Analyse en locuteurs de collections de documents multimédia

Gaël Le Lan

UNIVERSITE BRETAGNE LOIRE

Thèse de Doctorat

Gaël LE LAN

Mémoire présenté en vue de l'obtention du
grade de Docteur de Le Mans Université
sous le sceau de l'Université Bretagne Loire

École doctorale : STIM 503

Discipline : Informatique
Unité de recherche : ***Orange Labs - LIUM***

Soutenue le 6 octobre 2017
Thèse N° : 2017LEMA1020

Analyse en locuteurs de collections de documents multimédia

JURY

Rapporteurs :	**Claude BARRAS**, Maitre de Conférences, HDR, LIMSI, Université Paris-Sud **Jean-François BONASTRE**, Professeur des Universités, LIA, Université d'Avignon
Examinateurs :	**Corinne FREDOUILLE,** Maitre de Conférences, HDR, LIA, Université d'Avignon **Guillaume GRAVIER**, Directeur de Recherche, IRISA, Rennes
Invité(s) :	**Yannick ESTEVE**, Professeur des Universités, LIUM, Le Mans Université **Jean-Hugh THOMAS**, Maitre de Conférences, HDR, Le Mans Université
Directeur de Thèse :	**Sylvain MEIGNIER**, Professeur des Universités, LIUM, Le Mans Université
Encadrants de Thèse :	**Delphine CHARLET**, Chercheur, Orange Labs, Lannion **Anthony LARCHER**, Maitre de Conférences, LIUM, Le Mans Université

Remerciements

Mes premiers remerciements s'adressent à Jean-François Bonastre et Claude Barras pour avoir accepté d'être rapporteurs et pour leurs retours constructifs, ainsi qu'aux membres de mon jury de thèse que sont Corrine Fredouille, Yannick Estève, Jean-Hugh Thomas et Guillaume Gravier.

Un merci tout particulier à mon directeur de thèse, Sylvain Meignier, pour sa disponibilité et pour la confiance qu'il m'a accordée lors de ces trois années de thèse. J'ai constamment eu le sentiment de savoir dans quelle direction nous allions et c'était très agréable.

Je remercie également Delphine Charlet, pour son accompagnement et sa disponibilité au quotidien et pour les nombreuses discussions passionnées que nous avons eues. Merci à Anthony Larcher pour l'accompagnement aux conférences, pour les intuitions, les discussions techniques... et moins techniques !

Merci également à mes autres collègues lannionnais : Géraldine, Johannes, Olivier, Frédéric I et Frédéric II pour les moments conviviaux partagés au travail.

Je tiens également à remercier Joan, Julien et Robin pour tous ces vendredi soirs passés à rigoler, même lorsque l'électricité faisait défaut.

Enfin, un dernier merci pour Elaine qui a su trouver les mots pour m'encourager dans les moments difficiles, pour les nombreuses balades sur le sentier des douaniers, et sans qui cette thèse n'aurait pas eu lieu d'être.

Résumé

La segmentation et regroupement en locuteurs (SRL) de collection cherche à répondre à la question « qui parle quand ? » dans une collection de documents multimédia. C'est un prérequis indispensable à l'indexation des contenus audiovisuels. La tâche de SRL consiste d'abord à segmenter chaque document en locuteurs, avant de les regrouper à l'échelle de la collection. Le but est de positionner des labels anonymes identifiant les locuteurs, y compris ceux apparaissant dans plusieurs documents, sans connaître à l'avance ni leur identité ni leur nombre. La difficulté posée par le regroupement en locuteurs à l'échelle d'une collection est le problème de la variabilité intra-locuteur/inter-document : selon les documents, un locuteur peut parler dans des environnements acoustiques variés (en studio, dans la rue...). Cette thèse propose deux méthodes pour pallier le problème. D'une part, une nouvelle méthode de compensation neuronale de variabilité est proposée, utilisant le paradigme de *triplet-loss* pour son apprentissage. D'autre part, un procédé itératif d'adaptation non supervisée au domaine est présenté, exploitant l'information, même imparfaite, que le système acquiert en traitant des données, pour améliorer ses performances sur le domaine acoustique cible. De plus, de nouvelles méthodes d'analyse en locuteurs des résultats de SRL sont étudiées, pour comprendre le fonctionnement réel des systèmes, au-delà du classique taux d'erreur de SRL (*Diarization Error Rate* ou DER). Les systèmes et méthodes sont évalués sur deux émissions télévisées d'une quarantaine d'épisodes, pour les architectures de SRL globale ou incrémentale, à l'aide de la modélisation locuteur à l'état de l'art (*i-vector*).

Mots-Clés : segmentation et regroupement en locuteurs, réseau de neurones, adaptation au domaine, apprentissage supervisé, apprentissage non supervisé

Abstract

The task of speaker diarization and linking aims at answering the question « who speaks and when ? » in a collection of multimedia recordings. It is an essential step to index audiovisual contents. The task of speaker diarization and linking firstly consists in segmenting each recording in terms of speakers, before linking them across the collection. Aim is, to identify each speaker with a unique anonymous label, even for speakers appearing in multiple recordings, without any knowledge of their identity or number. The challenge of the cross-recording linking is the modeling of the within-speaker/across-recording variability : depending on the recording, a same speaker can appear in multiple acoustic conditions (in a studio, in the street...). This thesis proposes two methods to overcome this issue. Firstly, a novel neural variability compensation method is proposed, using the *triplet-loss* paradigm for training. Secondly, an iterative unsupervised domain adaptation process is presented, in which the system exploits the information (even inaccurate) about the data it processes, to enhance its performances on the target acoustic domain. Moreover, novel ways of analyzing the results in terms of speaker are explored, to understand the actual performance of a diarization and linking system, beyond the well-known Diarization Error Rate (DER). Systems and methods are evaluated on two TV shows of about 40 episodes, using either a global or longitudinal linking architecture, and state of the art speaker modeling (*i-vector*).

Keywords : speaker diarization and linking, neural network, domain adaptation, unsupervised training, supervised training

Sommaire

Chapitre 1

Introduction

1.1 Contexte

Avec l'explosion des volumes de données audio et vidéo produits chaque jour dans le monde (réseaux sociaux, médias traditionnels, conférences, réunions...) a émergé le besoin d'indexer automatiquement les personnes, les langues et les thèmes. Derrière ces problématiques se cachent des applications classiques du traitement automatique de la parole, telles que la reconnaissance du locuteur et de la langue, ou la transcription automatique. Pour l'indexation des personnes, dont l'objectif est de répondre de manière automatique à la question « qui parle quand ? » dans une collection de documents multimédia, la première étape est la Segmentation et Regroupement en Locuteurs (SRL, en anglais *Speaker Diarization and Linking*), suivie de l'identification des locuteurs.

La tâche de Segmentation et Regroupement en Locuteurs (SRL) a été popularisée par le *National Institute of Standards and Technology* (NIST), lors de l'organisation des campagnes d'évaluation *Rich Transcription* (RT) [NIST, 2003]. L'objectif de ces campagnes était d'améliorer les résultats des systèmes de transcription automatique de la parole en indiquant les tours de parole des différents locuteurs.

Historiquement, la SRL vise donc à étiqueter les locuteurs dans un document audio (on parle de SRL intra-document). On cherche d'abord à détecter les changements de locuteurs, ce qui permet d'en délimiter des segments, d'où le terme **segmentation**. Ensuite, on cherche à déterminer quels sont les segments (ou tours de parole) ayant été prononcés par un même locuteur. On souhaite donc les regrouper par locuteur, d'où le terme **regroupement**. La tâche globale est donc la segmentation et regroupement en locuteurs, dont le but est de positionner des labels anonymes identifiant les locuteurs. Cette tâche présente la difficulté de n'avoir aucune connaissance a priori sur l'identité des locuteurs ou leur nombre. Pour finaliser l'indexation des locuteurs, il est donc nécessaire de réaliser une étape d'identification, où l'on remplace les labels anonymes par l'identité réelle des locuteurs. [Tranter and Reynolds, 2006] et [Anguera et al., 2012] fournissent un historique assez complet

des différentes méthodes développées pour la SRL jusqu'au début des années 2010.

