poids de pénalité dans le cas présent. Quoiqu'il en soit, les deux valeurs aboutissent toutes deux à de bonnes partitions.

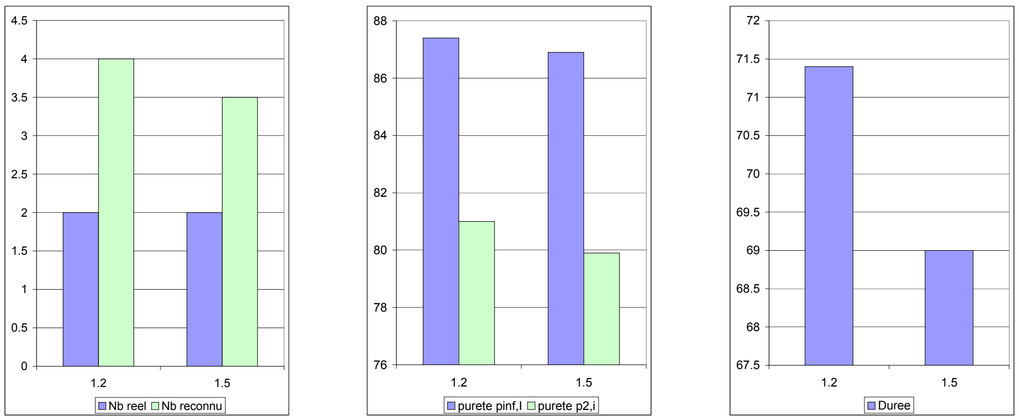

FIG. 9.24 – *Données DIAL : influence du poids de pénalité de la segmentation SILHYST + DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement : $\lambda = 1.5$)*

Le graphe 9.25 (tableau E.25) donne les résultats du regroupement hiérarchique précédé de DISTBIC pour des valeurs différentes du poids de pénalité intervenant dans la segmentation. Contrairement au graphe précédent, les résultats obtenus avec la valeur de λ de 1.5 sont meilleurs que ceux obtenus avec la valeur de 1.2. Le nombre reconnu de locuteurs est plus proche du nombre réel, la pureté et la durée moyenne sont plus élevées. Cependant, les différences constatées entre les résultats pour les deux valeurs du poids de pénalité sont faibles. Aussi, les deux valeurs sont valables, à condition toutefois que les différences de performances obtenues à l'étape suivante (modélisation des locuteurs et reconnaissance de la séquence de locuteurs) restent dans de faibles proportions.

Dans le graphe 9.26 (tableau E.26) sont exposés les résultats du regroupement précédé de la combinaison SILHYST+DISTBIC pour les données CONV. A nouveau, les valeurs de 1.2 et de 1.5 pour le poids de pénalité de la segmentation sont testées. Dans les deux cas, le nombre de locuteurs reconnus est inférieur au nombre réel (respectivement, 19 et 21 pour 23.6 réels). La pureté et la durée moyenne sont plus élevées pour $\lambda = 1.2$ mais cela se fait au détriment du nombre de locuteurs qui diminue et qui se trouve en dessous du nombre réel. Cela signifie qu'à l'étape suivante, certains locuteurs n'auront pas de modèle et, par conséquent, ne pourront être reconnus dans la séquence.

En conclusion, il semble que la valeur optimale du poids de pénalité trouvée à l'issue de la segmentation se confirme tant pour les conversations contenant de courts segments, que pour les conversations contenant de longs segments. Cela signifie également que le compromis fait entre le taux de détections manquées et le taux de fausses alarmes est un bon compromis. Cela justifie également l'emploi de la seconde passe basée sur le BIC qui vise à réduire les fausses alarmes. En effet, la baisse du poids de pénalité pour la segmentation amène à une

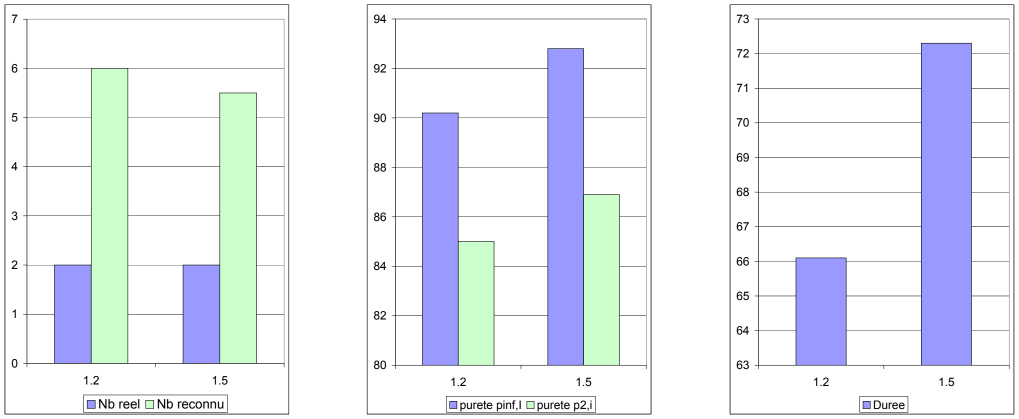

FIG. 9.25 – *Données DIAL : influence du poids de pénalité de la segmentation DISTBIC suivie du regroupement (regroupement : $\lambda = 1.5$)*

sur-segmentation (i.e. un TFA élevé). Cette hausse du TFA entraîne d'après ce que nous venons de voir, des modifications de la partition résultante qui ne vont pas forcément dans le sens souhaité. Dans les expériences que nous venons de relater, ces modifications sont faibles et auront sans doute un impact limité pour la suite, mais la hausse du TFA est également faible.

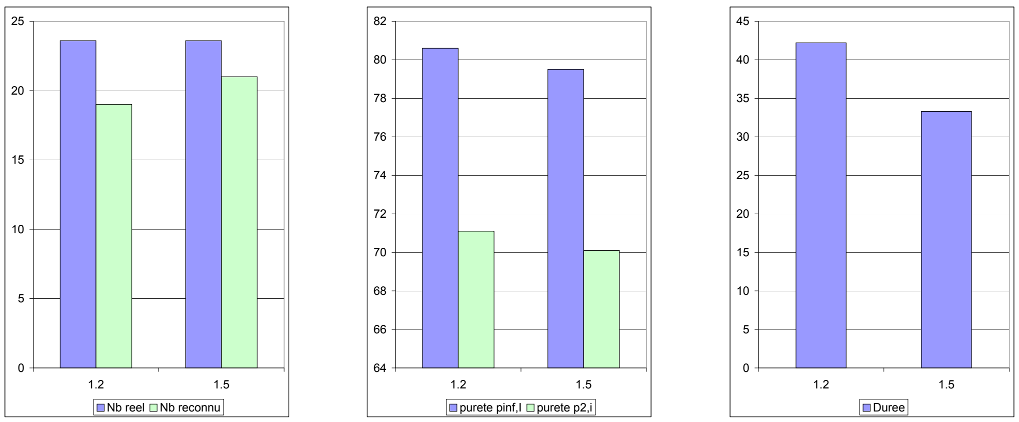

FIG. 9.26 – *Données CONV : influence du poids de pénalité de la segmentation SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement :* $\lambda = 1.5$*)*

Influence du poids de pénalité du regroupement

Dans ce paragraphe, nous faisons varier le paramètre λ du critère BIC du regroupement hiérarchique pour les différents types de de données.

D'après la segmentation de référence, la valeur optimale de λ pour les données CNET est de 0.8. Nous venons de voir qu'avec cette valeur pour le regroupement précédé de segmentations en locuteurs, cela aboutissait à un grand nombre de locuteurs (34.7 locuteurs trouvés pour 17.6 réels). En augmentant la valeur de λ, un plus grand nombre de regroupements s'effectuent et par conséquent, il y a moins de locuteurs trouvés. Les graphes 9.27 et 9.28 (tableaux E.27 et E.28) présentent les résultats du regroupement avec des poids de pénalité différents pour la segmentation SILHYST+DISTBIC et la segmentation DISTBIC.

L'augmentation de λ de 0.8 à 1.0 entraîne une baisse sensible du nombre de locuteurs pour les deux segmentations associées au regroupement : dans les deux cas, le nombre de locuteurs trouvés passe de 34.7 à 22 (toujours pour 17.6 réels). Cette baisse du nombre de locuteurs s'accompagne d'une baisse de la pureté : celle-ci passe de 77.6% à 69.7% pour la segmentation SILHYST+DISTBIC et le regroupement et de 78.2% à 69.9% pour la segmentation DISTBIC et le regroupement. Cette baisse de la pureté se fait dans les mêmes proportions dans les deux cas (environ -8%). De même, la durée moyenne augmente dans les mêmes proportions passant de 6.6s à une dizaine de secondes dans les deux cas.

Nous revenons cependant à la sempiternelle question, non résolue et dépendant de l'application, à savoir s'il vaut mieux une partition avec beaucoup de locuteurs par rapport au nombre réel et avec une certaine pureté ou une partition contenant beaucoup moins de locuteurs mais d'une pureté plus faible.

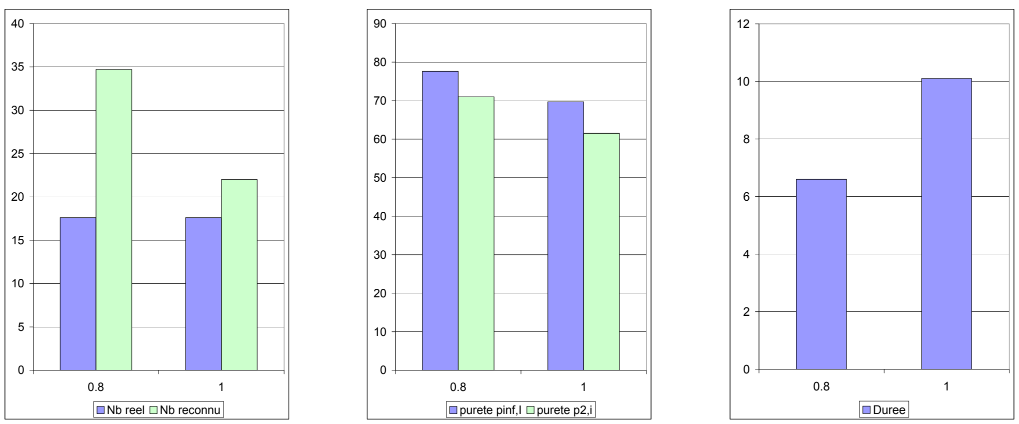

FIG. 9.27 – *Données CNET : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.15 s, segmentation :* $\lambda = 1.0$*)*

Des commentaires analogues peuvent être faits sur les résultats du regroupement appliqué aux données TIMIT. En effet, en augmentant la valeur de λ de 0.8 à 1.0, le nombre reconnu de locuteurs diminue de 36.0 à 23.3 (pour 23.6 réels) pour SILHYST+DISTBIC+regroupement et de 38.1 à 24.7 pour DISTBIC+regroupement. En parallèle, la pureté diminue de 86.0% à 77.8% pour SILHYST+DISTBIC+regroupement et de 81.7% à 74.8% pour la combinai-

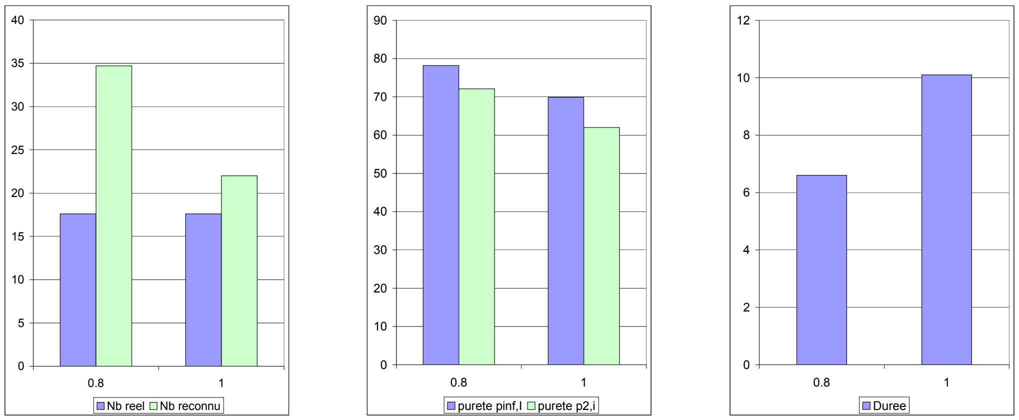

FIG. 9.28 – *Données CNET : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.0$))*

son DISTBIC+regroupement. Cette baisse varie de 9% à 7% pour les deux cas respectifs, pourcentages à rapprocher des 8% obtenus avec les données CNET.

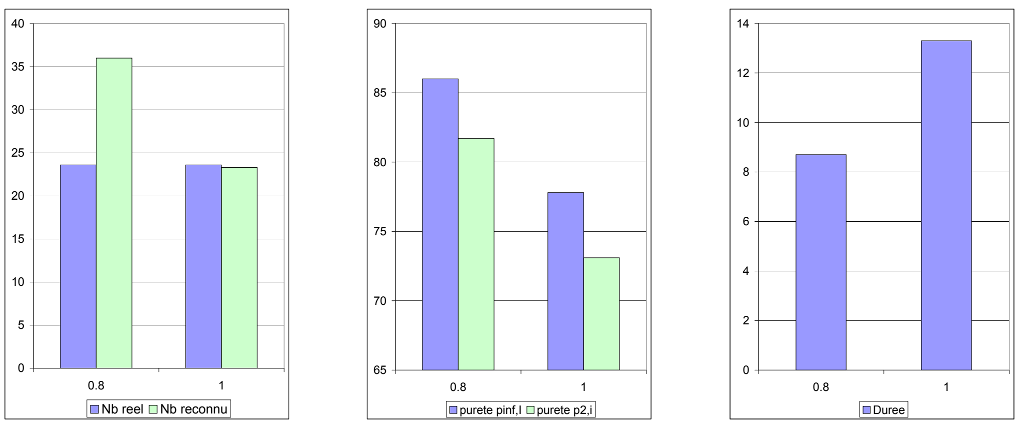

FIG. 9.29 – *Données TIMIT : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.2$)*

Pour les données DIAL et CONV, nous testons l'effet d'une diminution de la valeur du poids de pénalité pour le regroupement. A l'issue des expériences menées sur les segments de référence, une valeur de 1.5 semblait être optimale pour le regroupement. Cette valeur se confirme car le regroupement avec un poids de pénalité de 1.2 aboutit à de moins bonnes partitions qu'avec une valeur de 1.5 en termes de nombre de locuteurs reconnus, de pureté et de durée moyenne. Nous pouvons le constater pour les données DIAL dans le graphe 9.31 (tableau E.31) pour la segmentation combinée suivie du regroupement, et dans le graphe

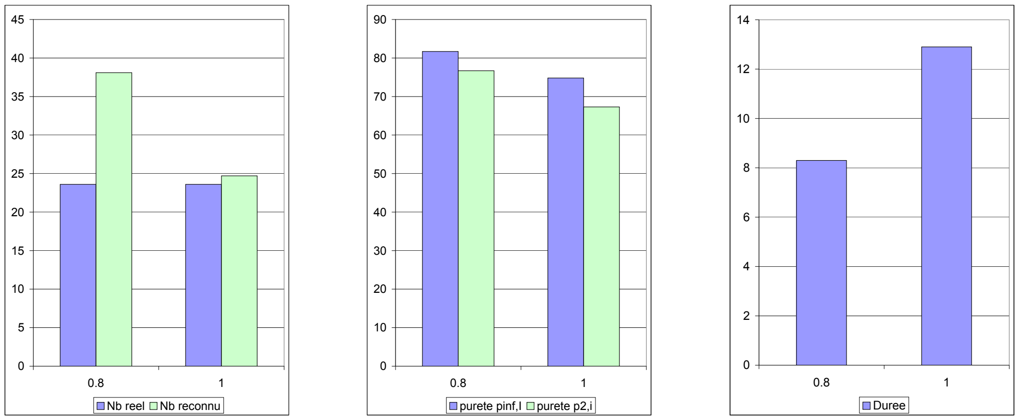

FIG. 9.30 – *Données TIMIT : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.2$)*

9.32 (tableau E.32) pour la segmentation DISTBIC suivie du regroupement. De même, les graphes pour les données CONV 9.33 et 9.34 (tableaux E.33 et E.34) respectivement pour la segmentation combinée suivie du regroupement et pour la segmentation DISTBIC suivie du regroupement confirment ces résultats.

FIG. 9.31 – *Données DIAL : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.5$)*

Les valeurs optimales du poids de pénalité trouvées pour les segments de référence restent identiques pour le regroupement précédé d'une segmentation en locuteurs. Cette valeur est de 0.8 à 1.0 pour les conversations contenant des segments de courte durée (seule la mise ne œuvre de l'étape suivante nous permettra de trancher définitivement) et de 1.5 pour les conversations contenant de plus longs segments. Cependant, une baisse d'environ 8% est constatée sur la

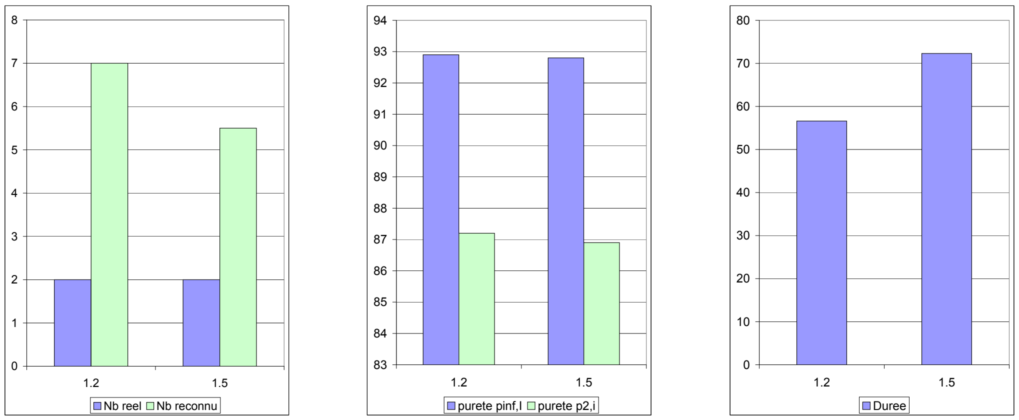

FIG. 9.32 – *Données DIAL : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.5$)*

pureté entre le regroupement sur les segments de référence et le regroupement précédé d'une segmentation en locuteurs.