La question de la SRL appliquée à des collections de documents s'est posée plus récemment [Tran et al., 2011a; Yang et al., 2011a; **?**]. Cette fois, il s'agit de traiter non plus un seul mais plusieurs documents afin d'identifier de manière unique les tours de parole de chaque locuteur à l'échelle de la collection, y compris les locuteurs dit récurrents, qui apparaissent dans plusieurs documents. L'application visée, l'indexation des locuteurs, doit permettre d'apporter un nouveau moyen d'explorer les collections telles que des archives audiovisuelles, dans lesquelles on trouve bon nombre de locuteurs récurrents (journalistes, présentateurs, politiques, personnalités du monde culturel...). Conceptuellement, elle est donc vue comme une étape de regroupement inter-document, supplémentaire à la SRL intra-document. Les documents peuvent contenir des prises de parole espacées dans le temps et dans des conditions acoustiques variées. La difficulté supplémentaire posée par le regroupement inter-document est donc la question de la variabilité intra-locuteur/inter-document.

Dans ce manuscrit, nous nous intéressons particulièrement à la **SRL de Collection** dédiée à l'indexation en locuteurs d'archives audiovisuelles. Les contenus audiovisuels se caractérisent par la grande variabilité de leur structure éditoriale, ce qui induit un nombre de locuteurs pouvant varier fortement d'une émission à l'autre, ainsi que la variété de l'environnement acoustique, dépendant du canal d'enregistrement (studio, téléphonique, dans la rue...). Il peut également y avoir de très courtes prises de parole (micro-trottoir) et de la parole superposée, dont la gestion peut constituer une difficulté pour la machine.

Pour faire progresser les systèmes de SRL dédiés aux collections de documents audiovisuels, des campagnes d'évaluation ont été organisées ces dernières années. La campagne REPERE [Galibert and Kahn, 2013a] était consacrée à l'évaluation de la reconnaissance multimodale des personnes dans des émissions audiovisuelles. A ce titre, l'évaluation de la SRL était au programme. Pour évaluer la qualité d'un système de SRL de collection, on utilise généralement une métrique appelée *taux d'erreur de SRL* (en anglais *Diarization Error Rate* ou DER), intra-document lorsqu'on évalue la performance sur un seul document, inter-document lorsqu'on travaille à l'échelle d'une collection. A titre d'exemple, les taux d'erreur de SRL inter-document des systèmes en compétition lors de la campagne REPERE variaient de 14.2% à 33.1%. On peut également citer la campagne d'évaluation *Multi-Genre Broadcast* [Bell et al., 2015], dédiée à l'exploitation d'émissions de la chaîne anglaise *BBC* et à laquelle nous avons participé en début de thèse, où les taux d'erreur des systèmes en compétition se situaient à plus de 40%. Les émissions à traiter étaient particulièrement difficiles puisqu'il s'agissait notamment de commentaires de matchs de football très bruités, ou de séries où les locuteurs présentaient une voix robotique. Cette dernière campagne a mis en évidence le fait qu'il existait encore une marge de progression.

1.2 Problématique

Répondre à la question « qui parle quand ? » suppose d'être capable de représenter la voix des locuteurs et de les comparer. Les derniers systèmes de SRL de collection à l'état de l'art nécessitent un apprentissage supervisé : pour apprendre à la machine à différencier les personnes à partir de leur voix, on doit lui fournir plusieurs exemples de différents locuteurs, et ce, en grande quantité. Cependant, il n'est pas nécessaire que les locuteurs exemples soient les mêmes que les locuteurs à différencier. En effet, la machine apprend à modéliser les variabilités intra- et inter-locuteurs plutôt que les locuteurs eux-mêmes. Le problème est que généralement, ces exemples sont produits, c'est-à-dire annotés, par des opérateurs humains, faute d'une précision suffisante de la machine, ce qui représente un coût important. De plus, la grande variabilité des contenus traités fait qu'il n'est pas toujours possible de produire de tels exemples pour une application précise. Par exemple, lorsqu'on cherche à traiter de l'allemand alors qu'on ne dispose que d'exemples en français (différence de langue), ou encore lorsqu'on cherche à traiter des documents télévisuels, alors qu'on ne dispose que d'exemples téléphoniques (différence de canal), les résultats sont souvent dégradés. C'est pourquoi il est toujours bénéfique pour la machine de disposer de données annotées représentatives d'un domaine acoustique donné.

Or, il existe actuellement une quantité importante de données brutes (non annotées) qui, si elles pouvaient être exploitées de façon non supervisée par la machine, pourraient l'aider à être plus performante. Actuellement, un système de SRL à l'état de l'art est imparfaitement capable de différencier les locuteurs. La problématique de cette thèse est donc la suivante : **la machine peut-elle utiliser l'information, même imparfaite, qu'elle acquiert lorsqu'elle traite des données, pour améliorer ses capacités sur le domaine acoustique traité ?**

Par ailleurs, historiquement, la performance d'un système de SRL est évaluée par un taux d'erreur global. Si ce taux d'erreur est une métrique très pratique pour comparer des systèmes, il ne permet pas toujours de dire si un système répond au problème posé. Si, par exemple, nous souhaitons indexer les prises de parole de personnalités politiques dans le paysage audiovisuel français et que notre système de SRL présente un taux d'erreur de 20%, avons-nous répondu au problème ? Peut-être notre système est-il très performant sur un type de locuteur particulier (par exemple les présentateurs de journal télévisé), mais très mauvais pour segmenter et regrouper les locuteurs présents dans les reportages ? Ce type de conclusion n'étant pas lisible à travers le taux d'erreur de SRL, **peut-on proposer des méthodes d'analyse en locuteurs pour comprendre le fonctionnement réel d'un système ?**

1.3 Plan du manuscrit

Hormis un court chapitre consacré à la présentation des données expérimentales (chapitre 2), ce manuscrit est composé de deux parties. La première partie, constituée des chapitres 3, 4 et 5, aborde la question de la SRL de collection tandis la seconde, composée des chapitres 6, 7 et 8 traite de l'adaptation au domaine d'un système de SRL à l'état de l'art.

Au chapitre 2, nous commençons par présenter les données étudiées dans ce manuscrit. Nous y définissons la notion de collection de documents multimédia, construisons nos collections expérimentales et en analysons la composition en locuteurs : identité, rôles, durée de parole...

Ensuite, dans une première partie, composée des chapitres 3, 4 et 5, nous abordons la problématique de la segmentation et regroupement en locuteurs appliquée à des collections. Le chapitre 3 est consacré à un état de l'art, prérequis théorique nous permettant de concevoir notre propre système de SRL de collection, à l'état de l'art, au chapitre 4. Nous en évaluons les performances, les limites, notamment à travers une analyse en locuteurs, sur les deux collections définies au chapitre 2. Le chapitre 5 présente une nouvelle méthode de SRL utilisant des réseaux de neurones.

La seconde partie, constituée des chapitres 6, 7 et 8, présente différentes contributions liées à la question de l'adaptation au domaine des systèmes de SRL. Au chapitre 6, nous présentons la problématique, notamment à travers un état de l'art, avant de proposer une stratégie d'adaptation au domaine pour la SRL et de l'évaluer. Le chapitre 7 est consacré à l'influence de la taille des collections traitées sur les performances de l'adaptation, tandis que le chapitre 8 traite de la question particulière de la SRL incrémentale dans le contexte de l'adaptation au domaine.

Enfin, le chapitre 9 résume les principaux résultats et contributions de cette thèse, ainsi que les axes de recherche futurs possibles.

Chapitre 2

Collections

Résumé

Dans ce chapitre, nous définissons la notion de collection de documents multimédia, avant de présenter les données à notre disposition, à partir desquelles nous définissons deux collections expérimentales intitulées LCP et BFM, correspondant à deux émissions télévisuelles. Nous les analysons du point de vue locuteur, en définissant quatre types de locuteurs, selon qu'ils sont journalistes ou invités, ponctuels (présents dans un seul document) ou récurrents (dans plus d'un document). D'une part, les locuteurs récurrents comptent pour plus de la moitié du temps de parole total de chaque collection. D'autre part, la structure éditoriale fait que la distribution des durées moyennes de parole par document est bi-modale pour les invités, et qu'une majorité d'entre eux parle en moyenne moins de 40 secondes par épisode. Ces deux informations montrent que le choix de ces collections est pertinent pour évaluer la tâche de SRL. Enfin, on définit un corpus d'apprentissage constitué d'émissions radiophoniques qui servira à la construction (l'apprentissage) d'un système de SRL.

2.1 Définition des collections

Pour évaluer la tâche de SRL de collection, il est nécessaire de définir ce qu'est une collection. Pour [Dupuy, 2015], une collection est définie comme un ensemble de documents multimédia qui présentent des caractéristiques communes. Dans le cadre de cette thèse, nous souhaitons travailler sur des collections qui ont en commun la présence de certains locuteurs. En effet, un des objectifs de la thèse est de proposer un nouveau moyen d'exploration des collections, via les locuteurs. Dans ce cadre, et au vu des données disponibles, nous proposons de travailler sur des collections correspondant à des émissions audiovisuelles. En effet, une émission télévisée a une certaine périodicité (journalière, hebdomadaire...), et certains locuteurs peuvent apparaître dans plusieurs épisodes.

A ce titre, nous proposons de travailler avec des données issues des campagnes d'évaluation REPERE [Galibert and Kahn, 2013a], ETAPE [Galibert et al., 2014] et ESTER [Galliano et al., 2009] pour générer plusieurs *corpora* expérimentaux. Ces données constituent les plus gros *corpora* audiovisuels disponibles comprenant des annotations en locuteurs précises. A titre de comparaison, les données du challenge MGB [Bell et al., 2015], bien que constituées de plus de 500 heures d'émissions, étaient très peu annotées.

L'ensemble des émissions qui constituent les données de REPERE, ETAPE et ESTER représentent 220 heures de diffusion (principalement de la chaîne radio France Inter et des chaînes de télévision LCP et BFM, mais pas seulement). Les émissions, de durées variant de dix minutes à une heure, ont été diffusées entre 1998 et 2012. Chaque émission du corpus comprend plusieurs épisodes. Pour chaque épisode, les locuteurs ont été annotés manuellement par leur nom et prénom, ainsi qu'en rôles. 5 rôles ont été définis (voir table 2.1), les catégories R1, R2 et R3 constituant les **journalistes** et les catégories R4 et R5 correspondant aux **invités**. Certains locuteurs peuvent apparaître dans plusieurs épisodes d'une même émission, voire dans plusieurs émissions. Les locuteurs apparaissant dans plus d'un épisode d'une émission sont appelés locuteurs **récurrents**, par opposition aux locuteurs **ponctuels**, qui ne sont présents que dans un épisode.

label	rôle
R1	présentateur principal
R2	journaliste plateau
R3	journaliste terrain
R4	invité plateau
R5	invité terrain

TABLE 2.1: Description des différents rôles annotés dans les corpora

Pour que l'analyse en locuteurs d'une collection soit pertinente, il faut s'intéresser aux émissions contenant un nombre d'épisodes relativement important. Dans la table 2.2, les émissions du corpus REPERE sont détaillées. Ce sont les émissions pour lesquelles le plus grand nombre d'épisodes est disponible. Le corpus REPERE est composé d'émissions télévisuelles des chaînes LCP et BFM, dont les émissions mettent en scène des débats, des reportages, des interviews. Ce corpus a été constitué pour évaluer les méthodes d'identification des personnes de manière multi-modale (à l'aide de la reconnaissance de visages, de la voix des locuteurs, de la transcription automatique et des textes incrustés à l'écran). Il a fait l'objet de trois campagnes d'évaluation de 2012 à 2014 [Galibert and Kahn, 2013a]. Dans la table, nous avons également indiqué le nombre de locuteurs ponctuels et récurrents.

Nous notons également la distinction entre durée totale (des enregistrements) et durée annotée. En effet, pour des raisons de coût, les organisateurs des campagnes

d'évaluation ont préféré faire annoter un grand nombre de documents partiellement plutôt qu'un nombre plus faible mais intégralement. Les émissions disponibles sont donc annotées partiellement. Pour chaque segment de parole annoté est indiquée l'identité du locuteur, dont son rôle, ainsi que la transcription manuelle.

émission	#épisodes	durée totale	durée annotée	#locuteurs	ponctuels	récurrents
BFM Story	42	43h12m	**19h47m**	**422**	345	**77**
Culture Et Vous	87	29h41m	2h04m	242	201	41
Planete Showbiz	73	08h46m	1h41m	195	164	31
Ca Vous Regarde	21	18m51m	4h49m	136	124	12
Entre Les Lignes	23	14h02m	4h46m	17	9	8
LCP Info	45	19h19m	**10h01m**	**220**	127	**93**
Pile Et Face	31	18h38m	4h40m	43	28	15
Top Questions	31	11h10m	5h52m	95	60	35

TABLE 2.2: Description des différentes émissions du corpus REPERE

Au regard de la durée annotée des émissions, du nombre d'épisodes disponibles et du nombre de locuteurs, l'émission la plus intéressante pour une approche par collection est *BFM Story*. *LCP Info* présente également un intérêt, le ratio durée de parole totale/nombre de locuteurs étant sensiblement proche et l'émission possèdant une centaine de locuteurs récurrents. Le nombre d'épisodes disponibles pour ces deux émissions, de l'ordre de la quarantaine, approche le nombre d'épisodes d'un hebdomadaire diffusé sur une année (compte-tenu de la pause estivale des programmes).

Nous définissons deux collections cibles, construites à partir de la fusion des *corpora* officiels d'apprentissage et de test de la campagne REPERE. La première, que nous appellerons LCP, est la collection de tous les épisodes disponibles de l'émission *LCP Info*. La seconde, que nous appellerons BFM, contient tous les épisodes disponibles de l'émission *BFM Story*.

Corpus	LCP	BFM
Nombre d'épisodes	45	42
Durée d'un épisode	25m	60m
Durée de parole annotée	10h08m	19h57m
Nombre de locuteurs...		
Ponctuels	127	345
Récurrents dans 2 épisodes ou plus	93	77
Récurrents dans 3 épisodes ou plus	48	35
Total	220	422
Part du temps de parole total pour les locuteurs...		
Ponctuels	20.1%	44,8%
Récurrents dans 2 épisodes ou plus	79,9%	55,2%
Récurrents dans 3 épisodes ou plus	67,1%	45,9%
Nombre de locuteurs moyen par épisode	16	18
Durée de parole moyenne d'un locuteur, par épisode	1m08s	1m58s

TABLE 2.3: Composition des deux collections cibles. L'annotation étant partielle, seules les statistiques de la parole annotée sont présentées.

2.2 Analyse en locuteurs

Les deux collections contiennent de nombreux locuteurs récurrents, qui parlent pour plus de 50% du temps de parole total de la collection. Quelques détails relatifs à la composition en locuteurs des deux collections sont présentés dans le tableau 2.3. Les deux collections étant partiellement annotées, seuls les chiffres des locuteurs annotés sont présentés.

2.2.1 Détail des collections cibles

Pour chaque collection cible, nous proposons d'en analyser la composition en locuteurs, en s'intéressant notamment aux distributions des temps de parole et des rôles des locuteurs.

2.2.1.1 Collection BFM

La collection BFM est constituée de 42 épisodes d'une heure, au sein desquels nous dénombrons en moyenne 18 locuteurs. Au total nous y comptons 422 locuteurs, dont 77 parlent dans plus d'un épisode. Dans l'optique d'une analyse par collection, nous proposons de décomposer l'analyse des locuteurs selon qu'ils sont ponctuels ou récurrents.