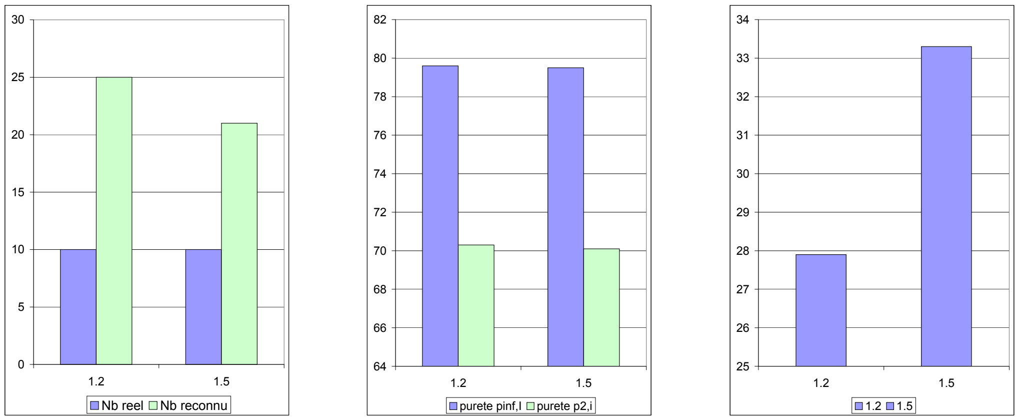

FIG. 9.33 – *Données CONV : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation :* $\lambda = 1.5$*)*

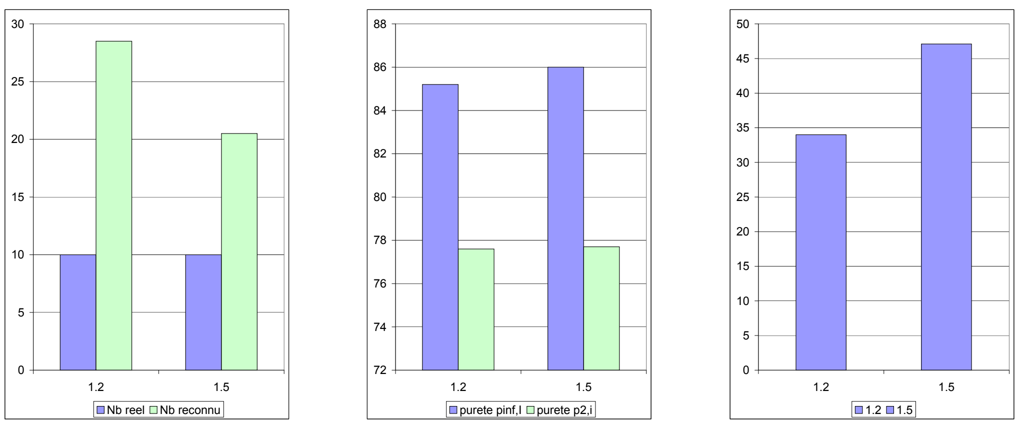

FIG. 9.34 – *Données CONV : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.5$*)*

Journaux télévisés JT

Dans ce paragraphe, nous présentons les résultats du regroupement obtenus sur les journaux télévisés JT. Pour évaluer les résultats, nous utilisons deux types d'indexations de référence :

- le premier type d'indexation de référence considère comme locuteur une personne quel que soit le fond sonore ou une sorte de fond sonore comme le silence ou le bruit. Un journaliste dans un studio ou en reportage dans la rue, ne bénéficie pas des mêmes conditions acoustiques. Cependant, il est considéré comme ne formant qu'un seul et même locuteur. Par la suite, cette indexation est référencée *évaluation de type I* ou *indexation de type I*.
- Le deuxième type d'indexation de référence procède inversement. Un locuteur signifie un locuteur dans un environnement sonore particulier. Aussi, le journaliste dans le studio et le journaliste dans la rue sont considérés comme deux locuteurs différents. Cette indexation distingue également des fonds sonores tels que le silence ou le bruit. Dans ce qui suit, cette indexation est désignée *évaluation de type II* ou *indexation de type II*.

Nous nous intéressons tout d'abord à l'*évaluation de type I*. Le graphe 9.35 (tableau E.35) présente les résultats du regroupement précédé de la combinaison SILHYST+DISTBIC. La segmentation aurait dû être réalisée avec une valeur de 1.5 pour le poids de pénalité, étant donnée la longueur des segments réels. Cependant, comme vu au chapitre 5, des erreurs de détections manquées ont lieu sur les segments courts en particulier, lors des reportages. D'où l'utilisation d'une valeur de 1.2 pour le λ de la segmentation. Nous testons dans ce graphe deux valeurs de poids de pénalité pour le regroupement. La valeur de 1.5 semble donner de meilleurs résultats car le nombre reconnu de locuteurs est proche du nombre réel (72.3 pour 71.5), la durée moyenne est de 33.2s et la pureté est de 72.3%.

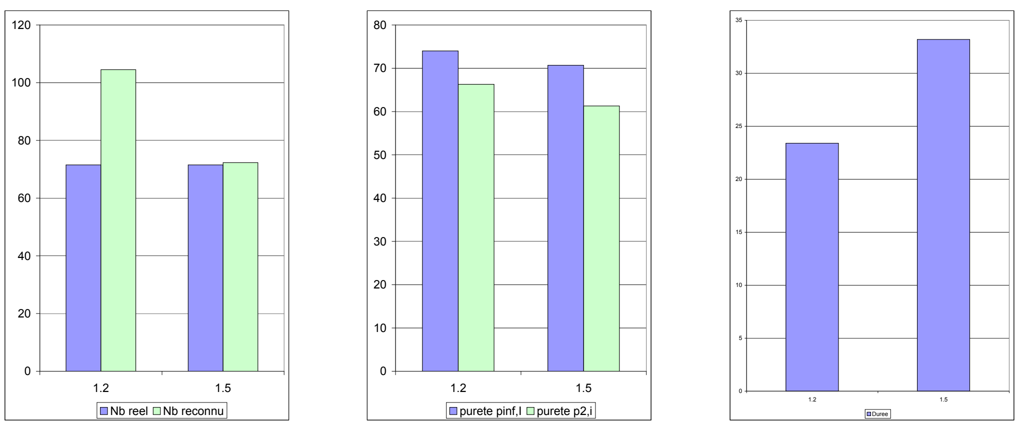

FIG. 9.35 – *Données JT (indexation type I) : segmentation SILHYST+DISTBIC suivie du regroupement (silhyst: durée=0.3s, segmentation :* $\lambda = 1.2$*)*

Les graphes 9.36 et 9.37 (tableaux E.36 et E.37) exposent les résultats du regroupement précédé de DISTBIC pour les JT, toujours avec l'*indexation de type I*. Chaque graphe correspond à une valeur du poids de pénalité de la segmentation : $\lambda = 1.5$ pour le premier graphe et $\lambda = 1.2$ pour le deuxième. Par ailleurs, chaque graphe teste plusieurs valeurs du poids de pénalité, mais cette fois-ci pour le regroupement. Pour les deux graphes, donc pour les deux segmentations, c'est la valeur de 1.2 pour le λ de regroupement qui fournit le meilleur résultat en termes de nombre reconnu de locuteurs. Les puretés sont quant à elles très proches de la valeur maximale obtenue : 76.4% pour une valeur maximale de 76.9% dans la première segmentation et 76.2% pour une valeur maximale de 77.2% pour la deuxième segmentation.

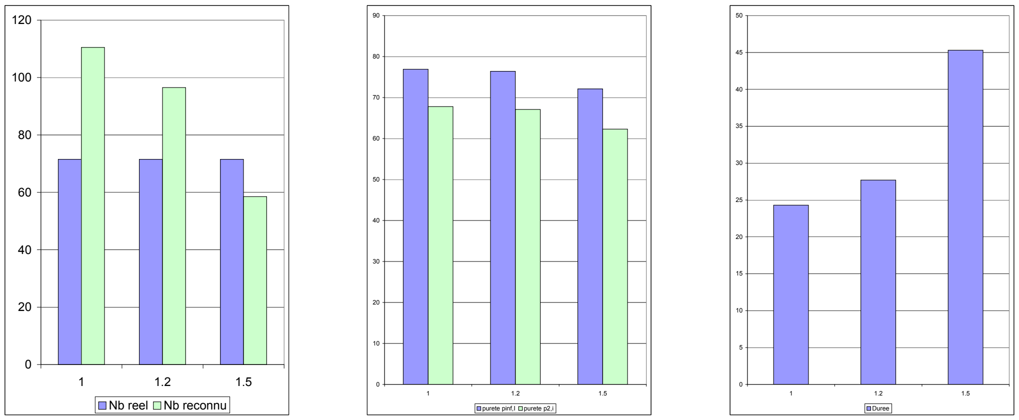

FIG. 9.36 – *Données JT (indexation type I) : segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.5$)*

Si nous comparons maintenant les partitions obtenues pour une valeur de 1.2 pour le poids de pénalité du regroupement et pour les deux segmentations (en clair, si nous nous attachons aux deuxièmes lignes de chaque tableau), la segmentation avec $\lambda = 1.5$ fournit de meilleurs résultats.

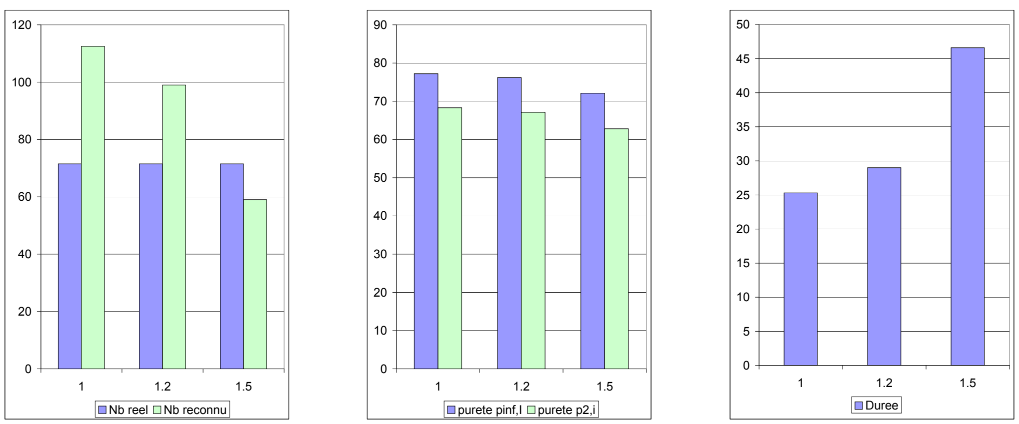

FIG. 9.37 – *Données JT (indexation type I) : segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.2$)*

Nous nous intéressons maintenant aux résultats évalués avec l'*indexation de type II*. Dans le tableau sont inscrits les chiffres du regroupement précédé de SILHYST+DISTBIC. Il y a lieu de tirer les mêmes conclusions qu'avec l'*indexation de type I*. C'est la valeur de 1.2 pour le λ de regroupement qui aboutit à la meilleure partition.

La pureté obtenue pour le *type II* (69.2%) est inférieure à la pureté pour le *type I* (74.0%). Ceci est dû au fait que le *type II* considère comme deux locuteurs différents un même locuteur avec un fond sonore différent. Or, nous faisons un pré-traitement qui consiste à supprimer les effets du canal de transmission (cf section 8) et qui doit réduire par la même occasion, les différences de conditions acoustiques. Aussi, les paroles d'un locuteur parlant dans des environnements sonores différents sont regroupées, d'où une pureté plus faible pour l'*évaluation de type II*.

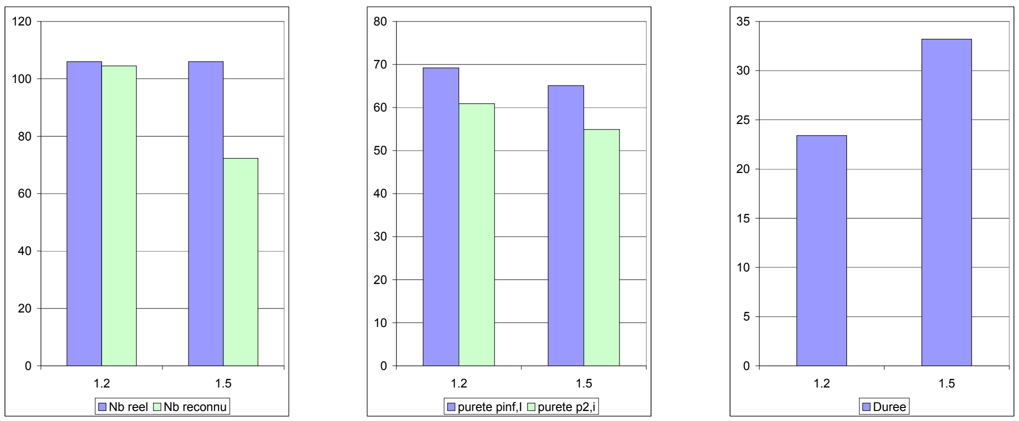

FIG. 9.38 – *Données JT (indexation type II) : segmentation SILHYST+DISTBIC suivie du regroupement (silhyst: durée=0.3s, segmentation : $\lambda = 1.2$)*

Enfin, comme pour l'*évaluation de type I*, les graphes 9.39 et 9.40(tableaux E.39 et E.40) donnent les performances du regroupement précédé de DISTBIC pour les JT, évaluées par l'*indexation de type II*. Comme précédemment, chaque graphe correspond à une valeur du poids de pénalité pour la segmentation et présente les résultats pour différentes valeurs de λ du regroupement. Pour les deux graphes, donc les deux segmentations, c'est $\lambda = 1.0$ (λ du regroupement) qui fournit les meilleurs résultats du point de vue nombre reconnu de locuteurs et pureté.

Si maintenant, nous fixons la valeur du poids de pénalité du regroupement à 1.0 et que nous analysons les résultats pour les deux segmentations (examen des premières lignes des tableaux), ces résultats sont quasiment égaux. Pour une valeur de 1.5 pour le λ de segmentation, le nombre reconnu de locuteurs est de 112.5, il est de 110.5 pour $\lambda = 1.2$. De même, les puretés sont respectivement de 74.1% et de 74.6% et la durée de 24.3s et de 25.3s. Les deux segmentations sont donc équivalentes au vu des résultats du regroupement.

De même que précédemment, nous constatons des baisses de pureté entre les graphes 9.36 et 9.39 (tableaux E.36 et E.39) et entre les graphes 9.37 et 9.40 (tableaux E.37 et E.40) liées au type d'indexation de référence.

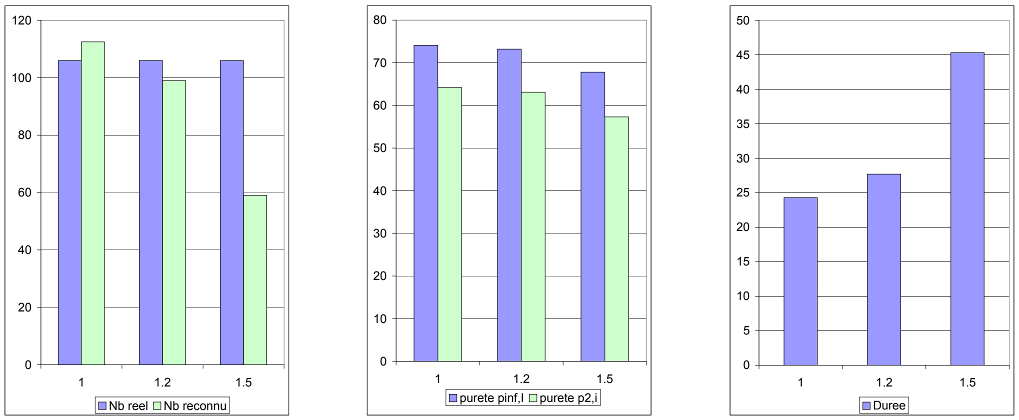

FIG. 9.39 – *Données JT (indexation type II) : segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.5$)*

Il est difficile de conclure dans ce paragraphe quant aux valeurs optimales des paramètres et ce pour trois raisons. La première tient au fait que la longueur des segments en sortie de la segmentation est beaucoup plus courte que la longueur réelle, aussi, les paramètres ne sont plus ajustés. Par ailleurs, toujours en ce qui concerne la longueur des segments, celle-ci varie énormément au sein d'un même journal télévisé. D'où la difficulté de choisir une valeur optimale de paramètre, sachant que celle-ci dépend essentiellement de la longueur des segments. Cette difficulté a d'ailleurs déjà été soulignée pour la segmentation au chapitre 5. Enfin, la troisième et dernière raison résulte de la finesse de l'indexation. En effet, ces journaux télévisés ont été indexés très finement. A savoir que le moindre bruit ou un silence supérieur à 0.3s ou encore les respirations entre les paroles d'un même locuteur sont répertoriés. Une telle finesse est peut-être exigée dans certaines applications mais à ce jour, aucun algorithme n'atteint cette précision sous les hypothèses que nous nous sommes fixées, i.e. pas de connaissance a priori sur la langue ou sur les locuteurs, et au stade du système d'indexation où nous sommes.

Conversations téléphoniques SWB

Nous présentons dans ce paragraphe les résultats obtenus par la segmentation DISTBIC sur les conversations téléphoniques SWB. Nous ne testons pas la détection de silences préalable car, dans une conversation téléphonique il est rare que de longs silences s'interposent entre les paroles des différents locuteurs. Nous expérimentons plusieurs valeurs du poids de pénalité λ pour le regroupement.

Plus la valeur de λ augmente, plus il y a de regroupements. Le nombre reconnu de locuteurs baisse sensiblement puisqu'il passe de 32.5 pour la valeur $\lambda = 1.0$ à 9 pour une valeur de $\lambda = 1.5$. En contrepartie, la pureté diminue. Cependant, la baisse de pureté n'est que de 3% et en parallèle, la durée moyenne triple quasiment, passant de 16.2 s à 44.8 s. La valeur de 1.5 pour λ nous paraît donc particulièrement recommandée pour les données SWB.