Lors de nos premières analyses sur les locuteurs récurrents, nous avons constaté que nous ne pouvions pas tous les catégoriser en rôles uniques selon la typologie définie dans REPERE : certains individus peuvent avoir des rôles multiples. En effet, une même personne peut être présente en plateau ou sur le terrain, selon l'épisode considéré. Parmi les 43 journalistes récurrents présents, 6 prennent des rôles différents selon les épisodes (présentateur principal, journaliste plateau ou terrain), et parmi les 35 invités récurrents, 14 apparaissent en plateau et sur le terrain. En revanche, dans cette collection, jamais un journaliste n'est catégorisé invité, et inversement. Ainsi, nous proposons de simplifier la catégorisation des locuteurs selon deux critères : ponctuel ou récurrent, journaliste ou invité. c'est-à-dire que nous fusionnons les rôles R1, R2 et R3 d'une part, R4 et R5 d'autre part, tels qu'ils étaient définis dans le corpus REPERE. Dans la table 2.4, nous affichons la répartition du temps de parole entre journalistes et invités, ponctuels et récurrents. Nous pouvons constater qu'il y a une forte distinction entre les locuteurs ponctuels, qui sont très souvent des invités, et les locuteurs récurrents, où, si le temps de parole est en faveur des journalistes, la répartition en nombre d'individus est à peu près équilibrée.

D'après la table 2.4, nous notons que les deux types de locuteurs les plus représentés sont les invités ponctuels et les journalistes récurrents, qui représentent respectivement 39.2% et 45.8% du temps de parole de la collection BFM. Si la représentation du temps de parole est équilibrée entre les deux, le nombre de journalistes récurrents est bien plus faible (43) que celui d'invités ponctuels (310). Corollaire,

le temps de parole moyen par émission des journalistes récurrents dépasse les 3 minutes, tandis que celui des autres classes de locuteurs est d'environ 1 minutes 30 secondes.

Classe de locuteurs	journalistes ponctuels	invités ponctuels	journalistes récurrents	invités récurrents
#locuteurs	48	297	42	35
temps de parole moyen par épisode	1m20s	1m30s	3m06s	1m34s
%temps de parole total	5.4%	39.2%	45.8%	9.6%

TABLE 2.4: Répartition de tous les locuteurs en quatre classes (journalistes = R1/R2/R3 & invités = R4/R5), selon qu'ils sont récurrents ou non, pour la collection BFM.

Concernant le temps de parole total des locuteurs (somme des temps de parole dans la collection), nous constatons que les 8 locuteurs les plus diserts sont des journalistes plateaux, récurrents, pour une durée de parole totale allant d'un peu plus de 12 minutes à plus de 2 heures pour le présentateur de l'émission. Ces 8 locuteurs, présentés en table 2.5, accaparent 35% du temps de parole total. En fin de liste (non présentée dans ce manuscrit), nous dénombrons 40% des locuteurs avec un temps de parole total inférieur à 30 secondes.

Nom du locuteur	Temps de parole total	Récurrence
Olivier TRUCHOT	2h09m07s	39
Jean-Remi BAUDOT	51m29s	9
Thomas LEQUERTIER	36m23s	6
Rachid M'BARKI	25m22s	8
Alain MARSCHALL	22m16s	4
Frederic DE LANOUVELLE	18m19s	4
Thomas MISRACHI	13m26s	2
Damien GOURLET	12m14s	3

TABLE 2.5: Liste des 8 locuteurs les plus diserts de la collection BFM.

Pour avoir un aperçu de la répartition des temps de parole moyens par épisode selon les types de locuteurs, nous les représentons sous forme d'histogramme, sur la figure 2.1. Chaque histogramme présente la distribution des temps de parole moyens par épisode pour un type de locuteur. Par exemple, le premier graphe permet de lire l'information : 33% des journalistes ponctuels ont un temps de parole moyen par épisode situé entre 40 et 60 secondes.

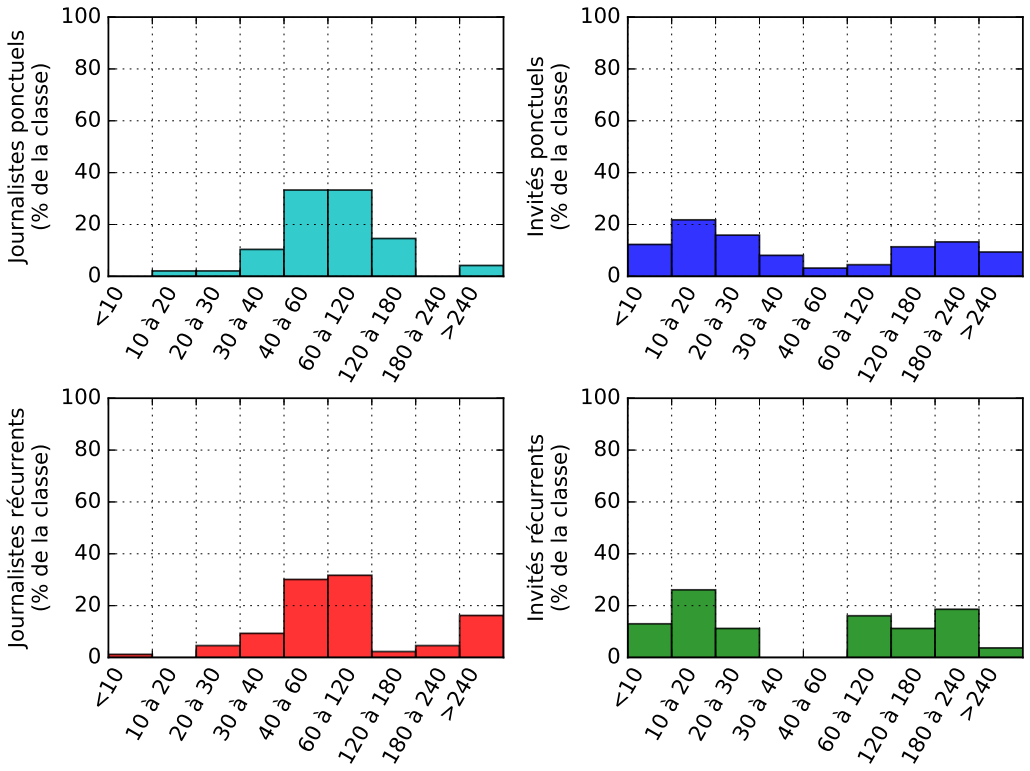

FIGURE 2.1 – Distribution du temps de parole moyen par épisode, en secondes, des différents types de locuteurs (en ordonnées, pourcentage des individus du type considéré), pour la collection BFM.

Sur la figure, les graphes de la première colonne représentent les journalistes. Nous remarquons assez clairement que les deux histogrammes sont assez similaires, avec une majeure partie (plus de 60%) du temps de parole moyen par épisode entre 40 secondes et 2 minutes. Qu'ils soient récurrents ou non, une très faible proportion des 91 journalistes de la collection affiche un temps de parole moyen par épisode inférieur à la trentaine de secondes. En revanche, parmi les invités, qu'ils soient ponctuels ou récurrents, la répartition des temps de parole se fait selon deux modes. On distingue les locuteurs dont le temps de parole par document est inférieur à 40 secondes (voire 30 secondes chez les récurrents), et ceux dont le temps de parole est supérieur à la minute. Cela illustre bien deux classes d'intervenants : les invités faisant l'objet d'un reportage, par exemple interrogés sur un sujet d'actualité, dont on diffuse un court extrait, et les invités « débat », qui disposent d'un temps de parole beaucoup plus conséquent, sous forme d'échanges. La répartition est équilibrée avec environ 50% des invités (qu'ils soient récurrents ou non) dans chaque mode. La présence importante de locuteurs dont le temps de parole avoisine la vingtaine de secondes est un critère à prendre en compte quand on sait que l'environnement acoustique des invités terrains peut être bruité (par exemple, bruits de circulation routière, applaudissements...).

2.2.1.2 Collection LCP

La collection LCP est constituée de 45 épisodes de 25 minutes, avec en moyenne 16 locuteurs par épisode. Comme les épisodes sont deux fois plus courts que la collection précédente, la durée moyenne par type de locuteur est plus faible, comme nous pouvons le lire dans la table 2.6. Nous y avons regroupé les locuteurs selon qu'ils sont journalistes ou invités, ponctuels ou récurrents. Encore une fois, la répartition du temps de parole entre invités et journalistes est équilibrée (environ 50% pour chaque), mais le grand nombre d'invités donne un temps de parole moyen par épisode inférieur à la minute. Notons la très grande proportion de la durée des locuteurs récurrents (plus de 80%), ce qui rend la collection intéressante dans une optique d'exploration par locuteur.

Classe de locuteurs	journalistes ponctuels	invités ponctuels	journalistes récurrents	invités récurrents
#locuteurs	5	126	24	73
temps de parole moyen par épisode	1m02s	52s	1m40s	37s
%temps de parole total	0.9%	18.3%	51.2%	29.7%

TABLE 2.6: Répartition de tous les locuteurs en deux classes (journalistes = R1/R2/R3 & invités = R4/R5), pour la collection LCP.

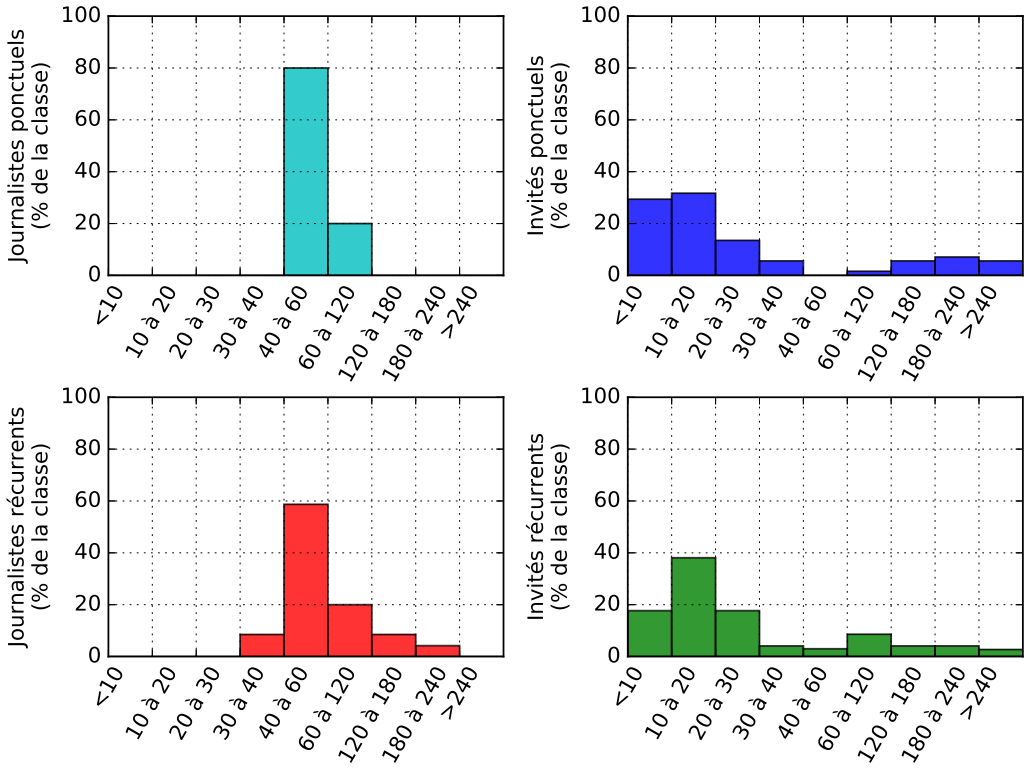

FIGURE 2.2 – Distribution du temps de parole moyen par épisode, en secondes, des différents types de locuteurs (en ordonnées, pourcentage des individus du type considéré), pour la collection LCP.

Du côté des distributions de temps de parole moyen par type de locuteur (figure 2.2), nous observons une allure assez proche de ceux de la collection BFM, avec deux différences significatives. Pour les journalistes, la répartition des temps de parole moyens est très concentrée autour de la minute, avec respectivement 80% et 60% des individus sur la tranche *40 à 60 secondes*. Nous notons même chez les journalistes ponctuels qu'aucun individu ne parle plus de 2 minutes dans un épisode, tandis que chez les récurrents la distribution est plus étalée, notamment en raison du présentateur principal de l'émission. Concernant les invités, nous devinons une distribution bimodale, mais cette fois la répartition des locuteurs selon les deux modes est déséquilibrée. La plupart des invités (environ 80%) parlent en moyenne moins de 40 secondes par épisode, qu'ils soient ponctuels ou récurrents. Cependant, si le premier mode de la distribution est centré sur la tranche *10 à 20* pour les deux types d'invités, nous constatons que le deuxième mode de la distribution des invités ponctuels est centré sur la tranche *180 à 240*. Chez ces locuteurs, les deux modes sont même clairement séparés, puisqu'un invité ponctuel parle en moyenne soit moins de 40 secondes, soit plus d'une minute par épisode. Chez les invités récurrents, le deuxième mode est plus étalé.

2.2.2 Corpus d'apprentissage

En complément des deux collections cibles, nous définissons un corpus pour les tâches nécessitant un apprentissage supervisé. Il contient les épisodes de toutes les émissions radiophoniques disponibles. Il s'agit de 313 documents audio issus des corpus d'entraînement et de développement des campagnes ETAPE, ESTER et REPERE. Nous y dénombrons 3888 locuteurs différents, dont 370 apparaissent dans au moins trois documents avec un temps de parole minimal de dix secondes par document.

Quelques locuteurs des collections cibles sont présents dans les données d'apprentissage : 24 dans la collection BFM, parmi lesquels 6 y sont récurrents et 27 dans la collection LCP, parmi lesquels 10 y sont récurrents. Ce sont principalement des hommes et femmes du paysage politique français des deux dernières décennies : les données d'apprentissage sont issues d'émissions diffusées entre 1998 et 2010, tandis que les épisodes des émissions cibles datent des années 2011 et 2012. Il n'y a donc pas de recouvrement temporel entre les données d'apprentissage et les collections cibles.

2.3 Bilan

Dans ce chapitre, nous avons défini deux collections expérimentales, LCP et BFM, issues des données de la campagne d'évaluation REPERE. Elles comprennent

chacune une quarantaine de documents télévisuels et contiennent plusieurs centaines de locuteurs. Les locuteurs récurrents y sont très présents, puisqu'ils constituent plus de 50% du temps de parole de chaque collection. A partir des annotations en rôle des locuteurs, nous avons constitué quatre classes de locuteurs, selon qu'ils sont journalistes ou invités, ponctuels ou récurrents.

La durée de temps de parole moyen par document d'un locuteur est de l'ordre de la minute, mais pour les invités, nous avons mis en évidence une répartition bimodale. En résumé, ces deux collections sont donc pertinentes du point de vue de la SRL de collection, en raison de la présence importante de locuteurs récurrents. Par ailleurs, la diversité des conditions acoustiques d'enregistrement (en plateau, dans la rue...) ainsi que la faible durée de parole par document d'un grand nombre d'invités peut rendre la tâche difficile, et donc mettre en évidence des axes d'amélioration.

Enfin, pour concevoir notre futur système de SRL, nous avons défini un corpus d'apprentissage de 313 documents radiophoniques, contenant 370 locuteurs récurrents. S'il existe quelques locuteurs communs aux collections cibles et au corpus d'apprentissage, il n'y a pas de recouvrement temporel.

Première partie

Segmentation et Regroupement en Locuteurs

Chapitre 3

Etat de l'art en SRL

Résumé

Ce chapitre est consacré à un état de l'art de la Segmentation et Regroupement en Locuteurs. Il débute par la définition de la tâche, avant d'aborder la question de la modélisation acoustique de segments de parole du point de vue locuteur. Les concepts théoriques de la modélisation des locuteurs nous permettent ensuite d'étudier les architectures classiques des systèmes de SRL, d'abord intra-document, puis de collection (inter-document). On y met en évidence les modèles classiques de locuteurs utilisés pour la SRL, ainsi que les méthodes de regroupement usuelles utilisées par la communauté scientifique. Le chapitre se conclut par l'étude des mesures d'évaluation de la SRL intra- et inter-document.