Il nous faut préciser que, comme pour les journaux télévisés, nous sommes confrontés à un problème d'indexation de référence. En effet, les transcriptions de ces conversations sont fournies avec les conversations et chaque phrase ou mot est attribué à un des locuteurs. Les

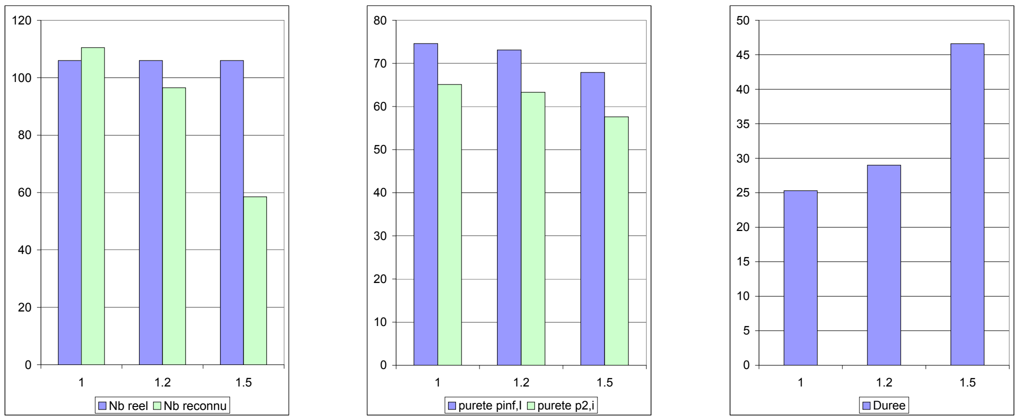

FIG. 9.40 – *Données JT (indexation type II) : segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.2$)*

fichiers d'indexation de référence sont alors générés à partir de ces fichiers de transcription.

Cependant, ces transcriptions comportent des cas particuliers particulièrement difficiles à traiter de manière automatique pour l'indexation. Il s'agit par exemple du recouvrement des paroles des deux personnes. Ce cas ne fait a priori pas partie de nos hypothèses. Néanmoins, ce cas se produit souvent lors de conversations spontanées. Par ailleurs, les silences, aussi longs soient-ils, ne sont pas répertoriés dans les fichiers de transcription. Certains segments ne peuvent donc être étiquetés et sont considérés comme inconnus (mais néanmoins comptabilisés dans notre processus d'évaluation). Pour une évaluation plus juste, il faudrait donc indexer manuellement ces conversations téléphoniques.

Conclusions

Dans cette section, nous avons étudié le regroupement sur des segments issus de segmentations préalables en locuteur. Dans tous les cas, nous pouvons conclure à une baisse des performances par comparaison avec le regroupement de segments de référence. Ceci est prévisible car nous savons que la segmentation ne fournit pas des segments aussi purs que les segments de référence. Sur les données traitées, cette baisse est de l'ordre de 6% à 8%. Dans la majorité des cas, les valeurs des paramètres jugées optimales pour la segmentation et pour le regroupement sont confirmées. Ces valeurs sont directement liées à la longueur des segments de locuteur traités. Pour les conversations réelles (JT et SWB), les puretés obtenues sont plus faibles que celles obtenues pour les conversations synthétiques (baisse de 5% à 15% selon les données). Mais nous avons vu que pour ces conversations, les évaluations peuvent être biaisées par les indexations de référence utilisées.

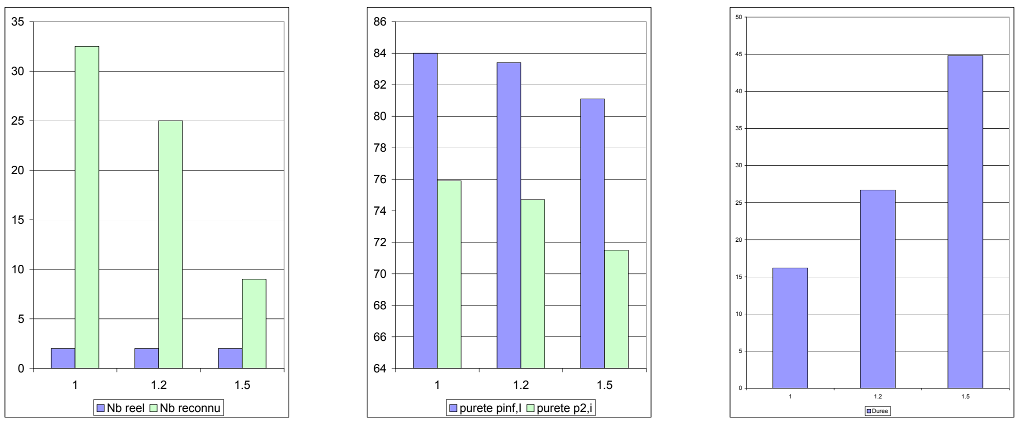

FIG. 9.41 – *Données SWB : segmentation DISTBIC (segmentation : $\lambda = 1.2$)*

Troisième partie

Conclusions et Perspectives

Conclusions et Perspectives

Dans cette thèse, nous avons conçu l'architecture d'un système d'indexation par locuteurs d'un document audio. Il répond aux hypothèses que nous nous sommes fixées, notamment qu'aucune information a priori sur les locuteurs présents dans la conversation n'est disponible et que le nombre de locuteurs est inconnu. Cependant, la connaissance du type de conversations, plus précisément de la rapidité des échanges, est nécessaire car elle permet d'ajuster la valeur des paramètres utilisés dans les techniques mises en place.

Ce système d'indexation est composé de plusieurs étapes :

- la segmentation en locuteurs
- le regroupement des segments par locuteurs
- la modélisation et la reconnaissance de la séquence de locuteurs

Les deux premières étapes visent à déterminer les différentes interventions des différents locuteurs à partir du document audio considéré, avant d'entreprendre la dernière étape de reconnaissance de la séquence de locuteurs.

Concernant la segmentation en locuteurs sans connaissance a priori, nous recommandons tout d'abord l'emploi de l'algorithme de détection de silences SILHYST comme pré-traitement à la segmentation. Cet algorithme permet de supprimer les longs silences qui pourraient venir perturber la segmentation en locuteurs DISTBIC. SILHYST présente l'avantage d'être adaptable à tout type de signal audio, i.e. il n'est pas nécessaire de recalculer les seuils de décision SILENCE/PAROLE.

Notre technique de segmentation par détection de changement de locuteurs DISTBIC fournit quant à elle de bons résultats. Une première passe met en jeu une segmentation par calcul de distances entre deux portions de signal. Nous proposons ensuite une méthode de détection de silences à partir de la courbe des distances qui se généralise à n'importe quel type de signal. Cette première passe fournit en général, un nombre élevé de fausses alarmes car nous privilégions la sur-segmentation. Une deuxième passe dans DISTBIC réduit significativement le nombre de fausses alarmes.

Nous avons vu que cette seconde passe pouvait conduire, dans le cas de conversations contenant de courts segments, à la détérioration du taux de détections manquées. Aussi, nous n'en recommandons pas l'usage pour ce type de conversations.

Par ailleurs, il est préférable dans certaines applications d'aboutir à une sur-segmentation. En effet, l'étape suivante peut soit contribuer à réduire le taux de fausses alarmes, soit s'accomoder de cette sur-segmentation. Le dernier cas de figure peut être illustré par les évaluations NIST (cf Annexe A).

Nous pourrions penser que le regroupement en locuteurs résoud le problème de la sur-segmentation et que, par conséquent, la deuxième passe de notre technique de sgementation s'avère inutile. Il n'en est rien : la longueur des segments issus de la première passe n'est pas suffisante pour effectuer un regroupement correct, comme nous l'avons constaté à la partie II.

Après avoir exposé les différentes méthodes de regroupement existant dans la littérature, nous nous sommes concentrés sur une méthode de regroupement hiérarchique qui utilise le rapport de vraisemblance généralisé comme critère de regroupement et le critère d'information Bayesien comme critère d'arrêt.

Cette méthode donne de très bons résultats sur des segments de référence, i.e. issus d'une segmentation manuelle et qui ne contiennent les paroles que d'un seul locuteur. Des puretés de 80% à 85% sont obtenues sur des groupes de segments courts (i.e. dont la durée est inférieure à 2 secondes) et des puretés de 95% à 100% sont obtenues sur des groupes de segments longs. Une comparaison avec le regroupement de segments issus de nos algorithmes de segmentation montre que la dégradation des puretés est de l'ordre de 6% à 8%.

Etant données les puretés obtenues avec des segments de référence, cet algorithme est particulièrement recommandé pour regrouper les messages laissés par un certain correspondant sur un répondeur téléphonique ou sur une boîte vocale.

L'algorithme a cependant un inconvénient majeur : son temps d'exécution. Nous ne le mentionnons pas car ce n'est pas notre objectif, mais nous sommes très loin du temps réel. Par ailleurs, la version hiérarchique utilise l'ensemble des segments d'une conversation. Quand cette conversation dure longtemps, cela complique singulièrement les calculs et donc augmente par la même occasion le temps de traitement. Une solution consisterait à découper le fichier en petites portions et effectuer le regroupement hiérarchique sur chaque portion de fichier. Une fois les groupes de segments obtenus sur chaque portion, il faudrait effectuer le regroupement de l'ensemble de ces groupes de segments. L'usage du regroupement séquentiel (cf introduction de la partie II et section 7.2.2) pourrait, si nécessaire, améliorer ces deux situations.

A plusieurs reprises, nous n'avons pu conclure sur le choix des paramètres optimaux pour le regroupement hiérarchique, au vu des résultats obtenus. En effet, un bon regroupement hiérarchique doit fournir un nombre de groupes de segments égal au nombre réel de locuteurs et chaque groupe de segments doit avoir une pureté égale à 100%. Compte tenu des deux conditions précédentes, cela signifie également que, la durée moyenne de chaque groupe de segments doit être la plus longue possible (i.e. tous les les segments d'un même locuteur doivent être réunis au sein du même groupe de segments). Cette contrainte sur la durée des segments est aussi imposée pour la modélisation des locuteurs : plus le volume de données d'un locuteur est important, meilleure sera la modélisation.

Toutefois, il est rare d'aboutir à la partition parfaite. Une difficulté apparaît alors : quel critère privilégier ? La pureté, la durée ou le nombre reconnu de locuteurs ? Un compromis entre ces trois critères est nécessaire et devra prendre en compte l'étape suivante de modélisation des locuteurs et de reconnaissance de la séquence de locuteurs.

Perspectives

Nous nous sommes concentrés dans cette thèse sur l'étude de deux étapes de notre sytème d'indexation, à savoir la segmentation et le regroupement par locuteurs. La suite de ce travail va consister à achever ce sytème d'indexation avec la modélisation des locuteurs à partir des groupes de segments obtenus et la reconnaissance de la séquence de locuteurs.

En effet, la disponibilité de données plus abondantes pour un locuteur après la phase de regroupement en permet une modélisation fiable par des méthodes désormais classiques, telles que les GMMs. Cependant, la quantité de données disponible peut être très variable d'un locuteur à l'autre et surtout s'avérer insuffisante pour quelques uns d'entre-eux. Par exemple, au cours d'un journal télévisé, nous aurons très probablement beaucoup de données de parole à disposition pour le journaliste présentateur. A l'inverse, les données de parole d'une personne anonyme interviewée lors d'un reportage risquent d'être peu abondantes compromettant ainsi une modélisation correcte de ce locuteur.

Lors de l'étape de modélisation, nous ne tiendrons pas compte des groupes de segments contenant trop peu de données du locuteur correspondant. Cela implique que lors de l'étape de reconnaissance de la séquence de locuteurs, un segment ne doit pas être sytématiquement étiqueté avec l'identité d'un des locuteurs modélisés. En effet, en ne tenant pas compte des groupes de segments trop petits ou en éliminant les segments trop courts lors du regroupement (cf 8.1.2), il se peut qu'un locuteur n'intervenant dans la conversation que de manière ponctuelle ne soit pas pris en compte. L'importance de cette erreur dépendra de l'application envisagée : nous pouvons attacher de l'importance soit à la détection de tous les locuteurs, soit à celle du locuteur majoritaire.

Pour en revenir à l'étape de reconnaissance de la séquence de locuteurs, nous venons de voir qu'en plus des locuteurs, plus exactement des modèles de locuteurs issus de la segmentation et du regroupement, il faut prévoir un locuteur "poubelle" (*garbage*). Ce modèle permet de classer tous les segments qui n'obtiennent pas un score suffisant avec les autres modèles de locuteurs (modèle est ici un abus de langage car les données qui vont être étiquetées comme telles ne vont pas servir à entraîner le modèle).

Par ailleurs, si une détection de silences a été réalisée au préalable, nous avons vu que les segments de silence obtenus sont de bonne qualité (cf 5.2.2). Ils peuvent alors servir à la construction d'un modèle de silence qui prendra place dans l'étape de reconnaissance de la séquence de locuteurs au même titre que les autres modèles de locuteur.

La reconnaissance de la séquence de locuteurs, à l'aide des modèles construits, peut être vue, dans le cadre de notre sytème d'indexation, sous deux angles differents :

- soit comme un problème de segmentation avec modèles de locuteurs. Dans ce cas, les travaux de [Olsen 95, Sugiyama et al. 93, Wilcox et al. 94] serviront de point d'entrée dans ce domaine.

- soit comme un problème de suivi multi-locuteurs. Le lecteur se reportera alors aux travaux de [Meignier et al. 00, Sonmez et al. 99] par exemple.

Même si les objectifs de ces deux axes de recherche ne sont pas identiques, ils ne sont pas non plus complètement décorrélés et les techniques employées dans les deux cas viennent le confirmer. Les travaux sur la séparation de différents types de signaux, par exemple Parole/Bruit/Silence/Musique peuvent aussi servir de référence pour aborder la reconnaissance de la séquence de locuteurs. En effet, chaque locuteur peut être vu comme un type de signal. [Seck et al. 99, Williams et al. 99] traitent de ce problème. Les travaux réalisés dans le cadre de la campagne 2000 des évaluations DARPA sur les systèmes de poursuite de locuteurs seront à suivre de près (*http://www.itl.nist.gov/iaui/894.01/tests/speaker/2000/index.htm*).

L'étape de reconnaisance de la séquence de locuteurs peut également contribuer au raffinement de la segmentation. Lors de l'étape de segmentation proprement dite et de l'étape de regroupement, la segmentation initiale n'est pas complètement remise en cause. Seule l'erreur de fausse alarme (les paroles d'un même locuteur se répartissent sur plusieurs segments consécutifs) peut être corrigée lors de la seconde passe de la segmentation ou lors du regroupement. Par contre, aucun dispositif n'existe pour éradiquer les erreurs de détection manquées (les paroles de locuteurs différents sont réunies au sein d'un même segment ou d'un même groupe de segments). Ayant désormais des modèles de locuteurs et éventuellement un modèle de silence à notre disposition, nous pouvons envisager de resegmenter à l'étape de reconnaissance de la séquence de locuteurs. Plus exactement, il s'agirait d'un raffinement de la segmentation : un segment qui n'aurait pas obtenu un score suffisant avec l'ensemble des modèles de locuteurs ou de silence pourrait être divisé en sous-segments de manière arbitraire et le processus de reconnaissance recommencerait sur chaque sous-segment. Si un sous-segment n'obtient toujours pas un score convenable avec l'ensemble des modèles alors il serait étiqueté à l'aide du "modèle poubelle".

Nous avons supposé que nous n'avions pas de contraintes de temps réel. Nous pouvons donc envisager d'itérer le processus suivant jusqu'à stabilisation de la reconnaissance de la séquence de locuteurs :

1. étiqueter les segments à l'aide des modèles à disposition et raffiner éventuellement la segmentation

2. réaliser un nouvel apprentissage des modèles avec les données étiquetées

3. revenir à l'étape 1.

Cette méthode, appliquée aux segments, ne remet pas complètement en cause la segmentation résultant de la première étape. Sa généralisation aux trames acoustiques pourrait fournir une segmentation entièrement nouvelle. Il s'agirait alors de segmenter l'enregistrement en utilisant les modèles de locuteurs dans une programmation dynamique semblable à celle utilisée pour la reconnaissance de mots enchaînés. Cette approche devra inclure une contrainte sur la durée pour éviter les éventuels changements de locuteurs de trame en trame. Comme dans les algorithmes d'entraînement des modèles de mots, l'algorithme sera itératif et raffinera successivement les modèles et la segmentation. Les techniques développées dans cette thèse constituent alors l'indispensable étape permettant l'initialisation des modèles de locuteurs et la détermination de leur nombre.

Enfin, nous rappelons qu'à l'issue de ce processus, la séquence de locuteurs présents dans la conversation est connue. Il reste à identifier chacun des locuteurs, mais nous sortons du cadre de notre étude.

Directions de recherche et questions ouvertes

Notre approche a délibérément fait l'hypothèse de l'absence de connaissances a priori sur les locuteurs et sur le langage. Il va de soi que tout ajout de connaissance ne peut qu'améliorer notre système d'indexation. Cet ajout de connaissance peut être, par exemple, de disposer :

- de modèles de genre féminin ou masculin permettant ainsi une segmentation préalable en genre
- de modèles de mots pour faire une segmentation en mots
- de la transcription de la conversation (par exemple, le script d'un film)
- de modèles des locuteurs d'intérêt (présentateur de nouvelles télévisées, hommes politiques célèbres...) pour des applications de poursuite mono- ou multi-locuteurs
- d'un modèle universel de locuteur pour améliorer les critères de regroupement
- etc....

Nous avons vu à plusieurs reprises que les valeurs des paramètres intervenant soit dans la segmentation, soit dans le regroupement hiérarchique, dépendent de la longueur réelle des segments de locuteurs. Quand la longueur des segments n'est pas homogène au sein d'un même document audio, i.e. varie dans de fortes proportions, alors quelle que soit la valeur choisie, elle ne sera pas adaptée à l'ensemble des segments. Une manière de résoudre ce problème consisterait à faire une analyse multi-échelle, i.e. à prendre successivement des valeurs de paramètres différentes et au regard des résultats obtenus pour ces différentes valeurs, valider ou non les changements de locuteurs pour la segmentation ou valider ou non des regroupements pour le regroupement hiérarchique.