3.1 Définition

La tâche de Segmentation et Regroupement en Locuteurs (SRL, en anglais *Speaker Diarization*) vise à étiqueter les locuteurs dans un ou plusieurs documents audio, sans connaissance a priori des locuteurs. Appliquée à des collections, c'est une tâche globale qui consiste à traiter un ensemble de documents audio bruts (non segmentés en tours de parole) afin d'identifier de manière unique les tours de parole de chaque locuteur à l'échelle de la collection. Cette tâche se décompose généralement en deux étapes : la SRL intra-document, où il s'agit de segmenter et regrouper les occurrences des locuteurs au sein d'un même document, et le regroupement inter-document, qui vise à regrouper les locuteurs de chaque document [Meignier et al., 2002] à travers la collection complète (en autorisant, ou non, d'éventuels regroupements intra-document supplémentaires), avec parfois la possibilité de remettre en question la segmentation.

Pour la SRL de collection, cette approche en deux étapes est naturelle et la plus couramment utilisée. Elle permet de traiter de grands volumes de données comme des

archives audiovisuelles ou radiophoniques [Dupuy et al., 2014a; Tran et al., 2011a; Van Leeuwen, 2010; Yang et al., 2011a], des enregistrements téléphoniques [Ghaemmaghami et al., 2012; Karam and Campbell, 2013; Shum et al., 2013a, 2014a], ou encore d'enregistrements de réunions [Ferràs and Bourlard, 2012]. D'autres implémentations sont possibles, où tous les enregistrements peuvent être concaténés en un super-enregistrement, pour ensuite être traité comme un problème artificiel de SRL intra-document [Tran et al., 2011a].

Dans tous les cas, comme son nom l'indique, la SRL consiste en deux tâches distinctes : la segmentation, dont le but est d'isoler la voix des locuteurs et d'en détecter les changements, et le regroupement, dont le but est de positionner des labels identifiant les locuteurs, de manière qu'un label doit identifier un unique locuteur, que ce soit à l'échelle d'un document ou d'une collection de documents. Dans ce manuscrit, nous nous intéressons particulièrement à la *SRL de Collection* appliquée à l'indexation d'archives audiovisuelles. Les contenus audiovisuels se caractérisent par la grande variabilité de leur structure éditoriale, ce qui induit un nombre de locuteurs pouvant varier fortement, ainsi que la variété de l'environnement acoustique, dépendant du canal d'enregistrement (studio, téléphonique, dans la rue...). On peut aussi avoir de très courtes prises de parole (micro-trottoir) et de la parole superposée.

La SRL pourrait se résumer à la question suivante : « qui parle quand ? ». Traiter cette question suppose d'être capable de représenter la voix des locuteurs et de les comparer. Dans la section suivante, nous aborderons la question de la modélisation et comparaison de segments de parole, du point de vue locuteur. Ensuite, nous disposerons des outils nécessaires pour rappeler les approches classiques qui permettent de répondre à la problématique de la SRL : intra-document d'abord, puis inter-document. Enfin, nous étudierons la question des métriques d'évaluation de la SRL de collection.

3.2 Modélisation acoustique des segments-locuteurs

Dans cette section, nous abordons les différentes méthodes de modélisation acoustique de segments de parole, ainsi que les métriques permettant de les comparer du point de vue locuteur.

3.2.1 Représentation paramétrique de la parole

En général, la donnée d'entrée pour une tâche de reconnaissance automatique de la parole est le signal audio, la plupart du temps échantillonné à 8 ou 16 kHz, ce qui représente 8000 ou 16000 échantillons par seconde. Le signal de parole étant quasi-stationnaire sur de courtes durées (de l'ordre de la centaine de millisecondes)

[Rabiner and Juang, 1993], il est envisageable de caractériser le spectre sur de courtes durées, appelées trames, d'en général 20 millisecondes (avec recouvrement de 10 millisecondes), ce qui permet d'en déduire des statistiques dans le domaine fréquentiel.

C'est le rôle de la représentation cepstrale, très largement utilisée dans la communauté de la reconnaissance automatique de la parole. Le cepstre d'un signal est égal à la transformée de Fourier inverse du logarithme du spectre. La représentation la plus utilisée est la représentation *Mel-Frequency Cepstral Coefficients* (MFCC) [Mermelstein, 1976], où le spectre est préalablement compressé par un banc de filtres en échelle Mel [Stevens et al., 1937] et où la transformée de Fourier inverse est implémentée par une transformée en cosinus discrète (DCT). L'intérêt de la représentation cepstrale est de décorréler le processus de convolution (le logarithme permet de passer de convolution à addition), afin de facilement décomposer le modèle signal d'excitation/canal de résonance, qui est le modèle simplifié de la production de la parole.

Le nombre de coefficients choisis dépend de la résolution de la représentation souhaitée, généralement de l'ordre de la dizaine. Ainsi, si on considère un signal échantillonné à 8 kHz, la représentation paramétrique permet de réduire le nombre de paramètres d'environ un ordre de grandeur.

Dans l'hypothèse du passage du signal dans un canal bruité, la composante *bruit* devient, selon le modèle théorique, une composante additive, plus simple à représenter statistiquement. En effet, si le bruit est stationnaire, normaliser les coefficients cepstraux suffit à l'éliminer. On peut normaliser par soustraction de la moyenne (*Cepstral Mean Subtraction*) [Furui, 1981], éventuellement en réduisant également la variance (*Mean and Variance Normalization*).

Enfin, il est courant d'ajouter à ces coefficients cepstraux leurs dérivées premières (Δ) et secondes ($\Delta\Delta$), afin d'ajouter de l'information sur la dynamique du spectre [Furui, 1981]. Pour une trame donnée, on appelle le vecteur de paramètres vecteur acoustique.

3.2.2 Comparaison de locuteurs : la notion de rapport de vraisemblance

Dans les problématiques liées à la reconnaissance du locuteur, une notion clé est la notion de rapport de vraisemblance. Soient s_i et s_j deux réalisations d'une variable descriptive de locuteurs , quelle que soit la modélisation associée, la question de savoir si ces deux réalisations sont issues du même locuteur ou non revient à exprimer le rapport de vraisemblance entre les deux hypothèses H_{tar} « s_i et s_j représentent le même locuteur » et H_{non} « s_i et s_j représentent deux locuteurs distincts ».

$$\Lambda = \frac{L(s_i, s_j|H_{tar})}{L(s_i, s_j|H_{non})} \tag{3.1}$$

On peut également définir un seuil qui permet de prendre la décision. Parfois, on considère plutôt le logarithme du rapport de vraisemblance $log(\Lambda)$, qui permet d'exprimer le rapport sous forme additive, tout en préservant les relations d'ordre.

3.2.3 Modèle mono-gaussien

Pour modéliser des segments de parole, le modèle le plus simple est le modèle mono-gaussien. Etant donné un segment de parole duquel on a extrait des vecteurs de coefficients MFCC, on peut représenter ce segment par un modèle gaussien $\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, de moyenne $\boldsymbol{\mu}$ et de covariance $\boldsymbol{\Sigma}$. Lorsqu'on considère deux segments de parole i et j et qu'on souhaite estimer s'ils ont été prononcés par le même locuteur ou par deux locuteurs distincts, on peut utiliser différences mesures, telles que la mesure GLR (*Generalized Likelihood Ratio*) [Gish et al., 1991] ou BIC (*Bayesian Information Criterion*) [Schwarz et al., 1978], détaillées ci-après. On note également l'existence d'autres mesures comme la divergence de Kullback-Leibler symétrique [Delacourt et al., 1999; Siegler et al., 1997], mesure de distance entre distributions gaussiennes, et la divergence gaussienne [Barras et al., 2006].