Dans cette thèse, nous avons mis en évidence les problèmes posés par la segmentation ou l'indexation de référence. Ces références sont nécessaires à l'évaluation des différentes étapes. Les problèmes sont d'origines diverses. Autant, il est facile d'obtenir la segmentation ou l'indexation de référence pour des conversations synthétiques, c'est-à-dire obtenues en concaténant des phrases de différents locuteurs. Autant, pour des documents audio réels (comme des conversations téléphoniques ou des journaux télévisés), cette tâche se complique singulièrement. La spontanéité de la parole constitue la première source de difficultés. Les paroles des différents locuteurs peuvent se recouvrir et dans ce cas, il est difficile d'indexer. Est-ce une portion de parole appartenant à l'un ou l'autre des locuteurs ou aux deux ? Quand l'un des locuteurs prononce un mot court sur les paroles d'un autre locuteur, est-ce nécessaire de l'indexer (et donc de le détecter) ? Quelle précision utiliser pour l'indexation ? Faut-il répertorier les silences, si courts soient-ils ? Quand les paroles d'un locuteur font l'objet d'une traduction simultanée, les paroles du traducteur viennent s'ajouter avec un volume sonore en général plus important. Comment faut-il alors indexer ? Certaines de ces questions trouvent leur réponse en fonction de l'application envisagée. Alors que d'autres restent ouvertes car elles relèvent également de la subjectivité de la personne qui indexe.

Annexes

Annexe A

Différentes stratégies pour le suivi de locuteur
Various strategies for speaker tracking

Jean-François Bonastre[1] Perrine Delacourt* [2] Corinne Fredouille [1]
Sylvain Meignier [1] Teva Merlin [1] Christian Wellekens [2]

[1] LIA/CERI Université d'Avignon, Agroparc,
BP 1228, 84911 Avignon Cedex 9, France
{jean-francois.bonastre, corinne.fredouille, sylvain.meignier, teva.merlin}@lia.univ-avignon.fr

[2] Institut Eurécom, 2229 route des crêtes,
BP 193, 06904 Sophia Antipolis Cedex, France
{perrine.delacourt, christian.wellekens}@eurecom.fr

Résumé

Ce travail concerne le suivi de locuteur (Speaker Tracking), une tâche proche de l'indexation selon le locuteur. Le travail à réaliser consiste à détecter les segments de document prononcés par un locuteur spécifique. Dans ce cadre, le système possède une référence correspondant au locuteur concerné (cependant, les autres locuteurs présents dans le document ne sont pas connus) et seuls les segments de document prononcés par ledit locuteur sont à prendre en compte.

Dans cet article, nous avons exploré deux stratégies différentes pour élaborer des systèmes de suivi de locuteur. La première s'appuie sur un système d'indexation selon le locuteur. Elle consiste à opérer une détection des changements de locuteurs en amont du système, sans connaissance sur les locuteurs potentiels. Une fois le signal segmenté, un système classique de vérification du locuteur est appliqué à chaque segment obtenu et détermine si ce segment a été prononcé ou non par le locuteur cible. La deuxième solution est élaborée à partir d'un système segmental de reconnaissance du locuteur, dont seule l'étape de prise de décision est adaptée à la tâche visée. Dans ce cas, la décision sur la présence ou non du locuteur cible dans le document est réalisée globalement sur l'ensemble du document. La détection des segments correspondant à ce même locuteur est menée conjointement. Enfin, une amélioration de la dernière technique est discutée, particulièrement dans le cas d'un document contenant de multiples locuteurs.

*Ce travail a bénéficié du soutien du Centre National d'Etudes des Télécommunications (CNET) au titre du contrat n° 98 1B

Mots Clés

suivi de locuteur - reconnaissance du locuteur - indexation de documents

Abstract

This work addresses speaker tracking, which is closely related to speaker indexing. The task consists in detecting the recorded segments uttered by a given speaker. In this approach, only the model of the target speaker is available and only the documents uttered by this given speaker are taken into account.
In this paper, two different stategies are explored to set up systems for speaker tracking. The first one relies on a speaker indexing tool. Speaker turns are detected in the front-end of the system without any knowledge on possible speakers. Once the signal has been segmented, a classical speaker verification process is applied to each segment and checks if this segment corresponds to the target speaker. The second solution is worked out from a segmental speaker recognition system from which only the decision step is adapted to the task at the hand. In this case, decision on the presence of the target speaker in the record is based on the whole recorded document. Segments corresponding to the target speaker are simultaneously detected. Eventually, an improvement of this last technique is discussed, more specifically for documents containing multiple speaker utterances.

Keywords

speaker tracking - speaker recognition - document indexing

1 Introduction

Dans le cadre de l'indexation par le contenu de documents multimédia, la recherche du nombre de locu-

teurs présents dans ledit document ainsi que l'affectation des différents segments de parole au locuteur correspondant constitue une tâche essentielle. Ce processus, appelé "indexation selon le locuteur", montre une grande complexité car, en plus des difficultés classiques rencontrées en traitement automatique de la parole, aucune information sur le document à traiter n'est disponible *a priori*. En particulier, le nombre d'interventions différentes, la durée moyenne de ces interventions, le nombre de locuteurs et les caractéristiques des locuteurs potentiellement présents dans le document ne sont pas connus à l'avance par le système.
Ce travail concerne le suivi de locuteur (Speaker Tracking), une tâche proche de la précédente. Le travail à réaliser consiste à détecter les segments de document prononcés par un locuteur spécifique. Deux différences principales entre l'indexation selon le locuteur et le suivi de locuteur sont à noter :

- Le locuteur est ici connu au préalable ; le système possède une référence correspondant au locuteur concerné. Cependant, les autres locuteurs présents dans le document ne sont pas connus.

- Dans le cadre du suivi de locuteur, on ne s'intéresse qu'aux segments de document prononcés par ledit locuteur. Le nombre de locuteurs et les segments correspondant à des locuteurs différents du locuteur cible ne sont pas pris en considération.

Enfin, la tâche peut être étendue au suivi simultané de plusieurs locuteurs.
Dans cet article, nous avons exploré deux stratégies différentes pour élaborer des systèmes de suivi de locuteur. La première s'appuie sur un système d'indexation selon le locuteur. Elle consiste à opérer une détection des changements de locuteurs en amont du système. Une fois le signal segmenté, un système classique de vérification du locuteur est appliqué indépendamment sur chacun des segments obtenus pour déterminer s'il a été prononcé ou non par le locuteur cible. Cette approche privilégie la qualité de la détection des segments par rapport à la vérification d'identité (réalisée sur des segments potentiellement courts). La deuxième stratégie proposée est élaborée à partir d'un système segmental de reconnaissance du locuteur, dont seule l'étape de prise de décision est adaptée à la tâche visée. Dans ce cas, la décision sur la présence ou non du locuteur cible dans le document est réalisée globalement sur l'ensemble du document. La détection des segments correspondant à ce même locuteur est menée conjointement à la phase précédente. Cette seconde solution privilégie la vérification d'identité (réalisée sur le document entier) par rapport à la fiabilité de la détection des segments.
Enfin, dans la dernière partie de l'article, une amélioration de la seconde approche, modélisant les changements de locuteur par un modèle de Markov caché (HMM), est proposée.

2 AMIRAL : un système général pour la reconnaissance du locuteur

Les deux approches proposées s'appuient sur le système de reconnaissance du locuteur AMIRAL. AMIRAL est un système multi-reconnaisseurs segmental, dédié aux tâches de reconnaissance du locuteur telles que l'Identification Automatique du Locuteur (IAL), la Vérification Automatique du Locuteur (VAL), et plus récemment, au Suivi de Locuteurs (SL). AMIRAL est composé de différents modules détaillés dans les paragraphes suivants.

2.1 Pré-traitement

Pour la phase de pré-traitement du signal de parole, le système AMIRAL utilise le module de paramétrisation standard du consortium ELISA [1]. Le signal de parole est représenté toutes les 10 ms, par 16 coefficients cepstraux, dérivés d'une analyse en bancs de filtres. Une normalisation, basée sur le retrait de la moyenne cepstrale (Cepstral Mean Subtraction) permet de minimiser les perturbations dues aux différents canaux de transmission de la voix.

2.2 Modélisation du locuteur

Le système AMIRAL exploite différentes techniques statistiques de modélisation de la voix. Dans cette étude, les locuteurs sont chacun modélisés par un modèle mono état. Les modèles de locuteurs utilisés sont des mixtures de gaussiennes (Gaussian Mixture Models [6]), entraînées à l'aide de l'algorithme EM (Expectation-Maximization [5]) basé sur le principe du maximum de vraisemblance. Les modèles sont constitués de 16 composantes, caractérisées par des matrices de covariance pleines. Enfin, la mesure de similarité utilisée entre un vecteur de paramètres décrivant une trame de signal et un modèle consiste à calculer la vraisemblance pour que ladite trame ait été émise par le modèle considéré.

2.3 Approche bloc-segmentale et normalisation

Un des aspects spécifiques du système AMIRAL est de considérer le signal de parole à un niveau segmental. Comme les segments considérés sont de taille fixe et de très courte durée (0,3 seconde), cette approche est nommée "bloc-segmentale". Durant la phase de test,

1. Le consortium ELISA est composé de laboratoires de recherche européens travaillant sur une plate-forme de référence pour l'évaluation des systèmes de reconnaissance du locuteur. Ces laboratoires sont: ENST (France), EPFL (Suisse), IDIAP (Suisse), IRISA (France), LIA (France), VUTBR (République Tchèque), RMA (Belgique), RIMO (Rice (Etats Unis) et Mons (Belgique)).

le signal de parole est découpé en blocs temporels, sur chacun desquels une mesure de similarité normalisée est calculée. Cette architecture permet de renforcer la robustesse du système en supprimant les zones non informatives lors de la décision. Elle a permis également une adaptation aisée d'AMIRAL aux tâches d'indexation ou de suivi de locuteur.

La normalisation appliquée au niveau de chaque bloc a pour but de pallier les problèmes de variabilité classiques en reconnaissance du locuteur et surtout d'homogénéiser les mesures de similarité.

La méthode utilisée combine deux techniques. Une première étape consiste à calculer un rapport de vraisemblance, pour chaque bloc, en divisant la vraisemblance obtenue par rapport au modèle du locuteur considéré (le signal a été prononcé par ledit locuteur) par la vraisemblance de l'hypothèse inverse (le signal provient d'un autre locuteur), modélisée par un modèle général de locuteur (souvent nommé "modèle du monde"). Une seconde normalisation de type MAP (Maximum A Posteriori) est, ensuite, appliquée sur chaque rapport de vraisemblance. Cette normalisation a pour objectif de prendre en compte le comportement du reconnaisseur, en remplaçant le rapport de vraisemblance par la probabilité *a posteriori* que le locuteur cible ait prononcé le segment de signal correspondant. Cette normalisation nécessite d'apprendre le comportement du système sur un ensemble séparé de données de développement et de choisir différentes probabilités *a priori* décrivant les conditions de test. L'avantage majeur de cette normalisation consiste à proposer des scores bornés ayant un sens dans le domaine probabiliste [7][8].

2.4 Architecture multi-reconnaisseurs

La deuxième particularité du système AMIRAL est de reposer sur une architecture multi-reconnaisseurs. Deux grands types de reconnaisseurs sont à distinguer selon la nature des informations prises en compte. Le premier type s'intéresse principalement au domaine spectral qui est divisé en sous-bandes fréquentielles traitées indépendamment. Le deuxième type exploite les informations dynamiques du signal de parole en concaténant des trames successives sur une fenêtre temporelle de taille constante et en sélectionnant un sous-ensemble de paramètres jugés optimaux pour la caractérisation du locuteur. Les mesures de similarité normalisées, issues de chaque reconnaisseur sont alors fusionnées au niveau de chaque bloc temporel. L'étape de fusion met en œuvre les avantages du processus de normalisation explicité dans la section 2.3.

2.5 Stratégies de décision

Le dernier module du système AMIRAL est constitué d'une étape de fusion et d'une étape de décision. La première phase consiste à fusionner les différentes mesures de similarités obtenues à raison d'une par bloc temporel. Plusieurs stratégies sont employées, de la simple moyenne arithmétique (notée AM) à des méthodes spécifiques aux tâches de suivi de locuteur en passant par des stratégies "d'élagage" ("pruning"), éliminant les blocs sur lesquels un niveau de bruit important est détecté.

3 Différentes stratégies pour le suivi de locuteur

Le but du suivi de locuteur est de rechercher dans un enregistrement audio les paroles prononcées par le locuteur cible, en d'autres termes de détecter le début et la fin des segments de parole attribuables à celui-ci. Dans cette section, plusieurs approches sont proposées.

La première approche consiste à réaliser une phase de détection des changements de locuteurs proposant une segmentation du message en segments homogènes (prononcés par un seul locuteur). Un processus de vérification du locuteur est alors appliqué indépendamment sur chacun des segments obtenus. L'étape de pré-segmentation n'utilise aucune connaissance *a priori* des locuteurs engagés dans la conversation. Elle est décrite au paragraphe 3.1.

La deuxième solution proposée dans cet article exploite directement les possibilités segmentales du système AMIRAL. Une phase de décision spécifique au suivi de locuteur suit l'étape de calcul des mesures de similarité entre les différents blocs temporels du message et le modèle du locuteur cible. Les processus de segmentation et de vérification du locuteur cible sont ici unifiés au sein du procédé de reconnaissance du locuteur. Le système dérivé d'AMIRAL est détaillé au paragraphe 3.2.

Enfin, un raffinement de la deuxième technique est présenté au paragraphe 3.3. Cette solution est particulièrement dédiée aux documents comprenant un nombre important de locuteurs (conférences, films, etc.) et pour lesquels un suivi simultané de plusieurs locuteurs est nécessaire.

3.1 Segmentation préliminaire en locuteurs

L'utilisation d'une segmentation préliminaire en locuteurs avant le processus de vérification repose sur l'idée suivante : le score de vérification est plus fiable si les segments considérés ne comportent que des trames (vecteurs acoustiques) provenant d'un unique locuteur. De même, la longueur des segments influe très fortement sur la qualité des résultats (cf [4]). Ainsi, le but de la segmentation en locuteurs est de découper le signal de parole en plusieurs segments homogènes : chaque segment ne doit contenir des paroles prononcées par un seul locuteur. Cette segmentation consiste à détecter les changements de locuteurs et est réalisée sans tenir compte de la connaissance du locuteur cible (le modèle de ce locuteur n'est pas exploité).

Détection d'un changement de locuteur. Etant données deux portions de signal paramétrisées (deux séquences de vecteurs acoustiques) $\mathcal{X}_1 = \{x_1, ..., x_i\}$ et $\mathcal{X}_2 = \{x_{i+1}, ..., x_{n_{\mathcal{X}}}\}$, nous considérons le test d'hypothèses suivant pour un changement de locuteur à l'instant i:

- H_0 : les deux portions sont relatives au même locuteur. Leur réunion est modélisée par un unique processus gaussien : $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \sim \mathcal{N}(\mu_{\mathcal{X}}, \Sigma_{\mathcal{X}})$
- H_1 : chaque portion a été prononcée par un locuteur différent et est modélisée par un processus gaussien différent : $\mathcal{X}_1 \sim \mathcal{N}(\mu_{\mathcal{X}_1}, \Sigma_{\mathcal{X}_1})$ et $\mathcal{X}_2 \sim \mathcal{N}(\mu_{\mathcal{X}_2}, \Sigma_{\mathcal{X}_2})$

Le rapport de vraisemblance généralisé, R, entre les hypothèses H_0 et H_1 est défini par :

$$R = \frac{L(\mathcal{X}, \mathcal{N}(\mu_{\mathcal{X}}, \Sigma_{\mathcal{X}})}{L(\mathcal{X}_1, \mathcal{N}(\mu_{\mathcal{X}_1}, \Sigma_{\mathcal{X}_1}).L(\mathcal{X}_2, \mathcal{N}(\mu_{\mathcal{X}_2}, \Sigma_{\mathcal{X}_2})}$$

Ce rapport de vraisemblance généralisé a été utilisé dans [2][3] en identification du locuteur et a prouvé son efficacité. La distance d_{R} est obtenue en prenant le logarithme de l'expression précédente : $d_{\mathrm{R}} = -\log R$ ("distance" est ici un abus de langage car d_{R} ne vérifie pas les propriétés d'une distance).
Une valeur élevée de R (i.e. une faible valeur de d_{R}) signifie que la modélisation avec une seule gaussienne (hypothèse H_0) s'accorde mieux aux données. A l'opposé, une faible valeur de R (i.e. une forte valeur de d_{R}) indique que l'hypothèse H_1, i.e. la modélisation par deux gaussiennes, correspond mieux aux données. Dans ce cas, un changement de locuteur est détecté à l'instant i.

Détection de tous les changements de locuteurs. La distance d_{R} est calculée pour chaque couple de fenêtres de signal de même durée (environ 2 secondes). Ces fenêtres doivent être suffisamment longues pour estimer de manière fiable les paramètres des gaussiennes et suffisamment courtes pour faire l'hypothèse qu'elles ne contiennent les paroles que d'un seul locuteur. Ces fenêtres sont glissantes et sont déplacées à chaque itération d'un laps de temps fixe (environ 0,1 seconde) le long du signal paramétrisé, comme le montre la figure 1.
Les distances calculées pour chaque couple de fenêtres sont stockées pour former à la fin du processus une courbe de distances. Les pics les plus significatifs (en terme d'amplitude) de cette courbe sont alors détectés : ces pics correspondent aux changements de locuteur recherchés. Un maximum local de la courbe des distances est considéré comme significatif si les différences entre son amplitude et celle des minima situés de part et d'autre sont supérieures à un certain seuil (dépendant de la variance de la distribution des

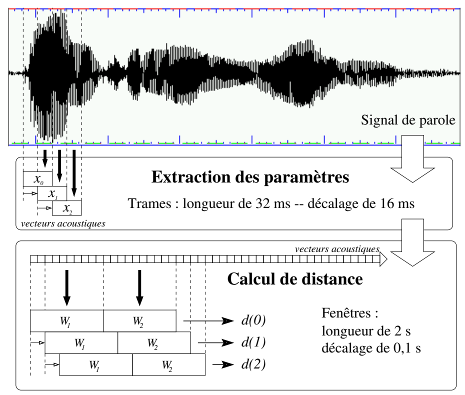

FIG. 1 – *Calcul de distance par fenêtres glissantes*

distances). Nous imposons également un intervalle de temps minimal entre deux changements de locuteurs consécutifs (cf figure 2). La détection des changements de locuteurs ne se fait donc pas en considérant l'amplitude absolue des pics mais plutôt en considérant leur facteur de forme, comme détaillé dans [1].