3.2.3.1 Rapports de vraisemblance

Mesure GLR Soient les paramètres acoustiques $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$ de chaque segment, modélisés par deux gaussiennes $\Theta_i(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$ et $\Theta_j(\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})$. Le calcul du rapport de vraisemblance généralisé (*Generalized Likelihood Ratio*, ou GLR) [Gish et al., 1991] consiste à exprimer le rapport de vraisemblance entre les deux hypothèses : H_{tar} « $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$ correspondent à un même locuteur » et H_{non} « $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$ correspondent à deux locuteurs différents ». Dans le premier cas, la meilleure représentation du locuteur serait un modèle gaussien $\Theta_{i,j}(\boldsymbol{\mu_{i,j}}, \boldsymbol{\Sigma_{i,j}})$ estimé sur $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$, tandis que dans le deuxième cas, la représentation par les deux modèles distincts Θ_i et Θ_j serait plus adaptée. On cherche dont à calculer :

$$GLR(\boldsymbol{x_i}, \boldsymbol{x_j}) = \frac{L(\boldsymbol{x_i}, \boldsymbol{x_j}|\Theta_{i,j}(\boldsymbol{\mu_{i,j}}, \boldsymbol{\Sigma_{i,j}}))}{L(\boldsymbol{x_i}|\Theta_i(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}))L(\boldsymbol{x_j}|\Theta_j(\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}))} \tag{3.2}$$

Cette mesure GLR permet de définir une distance entre les séquences $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$, $d(\boldsymbol{x_i}, \boldsymbol{x_j}) = -log(GLR(\boldsymbol{x_i}, \boldsymbol{x_j}))$, qui, si elle dépasse un seuil prédéfini, permet de décider que les locuteurs i et j sont différents.

Mesure BIC La mesure basée sur le critère d'information bayésien (*Bayesian Information Criterion*, ou BIC) [Schwarz et al., 1978] ressemble à la mesure GLR,

mais avec l'ajout d'un facteur de pénalité. Soit un segment $\boldsymbol{x}$ constitué de n trames, représenté par le modèle $\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, on définit :

$$BIC(\Theta) = log(L(\boldsymbol{x}|\Theta)) - \frac{\lambda}{2}\#(\Theta)log(n) \tag{3.3}$$

λ est un paramètre de pénalité à choisir, dépendant des données et $\#(\Theta)$ est la complexité du modèle Θ. Pour déterminer si deux segments $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$ correspondent au même locuteur ou non, on peut définir la mesure ΔBIC.

$$\Delta BIC(\boldsymbol{x_i}, \boldsymbol{x_j}) = BIC(\Theta_i) + BIC(\Theta_j) - BIC(\Theta_{i,j}) = R(i,j) - \lambda P \tag{3.4}$$

On peut noter que $R(i,j) = -log(GLR(\boldsymbol{x_i}, \boldsymbol{x_j}))$, qui se calcule, dans le cas du modèle mono-gaussien, de la façon suivante :

$$R(i,j) = \frac{n_i + n_j}{2}log(det(\Sigma_{i,j})) - \frac{n_i}{2}log(det(\Sigma_i)) - \frac{n_j}{2}log(det(\Sigma_j)) \tag{3.5}$$

et

$$P = \frac{\left(p + \frac{p(p+1)}{2}\right)}{2} * log(n_i + n_j) \tag{3.6}$$

p étant la dimension des vecteurs acoustiques. D'une façon similaire à la mesure GLR, l'approche ΔBIC permet de prendre une décision sur la similarité locuteur de deux segments. En réalité, si la durée des segments à comparer est strictement la même, on se ramène au calcul de la mesure GLR.

3.2.4 Modèle à mélange de gaussiennes

L'approche consistant à modéliser les locuteurs à l'aide de modèles statistiques basés sur des mélanges de gaussiennes date des années 1990. L'utilisation du modèle à mélanges de gaussiennes (*Gaussian Mixture Model*, ou GMM) repose sur l'idée que la distribution qui caractérise l'ensemble des vecteurs acoustiques d'un locuteur donné est une combinaison linéaire de plusieurs gaussiennes [Reynolds, 1992; Rose and Reynolds, 1990]. Soit $\mathbf{X} = (\mathbf{x_i})$, $i \in 1..N$, l'ensemble des vecteurs acoustiques et $\Theta = (\lambda_c, \mu_\mathbf{c}, \mathbf{\Sigma_c}), c \in 1..C$, le modèle GMM à C composantes et covariance diagonale, sa densité de probabilité s'écrit :

$$p(\mathbf{x_i}|\Theta) = \sum_{c=1}^{C} \lambda_c p_c(\mathbf{x_i}|\mu_\mathbf{c}, \mathbf{\Sigma_c}) \tag{3.7}$$

avec $\sum_{c=1}^{C} \lambda_c = 1$ et $(\mu_\mathbf{c}, \mathbf{\Sigma_c})$ les paramètres de la gaussienne c.

Il existe deux façons d'estimer Θ sur des données d'apprentissage : par maximum de vraisemblance (*Maximum Likelihood*, ou ML) ou par adaptation maximum *a posteriori* (MAP).

3.2.4.1 Approche par maximum de vraisemblance

L'algorithme *Expectation-Maximization* (EM) [Dempster et al., 1977] est un algorithme itératif visant à maximiser la vraisemblance des données $\mathbf{X} = (\mathbf{x_i})$ selon un modèle Θ, à l'aide de variables latentes $\mathbf{H} = (\mathbf{h_i})$ caractérisant les données. Dans notre problème, les variables latentes décrivent la gaussienne $c \in 1..C$ d'appartenance de chaque $\mathbf{x_i}$, ie. $p(\mathbf{h_i} = k) = \lambda_k$.

$L(\Theta, \mathbf{X}, \mathbf{H}) = p(\mathbf{X}, \mathbf{H}|\Theta)$ représente la probabilité qu'on cherche à maximiser.

Il comprend deux étapes :

- L'étape *Expectation* consiste à calculer la probabilité à posteriori d'appartenance de tout vecteur acoustique $\mathbf{x_i}$ à chaque gaussienne.

$$p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta) = \frac{p(\mathbf{x_i}|\mathbf{h_i} = k, \Theta)\lambda_k}{\sum_{c=1}^{C} p(\mathbf{x_i}|\mathbf{x_i} = c, \Theta)\lambda_c}, \forall i \tag{3.8}$$

- L'étape *Maximization* consiste à mettre à jour le modèle Θ à l'aide de la probabilité précédemment calculée.

$$\lambda_{k_{new}} = \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta) \tag{3.9}$$

$$\mu_{\mathbf{k_{new}}} = \lambda_{k_{new}} \frac{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta)\mathbf{x_i}}{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta)} \tag{3.10}$$

$$\mathbf{\Sigma_{k_{new}}} = \lambda_{k_{new}} \frac{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta)\mathbf{x_i}\mathbf{x_i}^T}{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta)} - \mu_{\mathbf{k_{new}}}\mu_{\mathbf{k_{new}}}^T \tag{3.11}$$

Les itérations sont répétées jusqu'à ce que la convergence soit atteinte, c'est-à-dire jusqu'à ce que l'augmentation de la vraisemblance passe sous un seuil prédéfini, ou que le nombre d'itérations choisi soit atteint. Comme l'algorithme converge vers un minimum local, la qualité de modélisation dépend beaucoup de l'initialisation des paramètres et de la quantité de données utilisées pour l'apprentissage. Souvent, pour estimer un modèle à 2^k gaussiennes, on procède par divisions sucessives. On commence par estimer une seule gaussienne sur l'ensemble des données, puis on multiplie le nombre de gaussiennes par deux en déviant chaque moyenne aléatoirement. On ajuste les nouvelles gaussiennes par l'algorithme EM. On répète le processus jusqu'à atteindre le nombre souhaité de gaussiennes.

3.2.4.2 Approche par adaptation MAP

Quand la quantité de données d'apprentissage par locuteur est trop faible, l'estimation GMM via EM est difficile. L'idée de l'adaptation MAP [Gauvain and Lee, 1994; Reynolds et al., 2000] est, pour chaque modèle de locuteur, de dériver d'un modèle initial, qu'on appelle modèle du monde (*Universal Background Model*, ou UBM) [Carey and Parris, 1992]. L'idée du GMM/UBM est de modéliser la parole, indépendamment du locuteur. Pour ce faire, l'estimation par EM d'un GMM est faite sur une quantité de données importante, contenant un grand nombre de locuteurs différents.