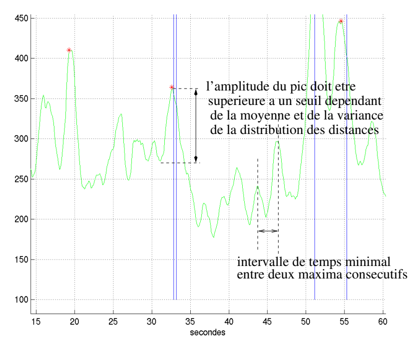

FIG. 2 – *Détection des points de changement de locuteur à l'aide du graphe de distances*

Une détection manquée (i.e. un changement de locuteur qui n'est pas détecté) est plus préjudiciable pour le processus de vérification qu'une fausse alarme (un changement est détecté alors qu'il n'existe pas). En effet, un segment contenant les paroles de plusieurs locuteurs (résultant d'une détection manquée) ne pourra être correctement identifié. Aussi, les paramètres impliqués dans la détection des changements de locuteurs ont été ajustés de manière à éviter les détections manquées au détriment du nombre de fausses alarmes. Le signal est probablement sur-segmenté : les paroles consécutives d'un même locuteur sont réparties sur plusieurs segments. Cependant, la durée des segments de locuteurs obtenus est suffisamment grande pour avoir une décision de vérification fiable.

Pour cette méthode, nous avons considéré que les segments temporels étaient mono-locuteur (et mono condition d'enregistrement) et nous avons configuré le système AMIRAL en conséquence : une simple moyenne arithmétique a été choisie comme procédé de fusion temporelle (pour obtenir la mesure de similarité finale, à partir des mesures provenant de chaque bloc temporel). Un procédé de décision classique, basé sur un seuil de décision, permet d'attribuer un segment (ou non) au locuteur cible.

3.2 Suivi de locuteur basé sur un système de reconnaissance du locuteur

Cette stratégie repose à la fois sur l'approche segmentale temporelle du système AMIRAL (section 2.3) et sur un algorithme de décision spécifique qui permet simultanément de choisir les zones de signal correspondant le mieux au locuteur cible et de décider si les informations retenues permettent d'identifier la présence dudit locuteur, sur l'ensemble du document. La technique proposée se décompose donc en deux phases :

1. Le système de reconnaissance du locuteur AMIRAL est mis en œuvre pour obtenir une mesure de similarité entre chaque bloc du signal et le modèle du locuteur cible;

2. Une stratégie de détection simultanée de la zone de décision appropriée et de prise de décision est alors mise en œuvre, à partir des données précédentes. Cette méthode est nommée SWGM (Sorted Weighted Geometric Mean). Elle consiste en quatre phases :

   – Une phase préliminaire, de tri des blocs dans l'ordre décroissant des mesures de similarité associées (i.e. : le plus probable en premier)

   – Une seconde phase permet la détection du sous-ensemble de blocs temporels optimal pour prendre la décision globale de présence du locuteur cible dans le document. Plus précisément, cette phase consiste à rechercher le sous-ensemble de blocs Eb tel que le score Sc attribué à l'ensemble du document soit maximal :

   $$Sc = f(Mg(Eb), Card(Eb))$$

   Avec : $Mg(Eb)$ correspondant à la moyenne géométrique des mesures de similarité associées aux blocs composant l'ensemble Eb. $Card(Eb)$ correspondant au cardinal de l'ensemble Eb. Et $f(x, y)$ une fonction pondérant la moyenne x en fonction du nombre y d'éléments à partir desquels x a été calculée. $f()$ correspond donc à une estimation de la confiance attribuée à la moyenne. Dans le cadre de cet article, $f(x, y) = x\sqrt[y]{0,1}$.

   – Une phase de décision est alors mise en œuvre. Cette phase compare le score Sc obtenu à l'étape précédente à un seuil ($Sdec$) prédéterminé. Si le score est inférieur au seuil, le document entier est rejeté (le locuteur cible n'est pas du tout présent dans ce document).

   – Enfin, dans le cas contraire, une étape d'extension du sous ensemble Eb est réalisée. Elle consiste à ajouter un à un les différents blocs dans Eb, toujours selon l'ordre décroissant des mesures de similarité associées aux blocs. Ce processus est arrêté dès que le score attribué à l'ensemble des blocs sélectionnés (par $f()$) est inférieur au seuil de décision $Sdec$.

   Finalement, ce nouvel ensemble de segments temporels est attribué dans sa totalité au locuteur cible.

NB:
Dans le cadre des évaluations NIST99 ([9]), une étape complémentaire de segmentation, facultative, a été utilisée pour prédécouper l'enregistrement en plusieurs zones. Cette étape, basée sur un seuil de rejet fixe permettant de détecter les zones très peu probables, était nécessaire vue la durée importante (plusieurs minutes) des enregistrements considérés. L'algorithme décrit dans le paragraphe précédent a ensuite été appliqué sur chacune des zones retenues.

L'avantage majeur de cette stratégie est de réaliser presque conjointement la décision dite "de vérification d'identité" et la segmentation. Cette étape de décision est plus robuste car réalisée sur l'ensemble des données présentes dans le document.

3.3 Détection simultanée de plusieurs locuteurs

La méthode présentée dans la section 3.2 offre l'avantage d'utiliser la connaissance *a priori* du locuteur cible durant les étapes de segmentation et de décision (réalisées simultanément). L'amélioration proposée ici concerne les documents contenant de multiples locuteurs. Dans ce cas, la tâche de suivi de locuteur est souvent réalisée pour différents locuteurs cibles.

Le principe novateur proposé ici consiste d'une part à réaliser simultanément l'ensemble des détections correspondant aux locuteurs cibles recherchés, en exploitant l'ensemble des modèles disponibles, et d'autre part, à modéliser les changements de locuteurs à l'aide d'un HMM.

Cette méthode réutilise la première étape de l'approche décrite dans la section 3.2. Les scores normalisés sont également calculés pour chaque bloc temporel mais maintenant pour l'ensemble des locuteurs connus du système (locuteurs cibles). De même, le système calcule des scores normalisés à partir d'un modèle générique de non parole (bruit, silence...) et un modèle générique de parole (appris sur un ensemble de données séparé). Un modèle HMM est alors construit, en associant un état à chaque locuteur cible. Deux états, correspondant respectivement au modèle de parole et au modèle de non parole sont ajoutés au modèle HMM. Un algorithme de type Viterbi attribue alors de manière optimale chaque bloc temporel à un des modèles. L'ensemble des règles utilisées pour définir la valeur des probabilités de transition du modèle HMM est exprimé sous forme d'une matrice contenant les poids de passage d'un état à un autre. Les poids choisis sont déterminés par l'opérateur en fonction des objectifs de la tâche d'indexation. En particulier, l'opérateur choisit le coût d'une erreur d'indexation d'un bloc (bloc attribué à un mauvais locuteur) par rapport au coût d'une non détection.
Les probabilités de transition vérifient trois conditions :

- La probabilité de transition entre les états doit être plus faible que la probabilité de rester dans le même état, car les interventions sont majoritairement composées de plusieurs blocs consécutifs.
- Les probabilités de rester dans le même état sont égales.
- Les locuteurs étant équiprobables, les probabilités de passer d'un état à un autre (différent) sont alors identiques.

Exemple : Transformation d'une matrice de poids en matrice de transition.
Soit la matrice de poids exprimée par le tableau 1.

Modèles	P.	Non P.	Loc. I	Loc. J ($\neq$ du Loc. I)
P.	5	1	5	5
Non P.	5	1	5	5
Loc. I	5	1	12	1

TAB. 1 – **Matrice des poids** *du modèle X (en ligne) vers le modèle Y (en colonne). P.: modèle de parole, Non P.: modèle de non parole, Loc. I, Loc. J: modèles des locuteurs I et J*

Pour un modèle de Markov à cinq états (Parole, Non Parole, Locuteur 1, Locuteur 2, Locuteur 3), le système construit la matrice de transition donnée par le tableau 2, où chaque poids est :

- reporté dans la matrice de transition;
- divisé par la somme marginale de la ligne, de sorte que la somme des probabilités des arcs sortant d'un état soit égale à 1 (propriété des modèles de Markov).

Modèles	P.	Non P.	Loc. 1	Loc. 2	Loc. 3
P.	5/21	1/21	5/21	5/21	5/21
Non P.	5/21	1/21	5/21	5/21	5/21
Loc. 1	5/20	1/20	12/20	1/20	1/20
Loc. 2	5/20	1/20	1/20	12/20	1/20
Loc. 3	5/20	1/20	1/20	1/20	12/20

TAB. 2 – **Matrice de transition** du modèle X (en ligne) vers le modèle Y (en colonne). P.: modèle de parole, Non P.: modèle de non parole, Loc. i: modèle du locuteur i

4 Expériences

4.1 Bases de données et protocoles d'évaluation

Les deux stratégies proposées ont été testées durant la campagne NIST/NSA99 d'évaluation des systèmes de reconnaissance du locuteur, qui proposait pour la première année des tests de suivi de locuteur.
Ces deux stratégies ont été élaborées de manière à respecter les conditions de cette évaluation : pour chaque test, il existe un seul locuteur cible et seule la connaissance sur ce locuteur est utilisée.
Dans ce contexte, les corpus utilisés sont composés d'enregistrements de conversations téléphoniques spontanées, issus d'un sous-ensemble du corpus Switchboard II, fourni dans le cadre de la campagne d'évaluation NIST/NSA99. Ce sous-corpus est composé de 230 hommes et 309 femmes. Les données d'entraînement pour chaque locuteur sont constituées de deux minutes de parole enregistrées sur deux sessions différentes.
L'apprentissage des modèles du monde, de bruit, et de parole, comme celui de la fonction de normalisation, sont réalisés en utilisant un corpus complètement séparé (cf section 2.3).
Les deux approches ont été évaluées durant la campagne NIST99, à l'aide de 4000 fichiers d'essai d'une minute de parole chacun.
La dernière méthode a été testée sur un corpus artificiel de 5000 messages, constitué en mélangeant des enregistrements mono-locuteur issus de la même base que précédemment (NIST/Switchboard).

4.2 Résultats et commentaires

La figure 3 montre les résultats obtenus par chacun des participants à la campagne NIST/NSA99, et en particulier par les deux systèmes présentés ici. Ces résultats sont fournis sous forme d'une courbe montrant les fausses acceptations (blocs attribués à tort au locuteur cible) en fonction des faux rejets (blocs prononcés par le locuteur cible et rejetés à tort par le système). Les résultats sont calculés à raison d'une décision tous les centièmes de seconde.

Ces résultats montrent peu d'écart entre les différents compétiteurs de la campagne NIST/NSA99. La différence entre les sytèmes est masquée d'une part par la difficulté intrinsèque de la tâche (les écarts entre les différents compétiteurs lors de la campagne NIST99 étaient très faibles, pour le suivi de locuteur) ainsi que par le mode de calcul des performances, qui pénalise de la même manière toute erreur. En particulier, une erreur de positionnement d'une frontière de segment, de 1/100 de seconde, est comptabilisée au même niveau qu'une fausse détection de segment.

Au regard de ces différents points, la comparaison des résultats des deux systèmes ne permet pas de mettre en avant l'une ou l'autre des stratégies proposées.

Les résultats correspondant à la variante dédiée à la détection simultanée de plusieurs locuteurs sont nettement plus encourageants. Ce système a été capable d'attribuer 77 % des blocs (de 0,3 s) au locuteur cible correspondant, avec simultanément 8 % d'erreur d'affectation de blocs (blocs attribués à un mauvais locuteur), sur un total de 810047 blocs à affecter. Ces résultats doivent être nuancés par les conditions de l'expérience : tous les modèles de locuteurs étaient connus du système et le corpus de test, bien qu'issu du même ensemble de données que pour les expériences précedentes, était construit artificiellement.

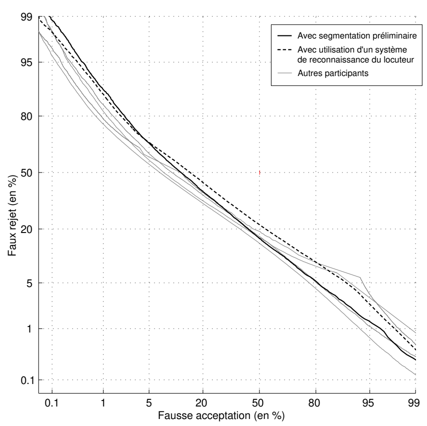

FIG. 3 – *Résultats obtenus lors de la campagne d'évaluation NIST99 – Taux de faux rejet en fonction du taux de fausse acceptation (taux calculés trame par trame)*

5 Conclusion

Nous avons présenté dans cet article deux approches très différentes dans le cadre d'un système de suivi de locuteur. Si les conditions expérimentales n'ont pas permis de conclure définitivement sur les avantages respectifs de ces méthodes, il apparait clairement que développer une technique de suivi de locuteur à partir d'un système de reconnaissance du locuteur, sans segmentation au préalable, est une voie d'investigation intéressante.

Enfin, l'association d'un tel système et d'un HMM a montré un très bon niveau de performances. Il reste cependant à tester cette option dans le cas où tous les locuteurs ne sont pas connus du système.

Références

[1] P. Delacourt, D. Kryze, C.J. Wellekens, Speaker-based segmentation for audio data indexing, *ESCA workshop: accessing information in audio data*, 1999.

[2] H. Gish, M.-H. Siu, R. Rohlicek, Segregation of speakers for speech recognition and speaker identification, *ICASSP*, pp 873-876, 1991.

[3] H. Gish, N. Schmidt, Text-independent speaker identification, *IEEE Signal Processing magazine*, pp 18-32, Oct. 1991.

[4] I. Magrin-Chagnolleau, A.E. Rosenberg, S. Parthasarathy, Detection of target speakers in audio databases, *ICASSP*, 1999.

[5] D. Dempster, N. Larid, D. Rubin, Maximum likelihood from incomplete data via EM algorithm, *J. Roy. Stat. Soc.*, Vol. 39, pp 1-38, 1977.

[6] D. A. Reynolds, Speaker identification and verification using gaussian mixture speaker models, *Speech Communication*, pp 91-108, Aug. 1995.

[7] C. Fredouille, J.-F. Bonastre, T. Merlin,

Segmental normalization for robust speaker verification, *Workshop on robust methods for speech recognition in adverse conditions*, 1999.

[8] C. Fredouille, J.-F. Bonastre, T. Merlin, Similarity normalization method based on world model and a posteriori probability for speaker verification, *EUROSPEECH*, 1999.

[9] M.A. Przybocki, A.F. Martin, NIST Speaker Recognition Evaluation 1997, *RLA2C*, pp 120-123, Apr. 1998.

Annexe B

Démonstration de la formule de la distance de KullbachLeibler pour des distributions Gaussiennes

Soit une séquence $X = \{x_1, ..., x_n\}$ de n vecteurs de dimension d et $Y = \{y_1, ..., y_m\}$ une autre séquence de m vecteurs, aussi de dimension d. La distance de Kullbach-Leibler entre ces deux séquences est définie comme suit :

$$\begin{aligned} KL(X,Y) &= \int_{-\infty}^{+\infty} p_X(X) \log \frac{P_X(X)}{P_Y(X)} dX \\ &= E_X \left(\log P_X(X) - \log P_Y(X) \right) \end{aligned}$$

où $E_X()$ désigne l'espérance mathématique calculée avec la distribution de probabilité P de X. Pour symétriser cette mesure, la mesure $KL2$ est définie par :

$$KL2(X,Y) = KL(X,Y) + KL(Y,X) \tag{B.2}$$

Dans ce qui suit, nous prenons les conventions de notations suivantes :
$\int$ signifie $\int_{-\infty}^{+\infty}$ et $\sum_i$ signifie $\sum_{i=1}^{d}$ où d est la dimension de l'espace dans lequel nous travaillons.

Si nous supposons que les séquences X et Y sont modélisées par des processus Gaussiens multi-dimensionnels P_X et P_Y respectivement, nous avons :

$$P_X(x) = \frac{\exp(-\frac{1}{2}(x-\mu_X)^T \Sigma_X^{-1} (x-\mu_X))}{(2\pi)^{\frac{d}{2}} |\Sigma_X|^{\frac{1}{2}}}$$

L'expression de la distance de Kullbach-Leibler devient alors :

$$\begin{aligned} KL(X,Y) &= \int \ldots \int P_X(x) \left(\frac{1}{2} \log \frac{|\Sigma_X|}{|\Sigma_Y|} - \frac{1}{2}(x-\mu_X)^T \Sigma_X^{-1} (x-\mu_X) + \frac{1}{2}(x-\mu_Y)^T \Sigma_Y^{-1} (x-\mu_Y) \right) dx \\ &= \frac{1}{2} \log \left(\frac{|\Sigma_X|}{|\Sigma_Y|} \right) \int \ldots \int P_X(x) dx - \frac{1}{2} \int \ldots \int P_X(x) \left((x-\mu_X)^T \Sigma_X^{-1} (x-\mu_X) \right) dx \\ &\quad + \frac{1}{2} \int \ldots \int P_X(x) \left((x-\mu_Y)^T \Sigma_Y^{-1} (x-\mu_Y) \right) dx \end{aligned}$$

$$\begin{aligned} &= \frac{1}{2}\log\frac{|\Sigma_X|}{|\Sigma_Y|} - \frac{1}{2}\underbrace{\int\ldots\int P_X(x)\left((x-\mu_X)^T\Sigma_X^{-1}(x-\mu_X)\right)dx}_{\text{A'}} \\ &+\frac{1}{2}\underbrace{\int\ldots\int P_X(x)\left((x-\mu_Y)^T\Sigma_Y^{-1}(x-\mu_Y)\right)dx}_{\text{A}} \end{aligned}$$

car $\int\ldots\int P_X(x)dx = 1$ par définition.
Nous nous intéressons maintenant plus particulièrement au calcul de A :

$$\begin{aligned} A &= \int\ldots\int P_X(x)\left((x-\mu_X)^T\Sigma_X^{-1}(x-\mu_X)\right)dx \\ &= \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_X|^{\frac{1}{2}}}\int\ldots\int(x-\mu_X)^T\Sigma_X^{-1}(x-\mu_X)exp\left(-\frac{1}{2}(x-\mu_X)^T\Sigma_X^{-1}(x-\mu_X)\right)dx \end{aligned}$$

Les matrices de covariance sont des matrices symétriques réelles. Elles sont donc définies positives et sont diagonalisables à l'aide de matrices unitaires. Nous pouvons donc écrire :

$$\Sigma_X = T_X^{-1}\Delta_X T_X$$

où Δ_X est la matrice diagonale des valeurs propres de Σ_X et T_X la matrice de passage orthogonale qui vérifie les propriétés suivantes :

$$T_X^{-1} = T_X^T \Rightarrow T_X^T T = I_d \qquad \text{et} \qquad |T_X| = 1$$

où I_d est la matrice identité. Nous pouvons donc en déduire que :

$$\Sigma_X = T_X^T\Delta_X T_X \qquad \text{et} \qquad |\Sigma_X^{\frac{1}{2}}| = |\Delta_X^{\frac{1}{2}}|$$

Soit les changements de variables suivants :

$$y = \Delta_X^{\frac{-1}{2}}T_X(x-\mu_X) \qquad \text{alors} \qquad dy = \frac{1}{|\Sigma_X^{\frac{1}{2}}|}dx = \frac{1}{|\Delta_X^{\frac{1}{2}}|}dx$$

Soient $V_X = \Delta_X^{\frac{-1}{2}}T_X\mu_X$ et $V_Y = \Delta_X^{\frac{-1}{2}}T_X\mu_Y$, nous définissons :

$$\eta = V_Y - V_X = \Delta_X^{\frac{-1}{2}}T_X(\mu_Y-\mu_X)$$

En intégrant ces changements de variable dans l'expression de A, nous obtenons :

$$A = \frac{1}{(2\pi)^{\frac{d}{2}}}\int\ldots\int\underbrace{(y-\eta)^T\Delta_X^{\frac{1}{2}}T_X\Sigma_Y^{-1}T_X^T\Delta_X^{\frac{1}{2}}(y-\eta)}_{\text{forme quadratique Q}}exp\left(-\frac{1}{2}y^Ty\right)dy \tag{B.3}$$

Si $a[i,j]$ désigne l'élément de la i-ème ligne et de la j-ième colonne du produit de matrices $\Delta_X^{\frac{1}{2}}T_X\Sigma_Y^{-1}T_X^T\Delta_X^{\frac{1}{2}}$ et si $y[i]$ et $\eta[i]$ désignent le i-ème élément du vecteur y et η respectivement, alors la forme quadratique Q peut s'écrire sous la forme :

$$\begin{aligned} Q &= \sum_i\sum_j a[i,j](y[i]-\eta[j])(y[j]-\eta[i]) \\ &= \sum_i a[i,i](y[i]-\eta[i])^2 + \sum_i\sum_{j\neq i}a[i,j](y[i]-\eta[i])(y[j]-\eta[j]) \end{aligned}$$

En remplaçant dans A, nous avons :

$$\begin{aligned} A &= \underbrace{\frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} \int \ldots \int (y[i]-\eta[i])^2 \exp\left(-\frac{1}{2}\sum_k y[k]^2\right) dy}_{\text{B}} \\ &\quad \underbrace{+\frac{\sum_i \sum_{j\neq i} a[i,j]}{(2\pi)^{\frac{d}{2}}} \int \ldots \int (y[i]-\eta[i])(y[j]-\eta[j]) \exp\left(-\frac{1}{2}\sum_k y[k]^2\right) dy}_{\text{C}} \end{aligned} \tag{B.4}$$

Nous rappelons les résultats suivants sur les intégrales d'Euler :

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{y^2}{2}\right) dy = \sqrt{2\pi}$$

$$\int_{-\infty}^{+\infty} y^2 \exp\left(-\frac{y^2}{2}\right) dy = \sqrt{2\pi}$$

En tenant compte de ces résultats, B peut se simplifier :

$$\begin{aligned} B &= \frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} \int \ldots \int (y[i]-\eta[i])^2 \exp\left(-\frac{1}{2}\sum_k y[k]^2\right) dy \\ &= \frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} \int \ldots \int (y[i]^2 - 2y[i]\eta[i] + \eta[i]^2) \prod_k \exp\left(-\frac{y[k]^2}{2}\right) dy \\ &= \frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} \left(\prod_{k\neq i} \int \exp\left(-\frac{y[k]^2}{2}\right) dy[k]\right) \left(\int (y[i]^2 - 2y[i]\eta[i] + \eta[i]^2) \exp\left(-\frac{y[i]^2}{2}\right) dy[i]\right) \\ &= \frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} \left(\prod_{k\neq i} \sqrt{2\pi}\right) \\ &\quad \left(\int y[i]^2 \exp\left(-\frac{y[i]^2}{2}\right) dy[i] - 2\eta[i] \underbrace{\int y[i] \exp\left(-\frac{y[i]^2}{2}\right) dy[i]}_{=0 \text{ fonction impaire}} + \eta[i]^2 \int \exp\left(-\frac{y[i]^2}{2}\right) dy[i]\right) \\ &= \frac{\sum_i a[i,i]}{(2\pi)^{\frac{d}{2}}} (2\pi)^{\frac{d-1}{2}} (\sqrt{2\pi} + \eta[i]^2 \sqrt{2\pi}) \\ &= \sum_i a[i,i](1+\eta[i]^2) \end{aligned}$$

De même, C se simplifie :

$$\begin{aligned} C &= \frac{\sum_i \sum_{j\neq i} a[i,j]}{(2\pi)^{\frac{d}{2}}} \int \ldots \int (y[i]-\eta[j])(y[j]-\eta[i]) \exp\left(-\frac{1}{2}\sum_k y[k]^2\right) dy \\ &= \frac{\sum_i \sum_{j\neq i} a[i,j]}{(2\pi)^{\frac{d}{2}}} \left(\prod_{k\neq i, k\neq j} \int \exp\left(\frac{-y[k]^2}{2}\right) dy[k]\right) \end{aligned}$$

$$\left(\underbrace{\int y[i]\exp\left(\frac{-y[i]^2}{2}\right)dy[i]}_{=0 \text{ fontion impaire}} - \int \eta[i]\exp\left(\frac{-y[i]^2}{2}\right)dy[i]\right)$$
$$\left(\underbrace{\int y[j]\exp\left(\frac{-y[j]^2}{2}\right)dy[j]}_{=0 \text{ fontion impaire}} - \int \eta[j]\exp\left(\frac{-y[j]^2}{2}\right)dy[j]\right)$$
$$\begin{aligned} &= \frac{\sum_i \sum_{j\neq i} a[i,j]}{(2\pi)^{\frac{d}{2}}}(2\pi)^{\frac{d-1}{2}}(-\eta[i]\sqrt{2\pi})(-\eta[j]\sqrt{2\pi}) \\ &= \sum_i \sum_{j\neq i} a[i,j]\eta[i]\eta[j] \end{aligned}$$

Alors, nous déduisons de la formule B.4 l'expression suivante de A :

$$\begin{aligned} A &= B + C \\ &= \sum_i a[i,i](1+\eta[i]^2) + \sum_i \sum_{j\neq i} a[i,j]\eta[i]\eta[j] \\ &= \sum_i a[i,i] + \sum_i \sum_j a[i,j]\eta[i]\eta[j] \end{aligned}$$

En reprenant les changements de variables, nous pouvons exprimer A comme suit :

$$\begin{aligned} A &= \mathrm{tr}(\Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}}) + \eta^T \Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}})\eta \\ &= tr(\Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}}) + (\mu_Y - \mu_X)^T T_X^T \Delta_X^{-\frac{1}{2}} \Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}} \Delta_X^{-\frac{1}{2}} T_X(\mu_Y - \mu_X) \\ &= \underbrace{tr(\Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}})}_{\text{D}} + (\mu_Y - \mu_X)^T \Sigma_Y^{-1}(\mu_Y - \mu_X) \end{aligned}$$

En utilisant la propriété $tr(AB) = tr(BA)$ et l'unitarité de T_X:

$$\begin{aligned} D &= tr(\Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}}) \\ &= tr(T_X T_X^T \Delta_X^{\frac{1}{2}} T_X \Sigma_Y^{-1} T_X^T \Delta_X^{\frac{1}{2}} T_X T_X^T) \\ &= tr(T_X \Sigma_X^{\frac{1}{2}} \Sigma_Y^{-1} \Sigma_X^{\frac{1}{2}} T_X^T) \\ &= tr(\underbrace{T_X T_X^T}_{=I_d} \Sigma_X^{\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}} \Sigma_X^{\frac{1}{2}}) \\ &= tr((\Sigma_X^{\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}})^T) \end{aligned}$$

En remplaçant D dans A puis A dans la formule de Kullbach-Leibler (cf B.3) et en constatant que A' s'exprime comme A mais en remplaçant tous les indices Y par X, nous obtenons :

$$KL(X,Y) = \frac{1}{2}\log\frac{|\Sigma_X|}{|\Sigma_Y|} + \frac{1}{2}tr((\Sigma_X^{\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}} \Sigma_Y^{-\frac{1}{2}})^T) + \frac{1}{2}(\mu_Y - \mu_X)^T \Sigma_Y^{-1}(\mu_Y - \mu_X)$$

$$
\begin{aligned}
& -\frac{1}{2}tr((\underbrace{\Sigma_X^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}}}_{=I_d})^T) - \frac{1}{2}(\underbrace{\mu_X - \mu_X}_{=0})^T\Sigma_X^{-1}(\mu_X - \mu_X) \\
= & \frac{1}{2}\log\frac{|\Sigma_X|}{|\Sigma_Y|} + \frac{1}{2}tr((\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})^T) + \frac{1}{2}(\mu_Y - \mu_X)^T\Sigma_Y^{-1}(\mu_Y - \mu_X) - \frac{d}{2}
\end{aligned}
$$

En prenant la version symétrisée de la distance de Kullbach-Leibler (cf B.2), nous retrouvons la formule de l'équation (4.6) :

$$
\begin{aligned}
KL2(X,Y) = & \frac{1}{2}\log\frac{|\Sigma_X|}{|\Sigma_Y|} + \frac{1}{2}tr((\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})^T) + \frac{1}{2}(\mu_Y - \mu_X)^T\Sigma_Y^{-1}(\mu_Y - \mu_X) - \frac{d}{2} \\
& + \frac{1}{2}\log\frac{|\Sigma_Y|}{|\Sigma_X|} + \frac{1}{2}tr((\Sigma_Y^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}})(\Sigma_Y^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}})^T) + \frac{1}{2}(\mu_X - \mu_Y)^T\Sigma_X^{-1}(\mu_X - \mu_Y) - \frac{d}{2} \\
= & \frac{1}{2}tr((\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})(\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-\frac{1}{2}})^T + (\Sigma_Y^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}})(\Sigma_Y^{\frac{1}{2}}\Sigma_X^{-\frac{1}{2}})^T) \\
& + \frac{1}{2}(\mu_Y - \mu_X)^T(\Sigma_Y^{-1} + \Sigma_X^{-1})(\mu_Y - \mu_X) - d
\end{aligned}
$$

Ce qui donne en dimension 1 :

$$
KL2(X,Y) = \frac{1}{2}\left(\frac{\sigma_X^2}{\sigma_Y^2} + \frac{\sigma_Y^2}{\sigma_X^2}\right) + \frac{1}{2}(\mu_X - \mu_Y)^2\left(\frac{1}{\sigma_X^2} + \frac{1}{\sigma_Y^2}\right) - 1
$$

Nous retrouvons ainsi la formule (3.3)
D peut aussi s'exprimer comme suit :

$$
\begin{aligned}
D &= tr(\Delta_X^{\frac{1}{2}}T_X\Sigma_Y^{-1}T_X^T\Delta_X^{\frac{1}{2}}) \\
&= tr(T_XT_X^T\Delta_X^{\frac{1}{2}}T_X\Sigma_Y^{-1}T_X^T\Delta_X^{\frac{1}{2}}T_XT_X^T) \\
&= tr(T_X\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-1}\Sigma_X^{\frac{1}{2}}T_X^T) \\
&= tr(\underbrace{T_XT_X^T}_{=I_d}\Sigma_X^{\frac{1}{2}}\Sigma_X^{\frac{1}{2}}\Sigma_Y^{-1}) \\
&= tr(\Sigma_X\Sigma_Y^{-1})
\end{aligned}
$$

En remplaçant D dans A puis A dans la formule B.3 et en prenant la version symétrisée, nous obtenons une autre expression de la distance de Kullbach-eibler :

$$
KL2(X,Y) = \frac{1}{2}tr(\Sigma_X\Sigma_Y^{-1} + \Sigma_Y\Sigma_X^{-1}) + \frac{1}{2}(\mu_Y - \mu_X)^T(\Sigma_Y^{-1} + \Sigma_X^{-1})(\mu_Y - \mu_X) - d
$$

qui n'est autre que la formule de la divergence dans le cas Gaussien (cf équation 7.15).

Annexe C

Expression du rapport de vraisemblance avec un modèle de données Gaussien

Soit une séquence $X = \{x_1, ..., x_N\}$ de N vecteurs de dimension d modélisée par une Gaussienne multi-dimensionnelle $N(\mu, \Sigma)$:

$$P(x, N(\mu, \Sigma)) = \frac{\exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}$$

La vraisemblance de cette séquence X est donnée par :

$$\mathcal{L}(X, \mu, \Sigma) = \prod_{i=1}^{n} \frac{\exp(-\frac{1}{2}(x_i-\mu)^T \Sigma^{-1}(x_i-\mu))}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}$$

En prenant le logarithme de cette expression, nous obtenons :

$$\log \mathcal{L}(X, \mu, \Sigma) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T \Sigma^{-1}(x_i-\mu)$$

Soit le test d'hypothèses suivant :

- H_0 : La séquence de vecteurs est modélisée par la Gaussienne multi-dimensionnelle

$$X = \{x_1, ..., x_N\} \sim N(\mu, \Sigma)$$

- H_1 : La séquence est générée par deux processus Gaussiens multi-dimensionnels différents :

$$X_1 = \{x_1, ..., x_{N_1}\} \sim N(\mu_1, \Sigma_1) \text{ et } X_2 = \{x_{N_1+1}, ..., x_N\} \sim N(\mu_2, \Sigma_2)$$

avec $N_2 = N - N_1$.

Le rapport de vraisemblance associé à ce test d'hypothèse est de la forme :

$$R = \frac{\mathcal{L}(X, \mu, \Sigma)}{\mathcal{L}(X_1, \mu_1, \Sigma_1)\mathcal{L}(X_2, \mu_2, \Sigma_2)}$$

Le logarithme de ce rapport de vraisemblance est donné par :

$$\begin{aligned}
\log R &= \log \mathcal{L}(X,\mu,\Sigma) - \log \mathcal{L}(X_1,\mu_1,\Sigma_1) - \log \mathcal{L}(X_2,\mu_2,\Sigma_2) \\
&= -\frac{n}{2}\log|\Sigma| + \frac{n_1}{2}\log|\Sigma_1| + \frac{n_2}{2}\log|\Sigma_2| \\
&\quad -\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) + \frac{1}{2}\sum_{i=1}^{n_1}(x_i-\mu_1)^T\Sigma_1^{-1}(x_i-\mu_1) + \frac{1}{2}\sum_{i=1}^{n_2}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\mu_2)
\end{aligned} \tag{C.1}$$

Si $\sigma[k,l]$ désigne l'élément de la k-ième ligne et de la l-ième colonne de Σ^{-1} et $x_i[j]$ et $\mu[j]$ les j-ièmes éléments des vecteurs x_i et μ respectivement. Nous pouvons alors écrire :

$$\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) = \sum_{i=1}^{n}\left(\sum_{l=1}^{d}\sum_{k=1}^{d}(x_i[l]-\mu[l])\sigma[k,l](x_i[k]-\mu[k])\right) \tag{C.2}$$

Soit S la matrice de covariance estimée à partir des échantillons et $s[k,l]$ ses éléments. Cette matrice est par définition de la forme :

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T$$

et

$$s[k,l] = \frac{1}{N}\sum_{i=1}^{N}(x_i[k]-\mu[k])(x_i[l]-\mu[l])$$

Alors l'équation C.2 peut se réécrire :

$$\begin{aligned}
\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) &= \sum_{l=1}^{d}\sum_{k=1}^{d}\sigma[k,l]\left(\sum_{i=1}^{n}(x_i[l]-\mu[l])(x_i[k]-\mu[k])\right) \\
&= N\sum_{l=1}^{d}\sum_{k=1}^{d}\sigma[k,l]s[k,l] \\
&= Ntr(\Sigma^{-1}S)
\end{aligned}$$

Si nous faisons de plus l'hypothèse que $S=\Sigma$, i.e. que l'estimée de la matrice de covariance sur les échantillons est égale à la matrice de covariance théorique (donc que l'estimation est parfaite) alors $\Sigma^{-1}S = I_d$ où I_d est la matrice identité. Dans ce cas, nous en déduisons que :

$$\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) = Nd$$

Avec ce dernier résultat, nous pouvons simplifier l'équation (C.1) :

$$\begin{aligned}
\log R &= -\frac{n}{2}\log|\Sigma| + \frac{n_1}{2}\log|\Sigma_1| + \frac{n_2}{2}\log|\Sigma_2| \\
&\quad -\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) + \frac{1}{2}\sum_{i=1}^{n_1}(x_i-\mu_1)^T\Sigma_1^{-1}(x_i-\mu_1) + \frac{1}{2}\sum_{i=1}^{n_2}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\mu_2) \\
&= -\frac{n}{2}\log|\Sigma| + \frac{n_1}{2}\log|\Sigma_1| + \frac{n_2}{2}\log|\Sigma_2| - \frac{N}{2}d + \frac{N_1}{2}d + \frac{N_2}{2}d \\
&= -\frac{n}{2}\log|\Sigma| + \frac{n_1}{2}\log|\Sigma_1| + \frac{n_2}{2}\log|\Sigma_2|
\end{aligned} \tag{C.3}$$

Annexe D

Interprétation de la pureté $p_{2,i}$

A la section 9.1, nous présentons différentes puretés, notamment la pureté $p_{2,i}$ définie par (en conservant les mêmes notations que précédemment) :

$$p_{2,i} = 100 \times \sum_j \frac{n_{ij}^2}{n_i^2} \%$$

Afin de mieux comprendre la signification de cette pureté, nous aimerions savoir pour quelle composition du groupe de segments i, cette pureté prend sa valeur optimale.

Pour le groupe i, soit n le nombre total de segments, n_1 le nombre de segments du locuteur majoritaire, n_j le nombre de segments relatifs au locuteur j et N_L le nombre de locuteurs présents dans le groupe. La formulation de la question précédente revient à un problème d'optimisation sous contrainte avec la contrainte suivante :

$$n - \sum_{j=1}^{N_L} n_j = 0 \tag{D.1}$$

La théorie de l'optimisation sous contraintes nous amène à optimiser l'expression suivante :

$$J = \left(\frac{n_1}{n}\right)^2 + \sum_{j=2}^{N_L} \left(\frac{n_j}{n}\right)^2 + \lambda(n - \sum_{j=1}^{N_L} n_j) \tag{D.2}$$

où λ est un multiplicateur de Lagrange. Pour ce faire, nous dérivons cette expression par rapport au paramètre n_j et nous l'annulons :

$$\begin{aligned} \frac{\partial J}{\partial n_j} &= 2\frac{n_j}{n} - \lambda \\ &= 0 \end{aligned} \tag{D.3}$$

De l'équations (D.3), nous en déduisons que :

$$\frac{n_j}{n} = \frac{\lambda}{2} \tag{D.4}$$

En remplaçant cette dernière expression dans la contrainte D.1, nous trouvons la valeur suivante pour λ :

$$\lambda = \frac{n - n_1}{n(N_L - n_1)} \tag{D.5}$$

Finalement, nous obtenons la valeur optimale de J en remplaçant λ dans (D.2) par la valeur trouvée à l'équation (D.5) :

$$\begin{aligned} J_{opt} &= \left(\frac{n_1}{n}\right)^2 + (N_L - 1)\left(\frac{n - n_1}{n(N_L - 1)}\right)^2 \\ &= \left(\frac{n_1}{n}\right)^2 + \frac{1}{N_L - 1}\left(\frac{n - n_1}{n}\right)^2 \end{aligned} \tag{D.6}$$

Cela signifie que la pureté $p_{2,i}$ prend une valeur optimale lorsque les locuteurs non majoritaires présents dans le groupe de segments i ont la même fréquence. Il nous reste à démontrer que cette valeur est minimale.

Soit le cas particulier suivant d'un groupe i composé comme suit : le locuteur 1 est majoritaire avec n_1 segments, le locuteur 2 possède n_2 segments, et pour les autres $N_L - 2$ locuteurs $n_i = 0$ pour $i > 2$. Nous avons alors $n_2 < n_1$ et $n_2 = n - n_1$.

Par définition, J est alors égal à :

$$\begin{aligned} J &= \left(\frac{n_1}{n}\right)^2 + \left(\frac{n_2}{n}\right)^2 \\ &= \left(\frac{n_1}{n}\right)^2 + \left(\frac{n - n_1}{n}\right)^2 \end{aligned}$$

et :

$$J > J_{opt}$$

La valeur optimale de J est donc une valeur minimale.

En conclusion, la pureté $p_{2,i}$ est minimale quand les locuteurs non majoritaires présents dans le groupe de segments i ont la même fréquence.

Annexe E

Tableaux de résultats du regroupement hiérarchique

Les tableaux de résultats suivants sont formés comme suit :

- la première colonne mentionne le **paramètre** que nous faisons varier
- la deuxième colonne donne le **nombre réel** *Nb réel* **de locuteurs** présents dans le document audio. Ce nombre est une moyenne sur l'ensemble des documents audio évalués.
- la troisième colonne donne le **nombre de locuteurs effectivement trouvés** *Nb reconnu* suite au regroupement hiérarchique. Ce nombre est une moyenne sur l'ensemble des documents audio évalués.
- la quatrième colonne fournit **la pureté** $p_{\infty,i}$ définie à l'équation (9.4) exprimée en pourcentage.
- la cinquième colonne fournit **la pureté** $p_{2,i}$ définie à l'équation (9.5) exprimée en pourcentage.
- la sixième colonne donne la **durée moyenne en secondes** des groupes de segments.

A part la valeur du paramètre testé, **les nombres apparaissant dans les autres colonnes des tableaux de résultats sont des moyennes établies sur l'ensemble des documents audio considérés durant le test**.

E.1 Evaluation avec des segments de référence

E.1.1 Influence de la dimension de l'espace acoustique

Dimension	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
12	17.6	14.7	66.8 %	57.3 %	15.8
16	17.6	14.2	72.9 %	64.5 %	15.2
24	17.6	8.6	65.6 %	55.3 %	25.2
12 + 12 Δ	17.6	3.8	41.1 %	31.9 %	60.5
16 + 16 Δ	17.6	3.2	36.4 %	28.3 %	73.6

TAB. E.1 – *Données CNET : influence de la dimension de l'espace acoustique et de l'apport des coefficients Δ ($\lambda = 1.0$)*

Dimension	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
12	23.6	31.5	86.2 %	82.7 %	10.0
16	23.6	25.3	85.9 %	82.2 %	12.3
24	23.6	15.1	74.9 %	70.3 %	21.3
12 + 12 Δ	23.6	12.1	63.8 %	58.2 %	26.7
16 + 16 Δ	23.6	6.6	49.0 %	53.6 %	51.4

TAB. E.2 – *Données TIMIT : influence de la dimension de l'espace acoustique et de l'apport des coefficients Δ ($\lambda = 1.0$)*

Dimension	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
12	2	7.8	100 %	100 %	49.9
16	2	4.3	100 %	100 %	84.3
24	2	2.8	99.9 %	99.8 %	127.3
12 + 12 Δ	2	2.8	99.8 %	99.7 %	129.5
16 + 16 Δ	2	2.5	99.5 %	99.7 %	138.3

TAB. E.3 – *Données DIAL : influence de la dimension de l'espace acoustique et de l'apport des coefficients Δ ($\lambda = 1.2$)*

Dimension	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
12	10	29.3	94.6 %	92.1 %	29.0
16	10	21.0	94.6 %	93.1 %	36.0

TAB. E.4 – *Données CONV : influence de la dimension de l'espace acoustique ($\lambda = 1.2$)*

E.1.2 Influence de la pénalité λ intervenant dans le critère d'arrêt

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
0.8	17.6	21.7	79.0 %	72.1 %	10.1
1.0	17.6	14.2	72.9 %	64.5 %	15.2
1.2	17.6	9.1	67.6 %	57.8 %	24.0

TAB. E.5 – *Données CNET : influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
0.8	23.6	36.5	90.5 %	88.4 %	8.6
1.0	23.6	25.3	85.9 %	82.2 %	12.3
1.2	23.6	17.6	77.4 %	72.3 %	17.9

TAB. E.6 – *Données TIMIT : influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
1.0	2	6	100 %	100 %	59.3
1.2	2	4.3	100 %	100 %	84.3
1.5	2	3.8	100 %	100 %	106.6

TAB. E.7 – *Données DIAL : influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
1.0	10	31.3	95.6 %	93.6 %	26.2
1.2	10	21.0	94.6 %	93.1 %	36.0
1.5	10	14.3	92.4 %	89.9 %	56.6

TAB. E.8 – *Données CONV: influence de la pénalité λ intervenant dans le critère d'arrêt (vecteurs acoustiques de dimension 16)*

E.1.3 Influence du pré-traitement et du post-traitement pour les segments courts

Traitements	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
avec pré/avec post	17.6	21.7	79.0 %	72.1 %	10.1
avec pré/sans post	17.6	21.7	93.2 %	91.6 %	6.8
sans pré/sans post	17.6	47.2	86.4 %	84.2 %	4.8

Tab. E.9 – *Données CNET : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt $\lambda = 0.8$)*

Traitements	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
avec pré/avec post	23.6	36.5	90.5 %	88.4 %	8.6
avec pré/sans post	23.6	36.5	90.9 %	89.2 %	8.4
sans pré/sans post	23.6	38.6	91.6 %	90.0 %	8.2

Tab. E.10 – *Données TIMIT : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt $\lambda = 0.8$)*

Traitements	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
avec pré/avec post	2	3.8	100 %	100 %	106.6
avec pré/sans post	2	3.8	100 %	100 %	100.4
sans pré/sans post	2	4.5	99.8 %	99.6 %	81.9

Tab. E.11 – *Données DIAL : influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

Traitements	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
avec pré/avec post	10	14.3	92.4 %	89.9 %	56.6
avec pré/sans post	10	14.3	94.6 %	93.4 %	34.3
sans pré/sans post	10	28.0	94.3 %	93.0 %	34.2

TAB. E.12 – *Données CONV: influence du pré-traitement et du post-traitement pour les segments courts (vecteurs de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

E.1.4 Influence de la mesure de distance inter-groupes de segments

Inter	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
min	17.6	28.5	78.5 %	72.2 %	14.1
max	17.6	25.1	80.0 %	75.1 %	8.7
moyenne	17.6	25.1	78.3 %	71.8 %	9.2
mise à jour	17.6	21.7	79.0 %	72.1 %	10.1

TAB. E.13 – *Données CNET : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt* $\lambda = 0.8$*)*

Inter	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
min	23.6	65.5	98.0 %	97.6 %	4.8
max	23.6	56.7	96.2 %	95.6 %	5.6
moyenne	23.6	57.8	97.1 %	96.8 %	5.5
mise à jour	23.6	36.5	90.5 %	88.4 %	8.6

TAB. E.14 – *Données TIMIT : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt* $\lambda = 0.8$*)*

Inter	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
min	2	3	99.4 %	98.9 %	122.1
max	2	4.5	99.9 %	99.8 %	112.8
moyenne	2	11	99.9 %	99.8 %	106.0
mise à jour	2	3.75	100 %	100 %	100.4

TAB. E.15 – *Données DIAL : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

Inter	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
min	10	98.8	97.5 %	96.4 %	9.2
max	10	28.3	92.2 %	90.0 %	31.6
moyenne	10	46.3	95.7 %	94.1 %	20.4
mise à jour	10	14.3	92.4 %	89.9 %	56.6

TAB. E.16 – *Données CONV : influence de la mesure de distance inter-groupes de segments (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt $\lambda = 1.5$)*

Nom	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
dial1	2	6	100 %	100 %	63.4
dial1_sans_sil	2	2	100 %	100 %	157.8
dial2	2	5	100 %	100 %	74.4
dial2_sans_sil	2	2	100 %	100 %	130.9

TAB. E.17 – *Données DIAL : influence de la présence de silences (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

E.1.5 Influence de la présence de silences

Nom	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
conv1	10	18	89.5 %	86.8 %	53.7
conv1_sans_sil	10	10	86.7 %	83.2 %	63.1
conv2	10	18	95.7 %	93.5 %	55.5
conv2_sans_sil	10	11	97.7 %	95.8 %	54.1

TAB. E.18 – *Données CONV : influence de la présence de silences (vecteurs acoustiques de dimension 16, pénalité du critère d'arrêt* $\lambda = 1.5$*)*

E.2 Evaluation avec des segments résultant de la segmentation

Les tableaux de résultats sont formés comme précédemment (cf description page 157). Dans les légendes des tableaux, nous mentionnons les valeurs des paramètres qui interviennent dans les différents traitements et qui sont susceptibles d'être modifiés. Ces paramètres sont les suivants :

- pour SILHYST, la durée minimale des silences (cf section 4.1)
- pour DISTBIC, la valeur du paramètre λ intervenant dans le critère BIC à la seconde passe (cf section 4.2)
- pour le regroupement hiérarchique, la valeur du paramètre λ intervenant dans le critère d'arrêt BIC

E.2.1 Comparaison des différentes méthodes de segmentation suivies du regroupement

Segmentation	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
REFERENCE	17.6	21.7	79.0 %	72.1 %	10.1
SILHYST+DISTBIC	17.6	34.7	77.6 %	71.0 %	6.6
DISTBIC	17.6	34.7	78.2 %	72.1 %	6.6

TAB. E.19 – *Données CNET : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.15 s, segmentation : $\lambda = 1.0$, regroupement : $\lambda = 0.8$)*

Segmentation	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
REFERENCE	23.6	36.5	88.4 %	90.5 %	8.6
SILHYST+DISTBIC	23.6	36.0	81.7 %	86.0 %	8.7
DISTBIC	23.6	38.1	81.7 %	76.7 %	8.3

TAB. E.20 – *Données TIMIT : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.2$, regroupement : $\lambda = 0.8$)*

Segmentation	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
REFERENCE	2	6	100 %	100 %	63.4
SILHYST+DISTBIC	2	3.5	85.9 %	79.9 %	69.0
DISTBIC	2	5.5	92.8 %	86.9 %	72.3

TAB. E.21 – *Données DIAL : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.5$, regroupement : $\lambda = 1.5$)*

Segmentation	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
REFERENCE	10	18	92.6 %	90.2 %	54.6
SILHYST+DISTBIC	10	21	79.5 %	70.1 %	33.3
DISTBIC	10	20.5	86.0 %	77.7 %	47.1

TAB. E.22 – *Données CONV : comparaison des différentes méthodes de segmentation suivies du regroupement (silhyst : durée=0.3 s, segmentation :* $\lambda = 1.5$*, regroupement :* $\lambda = 1.5$*)*

E.2.2 Influence du poids de pénalité de la segmentation

λ (segmentation)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	23.6	38.1	80.9 %	75.9 %	8.2
$\lambda = 1.2$	23.6	36.0	86.0 %	81.7 %	8.7

TAB. E.23 – *Données TIMIT : influence du poids de pénalité de la segmentation SILHYST + DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement :* $\lambda = 0.8$*)*

λ (segmentation)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	2	4.0	87.4 %	81.0 %	71.4
$\lambda = 1.5$	2	3.5	85.9 %	79.9 %	69.0

TAB. E.24 – *Données DIAL : influence du poids de pénalité de la segmentation SILHYST + DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement :* $\lambda = 1.5$*)*

λ (segmentation)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	2	6.0	90.2 %	85.0 %	66.1
$\lambda = 1.5$	2	5.5	92.8 %	86.9 %	72.3

TAB. E.25 – *Données DIAL : influence du poids de pénalité de la segmentation DISTBIC suivie du regroupement (regroupement :* $\lambda = 1.5$*)*

λ (segmentation)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	23.6	19	80.6 %	71.1 %	42.2
$\lambda = 1.5$	23.6	21	79.5 %	70.1 %	33.3

TAB. E.26 – *Données CONV : influence du poids de pénalité de la segmentation SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3s, regroupement : $\lambda = 1.5$)*

E.2.3 Influence du poids de pénalité du regroupementt

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 0.8$	17.6	34.7	77.6 %	71.0 %	6.6
$\lambda = 1.0$	17.6	22.0	69.7 %	61.5 %	10.1

TAB. E.27 – *Données CNET : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.15 s, segmentation :* $\lambda = 1.0$*)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 0.8$	17.6	34.7	78.2 %	72.1 %	6.6
$\lambda = 1.0$	17.6	22.0	69.9 %	62.0 %	10.1

TAB. E.28 – *Données CNET : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.0$*))*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 0.8$	23.6	36.0	86.0 %	81.7 %	8.7
$\lambda = 1.0$	23.6	23.3	77.8 %	71.3 %	13.3

TAB. E.29 – *Données TIMIT : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation :* $\lambda = 1.2$*)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 0.8$	23.6	38.1	81.7 %	76.7 %	8.3
$\lambda = 1.0$	23.6	24.7	74.8 %	67.3 %	12.9

Tab. E.30 – *Données TIMIT : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.2$)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	2	7.0	85.8 %	79.9 %	53.0
$\lambda = 1.5$	2	3.5	85.9 %	79.9 %	69.0

Tab. E.31 – *Données DIAL : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.5$)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	2	7	92.9 %	87.2 %	56.6
$\lambda = 1.5$	2	5.5	92.8 %	86.9 %	72.3

Tab. E.32 – *Données DIAL : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.5$)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	10	25	79.6 %	70.3 %	27.9
$\lambda = 1.5$	10	21	79.5 %	70.1 %	33.3

Tab. E.33 – *Données CONV : influence du poids de pénalité du regroupement pour la segmentation combinée SILHYST+DISTBIC suivie du regroupement (silhyst : durée=0.3 s, segmentation : $\lambda = 1.5$)*

λ (regroupement)	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	10	28.5	85.2 %	77.6 %	34.0
$\lambda = 1.5$	10	20.5	86.0 %	77.7 %	47.1

Tab. E.34 – *Données CONV : influence du poids de pénalité du regroupement pour la segmentation DISTBIC suivie du regroupement (segmentation : $\lambda = 1.5$)*

E.2.4 Données JT

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	71.5	104.5	74.0 %	66.3 %	23.4
$\lambda = 1.5$	71.5	72.3	70.7 %	61.3 %	33.2

TAB. E.35 – *Données JT (indexation type I) : segmentation SILHYST+DISTBIC suivie du regroupement (silhyst: durée=0.3s, segmentation :* $\lambda = 1.2$*)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	71.5	110.5	76.9 %	67.8 %	24.3
$\lambda = 1.2$	71.5	96.5	76.4 %	67.1 %	27.7
$\lambda = 1.5$	71.5	58.5	72.1 %	62.3%	45.3

TAB. E.36 – *Données JT (indexation type I) : segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.5$*)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	71.5	112.5	77.2 %	68.3 %	25.3
$\lambda = 1.2$	71.5	99.0	76.2 %	67.1 %	29.0
$\lambda = 1.5$	71.5	59.0	72.1 %	62.8%	46.6

TAB. E.37 – *Données JT (indexation type I) : segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.2$*)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.2$	106	104.5	69.2 %	60.9 %	23.4
$\lambda = 1.5$	106	72.3	65.1 %	54.9 %	33.2

TAB. E.38 – *Données JT (indexation type II) : segmentation SILHYST+DISTBIC suivie du regroupement (silhyst: durée=0.3s, segmentation :* $\lambda = 1.2$*)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	106	112.5	74.1 %	64.2 %	24.3
$\lambda = 1.2$	106	99	73.2 %	63.1 %	27.7
$\lambda = 1.5$	106	59	67.8 %	57.3 %	45.3

TAB. E.39 – *Données JT (indexation type II) : segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.5$*)*

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	106	110.5	74.6 %	65.1 %	25.3
$\lambda = 1.2$	106	96.5	73.1 %	63.3 %	29.0
$\lambda = 1.5$	106	58.5	67.9 %	57.6 %	46.6

TAB. E.40 – *Données JT (indexation type II) : segmentation DISTBIC suivie du regroupement (segmentation :* $\lambda = 1.2$*)*

E.2.5 Données SWB

λ	Nb réel	Nb reconnu	pureté $p_{\infty,i}$	pureté $p_{2,i}$	Durée
$\lambda = 1.0$	2	32.5	84.0 %	75.9 %	16.2
$\lambda = 1.2$	2	25	83.4 %	74.7 %	26.7
$\lambda = 1.5$	2	9	81.1 %	71.5 %	44.8

TAB. E.41 – *Données SWB : segmentation DISTBIC (segmentation : $\lambda = 1.2$)*

Bibliographie

[André-Obrecht 88] André Obrecht (R.). – A new statistical approach for the automatic segmentation of continuous speech signals. *IEEE Trans. ASSP*, vol. 36, n° 1, Jan 1988, pp. 29–40.

[Bakis et al. 97] Bakis (R.), Chen (S.), Gopalakrishnan (P.), Gopinath (R.), Maes (S.) et Polymenakos (L.). – Transcription of broadcast news - system robustness issues and adaptation techniques. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 711–714. – 1997.

[Barker et al. 98] Barker (J.), Williams (G.) et Renals (S.). – Acoustic confidence measures for segmenting broadcast news. *International Conference on Spoken Language Processing*, pp. 2719–2722. – 1998.

[Beigi et al. 98] Beigi (H.S.M.) et Maes (S.). – Speaker, channel and environment change detection. *World Congress of Automation.* – 1998.

[Bimbot et al. 95] Bimbot (F.), Magrin-Chagnolleau (I.) et Mathan (L.). – Second order statistical measures for text-independant speaker identification. *Speech communication*, vol. 17, n° 1-2, Aug 1995, pp. 177–192.

[Bonastre et al. 00a] Bonastre (J-F.), Delacourt (P.), Fredouille (C.), Meignier (S.), Merlin (T.) et Wellekens (C.J.). – Différentes stratégies pour le suivi du locuteur. *Reconnaissance des Formes et Intelligence Artificielle RFIA2000*, pp. 123–129. – 2000.

[Bonastre et al. 00b] Bonastre (J-F.), Delacourt (P.), Fredouille (C.), Merlin (T.) et Wellekens (C.J.). – A speaker tracking system based on speaker turn detection for NIST evaluation. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* – 2000. To be published.

[Canny 83] Canny (J.F.). – *Finding edges and lines in images.* – Rapport technique, Massachusets Institute of Technology Artificial Intelligence Laboratory, 1983.

[Chen et al. 98a] Chen (S.), Gales (J.F.), Gopalakrishnan (P.), Gopinath (R.), Printz (H.), Kanevsky (D.), Olsen (P.) et Polymenakos (L.). – IBM's LVCSR system for transcription of broadcast news used in the 1997 HUB4 english evaluation. *DARPA speech recognition workshop.* – 1998.

[Chen et al. 98b] Chen (S.S.) et Gopalakrishnan (P.S.). – Clustering via the Bayesian Information Criterion with applications in speech recognition. *to be published ?* – 1998.

[Chen et al. 98c] Chen (S.S.) et Gopalakrishnan (P.S.). – Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion. *DARPA speech recognition workshop.* – 1998.

[Cohen 81] Cohen (J. R.). – Segmenting speech using dynamic programming. *J. Acoust. Soc. Am.*, vol. 69, n° 5, May 1981, pp. 1430–1438.

[Cook et al. 98] Cook (G.), Robinson (T.) et Christie (J.). – Real-time recognition of broadcast news. *International Conference on Spoken Language Processing*, pp. 1319–1322. – 1998.

[Cover et al. 91] Cover (T.M.) et Thomas (J.A.). – *Elements of information theory*, chap. 2. – John Wiley and sons Inc., 1991.

[Davis et al. 80] Davis (S.B.) et Mermelstein (P.). – Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on acoustics, speech and signal processing*, vol. 28, 1980.

[Delacourt et al. 99a] Delacourt (P.), et Wellekens (C.J.). – A first step into a speaker-based indexing system. *Workshop on Content-Based Multimedia Indexing.* – 1999.

[Delacourt et al. 99b] Delacourt (P.), Kryze (D.) et Wellekens (C.J.). – Detection of speaker changes in an audio document. *Eurospeech.* – 1999.

[Delacourt et al. 99c] Delacourt (P.), Kryze (D.) et Wellekens (C.J.). – Speaker-based segmentation for audio data indexing. *ESCA Workshop: Accessing Information in Audio Data.* – 1999.

[Delacourt et al. 99d] Delacourt (P.) et Wellekens (C.J.). – Audio data indexing: use of second-order statistics for speaker-based segmentation. *IEEE International Conference on Multimedia Computing and Systems.* – 1999.

[Delacourt et al. 99e] Delacourt (P.) et Wellekens (C.J.). – Segmentation en locuteurs d'un document audio. *CORESA : COmpression et REprésentation des Signaux Audiovisuels.* – 1999.

[Delacourt et al. 00] Delacourt (P.), et Wellekens (C.J.). – DISTBIC: a speaker-based segmentation for audio data indexing. *Speech Communication*, vol. 32, Sept. 2000. – Special issue on Accessing information in spoken audio, to be published.

[Dempster et al. 77] Dempster (D.), Laird (N.) et Rubin (D.). – Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Roy. Stat. Soc.*, 1977.

[Denenberg et al.93] Denenberg (L.), Gish (H.), Meteer (M.), Miller (T.), Rohlicek (J.R.), Sadkin (W.) et Siu (M.). – Gisting conversational speech in real time. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 131–134. – 1993.

[Duda et al. 73] Duda (R.O.) et Hart (P.E.). – *Pattern classification and scene analysis*. – John Wiley and Sons, Inc., 1973.

[Fiscus et al. 99] Fiscus (J.), Doddington (G.), Garofolo (J.) et Martin (A.). – NIST's 1998 topic detection and tracking evaluation. *Eurospeech*. – 1999.

[Fukunaga 90] Fukunaga (K.). – *Introduction to statistical pattern recognition*, chap. 9, p. 412. – Academic Press, Inc., 1990.

[Furui 81] Furui (S.). – Cepstral analysis technique for speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, n° 2, 1981, pp. 254–272.

[Furui 95] Furui (S.). – An overview of speaker recognition technology. *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*. – 1995.

[Gales 98] Gales (M.J.F.). – Cluster adaptative training for speech recognition. *International Conference on Spoken Language Processing*, pp. 1783–1786. – 1998.

[Gauvain et al. 92] Gauvain (J-L.) et Lee (C.-H.). – Bayesian Learning for Hidden Markov Model with Gaussian Mixture Observation of Markov Chains. *Speech Communication*, vol. 11, 1992, pp. 205–213.

[Gauvain et al. 94] Gauvain (J-L.) et Lee (C.-H.). – Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing*, vol. 2, n° 2, avril 1994, pp. 291–298.

[Gauvain et al. 97] Gauvain (J-L.), Adda (G.), Lamel (L.) et Adda-Decker (M.). – Transcribing broadcast news shows. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 715–718. – 1997.

[Gauvain et al. 98] Gauvain (J-L.), Lamel (L.) et Adda (G.). – Partitioning and transcription of broadcast news data. *International Conference on Spoken Language Processing*, pp. 1335–1338. – 1998.

[Gauvain et al. 99] Gauvain (J-L.), Lamel (L.), Adda (G.) et Jardino (M.). – The LIMSI 1998 HUB4-E transcription system. *DARPA Broadcast News Workshop*. – http://www.itl.nist.gov/iaui/894.01/proc/darpa99/index.htm, 1999.

[Gelin et al. 97] Gelin (P.) et Wellekens (C.J.). – Keyword spotting for multimedia document indexing. *SPIE97*. – 1997.

[Gish et al. 91] Gish (H.), Siu (M-H.) et Rohlicek (R.). – Segregation of speakers for speech recognition and speaker identification. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 873–876. – 1991.

[Gish et al. 94] Gish (H.) et Schmidt (N.). – Text-independent speaker identification. *IEEE Signal Processing Magazine*, oct. 1994, pp. 18–32.

[Gish 90] Gish (H.). – Robust discrimination in automatic speaker identification. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 289–292. – 1990.

[Godfrey et al. 92] Godfrey (J.J), Holliman (E.C.) et McDaniel (J.). – SWITCHBOARD: telephone speech corpus for research and development. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 517–520. – 1992.

[Golub et al. 96] Golub (G.H.) et Van Loan (C.F.). – *Matrix computations*. – The Johns Hopkins University Press, 1996, third édition, 70–73p.

[Hain et al. 98a] Hain (T.), Johnson (S.E.), Tuerk (A.), Woodland (P.C.) et Young (S.J.). – Segment generation and clustering in the HTK boradcast news transcription system. *DARPA Broadcast News Transcription and Understanding Workshop*. – 1998.

[Hain et al. 98b] Hain (T.) et Woodland (P.C.). – Segmentation and classification of broadcast news audio. *International Conference on Spoken Language Processing*, pp. 2727–2730. – 1998.

[Harris et al. 99] Harris (M.), Aubert (X.), Haeb-Umbach (R.) et Beyerlein (P.). – A study of broadcast news audio stream segmentation and segment clustering. *Eurospeech*. – 1999.

[Hayes 96] Hayes (M.H.). – *Statistical digital signal processing and modeling*, chap. 8, pp. 445–447. – John Wiley and sons Inc., 1996.

[Heck et al. 97] Heck (L.) et Sankar (A.). – Acoustic clustering and adaptation for robust speech recognition. *Eurospeech*. – 1997.

[Itakura 75] Itakura (F.). – Minimum prediction resiual principle applied to speech recognition. *IEEE Trans. ASSP*, 1975.

[Jin et al. 97] Jin (H.), Kubala (F.) et Schwartz (R.). – Automatic speaker clustering. *DARPA Speech Recognition Workshop*. – 1997.

[Johnson et al. 98] Johnson (S.E.) et Woodland (P.C.). – Speaker clustering using direct maximisation of the MLLR-adapted likelihood. *International Conference on Spoken Language Processing*, pp. 1775–1778. – 1998.

[Johnson 99] Johnson (S.E.). – Who spoke when ? - automatic segmentation and clustering for determining speaker turns. *Eurospeech*. – 1999.

[Kubala et al. 97] Kubala (F.), Jin (H.), Matsoukas (S.), Nguyen (L.), Schwartz (R.) et Makhoul (J.). – The 1996 BBN Byblos Hub-4 transcription system. *DARPA Speech Recognition Workshop*. – 1997.

[Leggetter et al. 95] Leggetter (C.J.) et Woodland (P.C.). – Maximum likelihood linear regression for speaker adaption of continuous density hidden Markov models. *Computer Speech and Language*, vol. 9, 1995, pp. 171–185.

[Linde et al. 80] Linde (Y.), Buzo (A.) et Gray (R.M). – An algorithm for vector quantizer design. *IEEE Transactions on Comm.*, vol. 28, Jan 1980.

[Liu et al. 99] Liu (D.) et Kubala (F.). – Fast speaker change detection for broadcast news transcription and indexing. *Eurospeech*, pp. 1031–1034. – 1999.

[Matsui et al. 91] Matsui (T.) et Furui (S.). – A text-independent speaker recognition method robust against utterance variations. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1991.

[Matsui et al. 92] Matsui (T.) et Furui (S.). – Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1992.

[Matsui et al. 95] Matsui (T.) et Furui (S.). – Likelihood normalization for speaker verification using a phoneme- and speaker-independent model. *Speech Communications*, vol. 17, 1995, pp. 109–116.

[Magrin-Chagnol. et al. 99] Magrin Chagnolleau (I.), Rosenberg (A.E.) et Parthasarathy (S.). – Detection of target speakers in audio databases. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1999.

[Magrin-Chagnol. 97] Magrin Chagnolleau (I.). – *Approches statistiques et filtrage vectoriel de trajectoires spectrales pour l'identification du locuteur indépendante*. – Thèse de PhD, Ecole Normale Supérieure des Télécommunications, 1997.

[Mclaughlin et al. 99] McLaughlin (J.), Reynold (D.), Singer (E.) et O'Leary (G.C). – Automatic speaker clustering from multi-speaker utterances. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1999.

[Meignier et al. 00] Meignier (S.), Bonastre (J.F.), Fredouille (C.) et Merlin (T.). – Evolutive HMM for multi-speaker tracking system. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 2000. To be published.

[Montacie et al. 92] Montacié (C.), Deleglise (P.), Bimbot (F.) et Caraty (M.J.). – Cinematic techniques for speech processing : temporal decomposition and multivariate linear prediction. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1992.

[Montacie et al. 97] Montacié (C.) et Caraty (M-J.). – Sound channel video indexing. *Eurospeech*, pp. 2359–2362. – 1997.

[Montacie et al. 98] Montacié (C.) et Caraty (M-J.). – A Silence/Noise/Music/Speech splitting algorithm. *International Conference on Spoken Language Processing*, pp. 1579–1582. – 1998.

[Moon 96] Moon (T.K.). – The Expectation-Maximisation algorithm. *IEEE Signal Processing Magazine*, 1996.

[Nishida et al. 98] Nishida (M.) et Ariki (Y.). – Real time speaker indexing based on subspace method: applications to TV news articles and debate. *International Conference on Spoken Language Processing*, pp. 1347–1350. – 1998.

[Nishida et al. 99] Nishida (M.) et Ariki (Y.). – Speaker indexing for news articles, debates and drama in broadcasted TV programs. *IEEE International Conference on Multimedia Computing and Systems*, pp. 466–471. – 1999.

[Oglesby 95] Oglesby (John). – What's in number? moving beyond the equal error rate. *Speech Communication*, 1995.

[Olsen 95] Olsen (J.O.). – Separation of speakers in audio data. *Eurospeech*, pp. 355–358. – 1995.

[Parris et al. 98] Parris (E.S.) et Carey (M.J.). – Multilateral techniques for speaker recognition. *International Conference on Spoken Language Processing*, pp. 1343–1346. – 1998.

[Pea 92] Press et al. (W.). – *Numerical recipes in C.* – Cambridge university press, 1992.

[Pye et al. 98] Pye (D.), Hollinghurst (N.J), Mills (T.J.) et Wood (K.R.). – Audio-visual segmentation for content-based retrieval. *International Conference on Spoken Language Processing*, pp. 1583–1586. – 1998.

[Bakis et al. 97] Bakis. (R), Chen (S.), Gopalakrishnan (P.), Gopinath (R.), Maes (S.), Polymenakos (L.) et Franz (M.). – Transcription of broadcast news shows with the IBM large vocabulary speech recognition system. *DARPA Speech Recognition Workshop*. – 1997.

[Rabiner et al. 78] Rabiner (L.R.) et Schafer (R.W.). – *Digital processing of speech signals*. – Prentince-Hall, 1978.

[Rapp et al. 98] Rapp (S.) et Dogil (G.). – Same news is good news: automatically collecting reoccuring radio news stories. *International Conference on Spoken Language Processing*, pp. 1587–1590. – 1998.

[Reynolds et al. 98] Reynolds (D.A.), Singer (E.), Carlson (B.A.), O'Leary (G.C.), McLaughlin (J.J.) et Zissman (M.A.). – Blind clustering of speech utterances based on speaker and language characteristics. *International Conference on Spoken Language Processing*, pp. 3193–3196. – 1998.

[Reynolds 95] Reynolds (D.A.). – Speaker identification and verification using Gaussian Mixture Models. *Speech Communication*, vol. 17, 1995, pp. 91–108.

[Reynolds 97] Reynolds (D.A.). – Comparison of background normalization methods for text-indpendent speaker verification. *Eurospeech*, pp. 963–966. – 1997.

[Rissanen 89] Rissanen (J.). – *Stochastic complexity in statistical inquiry*, chap. 3. – World Scientific, 1989, *Series in Computer Science*, volume 15.

[Rohlicek et al. 92] Rohlicek (J.R.), Ayuso (D.), Bates (M.), Bobrow (R.), Boualnger (A.), Gish (H.), Jeanreanud (P.), Meteer (M.) et Siu (M). – Gisting conversational speech. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 113–116. – 1992.

[Rose et al. 90] Rose (R.C.) et Reynolds (D.A.). – Text independent speaker identification using automatic acoustic segmentation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 293–296. – 1990.

[Rosenberg et al. 98] Rosenberg (A.E.), Magrin-Chagnolleau (I.), Parthasarathy (S.) et Huang (Q.). – Speaker detection in broadcast speech databases. *International Conference on Spoken Language Processing*, pp. 1339–1342. – 1998.

[Schwarz 78] Schwarz (G.). – Estimating the dimension of a model. *The Annals of Statistics*, vol. 6, n° 2, 1978, pp. 461–464.

[Seck et al. 99] Seck (M.), Bimbot (F.), Zugaj (D.) et Delyon (B.). – Two-class signal segmentation for speech/music in audio tracks. *Eurospeech*. – 1999.

[Siegler et al. 97] Siegler (M.A.), Jain (U.), Raj (B.) et Stern (R.M.). – Automatic segmentation, classification, and clustering of broadcast news audio. *DARPA Speech Recognition Workshop*. – 1997.

[Siu et al. 92] Siu (M-H.), Yu (G.) et Gish (H.). – An unsupervised, sequential learning algorithm for the segmentation of speech waveforms with multiple speakers. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1992.

[Solomonoff et al. 98] Solomonoff (A.), Mielke (A.), Schmidt (M.) et Gish (H.). – Clustering speakers by their voices. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. – 1998.

[Sonmez et al. 99] Sönmez (K.), Heck (L.) et Weintraub (M.). – Speaker tracking and detection with multiple speakers. *Eurospeech*. – 1999.

[Soong et al.88] Soong (F.K.) et Rosenberg (A.E.). – On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Transactions on acoustics, speech, and signal processing*, vol. 36, n° 6, 1988.

[Sugiyama et al. 93] Sugiyama (M.), Murakami (J.) et Watanabe (H.). – Speech segmentation and clustering based on speaker features. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 395–398. – 1993.

[Tritschler et al. 99] Tritschler (A.) et Gopinath (R.). – Improved speaker segmentation and segments clustering using the Bayesian Information Criterion. *Eurospeech*. – 1999.

[Tritschler 98] Tritschler (A.). – *A segmentation-enabled speech recognition application using the BIC criterion*. – Thèse, Institut EURECOM, France, 1998.

[Tsekeridou et al. 99] Tsekeridou (S.) et Pitas (I.). – Audio-visual content analysis for content-based video indexing. *IEEE International Conference on Multimedia Computing and Systems*. – 1999.

[Wilcox et al. 94] Wilcox (L.), Chen (F.), Kimber (D.) et Balasubramanian (V.). – Segmentation of speech using speaker identification. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 161–164. – 1994.

[Williams et al. 99] Williams (G.) et Ellis (D.P.W). – Speech/music discrimination based on posterior probability features. *Eurospeech*. – 1999.

[Woodland et al. 97] Woodland (P.C.), Gales (M.J.F.), Pye (D.) et Young (S.J.). – The development of the 1996 HTK broadcast news transcription system. *DARPA Speech Recognition Workshop*. – 1997.

[Yu et al. 93] Yu (G.) et Gish (H.). – Identification of speakers engaged in dialog. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 383–386. – 1993.