Une fois le GMM/UBM obtenu $\Theta_{ubm} = (\lambda_k, \mu_{\mathbf{k}}, \mathbf{\Sigma_k})$, pour chaque locuteur, on cherche à adapter le modèle du monde pour estimer celui du locuteur, en adaptant les moyennes. L'algorithme est très semblable à l'approche par maximum de vraisemblance, à ceci près qu'on remplace Θ par Θ_{ubm} à l'étape E et qu'on introduit un facteur de pondération α_k dans la mise à jour des paramètres, tel que :

$$\mu_{\mathbf{k_{new}}} = \alpha_k \lambda_{k_{new}} \frac{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta_{ubm})\mathbf{x_i}}{\sum_{i=1}^{N} p(\mathbf{h_i} = k|\mathbf{x_i}, \Theta_{ubm})} + (1 - \alpha_k)\mu_{\mathbf{k_{ubm}}} \tag{3.12}$$

avec,

$$\alpha_k = \frac{\sum_{i=1}^{N} p(\mathbf{z_i} = k|\mathbf{x_i}, \Theta)}{\sum_{i=1}^{N} p(\mathbf{z_i} = k|\mathbf{x_i}, \Theta) + \gamma} \tag{3.13}$$

γ est un facteur permettant d'équilibrer l'importance des paramètres du modèle du monde par rapport aux paramètres spécifiques du locuteur.

3.2.4.3 Rapports de vraisemblance

Entropie Croisée Dans le cadre de modèles GMM, on cherche à comparer deux modèles Θ_i et Θ_j de locuteurs i et j pour lesquels on dispose d'ensembles de variables descriptives (s_i) et (s_j). Une première mesure de similarité entre locuteurs est l'entropie croisée (CE). Elle a d'abord été proposée par [Reynolds, 1995], puis reprise par [Solomonoff et al., 1998] qui l'a appelée entropie croisée. On retrouve chez [Le et al., 2007] la même méthode mais nommée différemment (*Normalized Cross Likelihood Ratio* ou NCLR).

$$CE(\Theta_i, \Theta_j) = log(\frac{p(s_i|\Theta_i)}{p(s_i|\Theta_j))}) + log(\frac{p(s_j|\Theta_j)}{p(s_j|\Theta_i)}) \tag{3.14}$$

Rapport de vraisemblance croisé Inspiré de la méthode précédente, mais pensé pour le cas où les deux modèles ont été entraînés par adaptation d'un modèle du monde Θ_{ubm}, le rapport de vraisemblance croisé (*Cross Likelihood Ratio* ou CLR) a été proposé par [Reynolds et al., 1998] avant d'être repris par [Barras et al., 2006]. Il se définit de la façon suivante.

$$CLR(\Theta_i, \Theta_j) = \frac{1}{N_i} log(\frac{p(s_i|\Theta_j)}{p(s_i|\Theta_{ubm})}) + \frac{1}{N_j} log(\frac{p(s_j|\Theta_i)}{p(s_j|\Theta_{ubm})}) \tag{3.15}$$

3.2.4.4 La notion de supervecteur GMM

Après l'adaptation MAP effectuée pour chaque locuteur s du corpus d'apprentissage, on peut définir $\mathbf{u} = [\mu_\mathbf{1}, \mu_\mathbf{2}, \ldots, \mu_\mathbf{N}]$, un supervecteur représentant le locuteur en question [Ben et al., 2004]. Le supervecteur permet de représenter un segment de parole par un vecteur de taille fixe, quelle que soit la taille du segment. La représentation a initialement été utilisée avec un classifieur de type séparateur à vaste marge (SVM) [Campbell et al., 2006].

La représentation en supervecteur a ouvert la porte à des méthodes de décomposition en facteur et de réduction de dimensionalité, [Burget et al., 2007; Kenny, 2010; Kuhn et al., 1998; Matrouf et al., 2007], qui ont mené à la modélisation *Joint Factor Analysis* (JFA) [Kenny et al., 2007] et *i-vector* [Dehak et al., 2011].

3.2.5 Le paradigme *i-vector*

Généralement, le nombre de gaussiennes utilisées dans la modélisation GMM est de l'ordre de la centaine voire du millier, ce qui amène à des supervecteurs GMM de l'ordre de la dizaine de milliers de paramètres pour représenter les locuteurs (le nombre de gaussiennes multiplié par la dimension des vecteurs acoustiques). L'idée des méthodes de décomposition en facteurs est d'estimer une représentation dans un espace discriminant pour une tâche donnée. En reconnaissance du locuteur, la décomposition en facteurs (*Joint Factor Analysis*, ou JFA) [Kenny et al., 2007] consiste à exprimer le supervecteur locuteur comme la somme de trois composantes indépendantes : la composante liée au locuteur, la composante liée au canal de transmission et une composante résiduelle. Plus récente, la modélisation *i-vector* [Dehak et al., 2011] consiste à projeter le supervecteur dans un espace dit de variabilité totale, pour obtenir une représentation compacte des segments acoustiques. Selon ce paradigme, un supervecteur GMM moyen m_s, représentant une session-locuteur s, peut s'écrire comme une observation du modèle génératif suivant :

$$\boldsymbol{u_s} = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y_s} \tag{3.16}$$

Dans cette équation, $\boldsymbol{y_s}$ est une variable aléatoire dont l'estimation MAP est le *i-vector* $\boldsymbol{\phi_s}$. Il permet une représentation du locuteur de faible dimension, et surtout d'une taille fixe indépendante de la durée de parole ayant servi à le calculer. L'estimation de la matrice de variabilité totale $\boldsymbol{V}$ (*Total Variability*, ou TV) se fait via l'algorithme EM.

3.2.5.1 Estimation de la matrice de Variabilité Totale

La première étape consiste, pour chaque composante du GMM/UBM c et chaque session-locuteur s d'un corpus d'apprentissage, à calculer les statistiques, $\boldsymbol{\gamma_t}(c)$ étant la probabilité *a posteriori* de la composante c pour une observation t d'un locuteur s, et $\mathbf{Y_t}$ le vecteur de coefficients acoustiques à la trame t. $\boldsymbol{\mu_c}$ est la moyenne de la composante c du GMM/UBM. Remarquons que si l'on dispose de plusieurs sessions d'un même locuteur, pour l'apprentissage, elles sont considérées comme des sessions de locuteurs différents.

$$\boldsymbol{N_c}(s) = \sum_{t \in s} \boldsymbol{\gamma_t}(c) \tag{3.17}$$

$$\boldsymbol{F_c}(s) = \sum_{t \in s} \boldsymbol{\gamma_t}(c) \boldsymbol{Y_t} \tag{3.18}$$

$$\boldsymbol{S_c}(s) = diag(\sum_{t \in s} \boldsymbol{\gamma_t}(c) \boldsymbol{Y_t} \boldsymbol{Y_t}^T) \tag{3.19}$$

$$\tilde{\boldsymbol{F}}_{\boldsymbol{c}}(s) = \boldsymbol{F_c}(s) - \boldsymbol{N_c}(s) \boldsymbol{\mu_c} \tag{3.20}$$

$$\tilde{\boldsymbol{S}}_{\boldsymbol{c}}(s) = \boldsymbol{S_c}(s) - diag(\boldsymbol{F_c}(s) \boldsymbol{\mu_c}^T + \boldsymbol{\mu_c} \boldsymbol{F_c}(s)^T - \boldsymbol{N_c}(s) \boldsymbol{\mu_c} \boldsymbol{\mu_c}^T) \tag{3.21}$$

On définit $\boldsymbol{NN}(s)$, matrice diagonale par blocs contenant les $\boldsymbol{N_c}(s)$, $\boldsymbol{FF(s)}$, matrice colonne par blocs contenant les $\tilde{\boldsymbol{F}}_{\boldsymbol{c}}(s)$, et $\boldsymbol{SS}(s)$, matrice diagonale par blocs contenant les $\tilde{S}_c(s)$, pour $c \in 1..C$. A partir de $\boldsymbol{V}$, initialisée aléatoirement, on définit :

$$\boldsymbol{l_v}(s) = \boldsymbol{I} + \boldsymbol{V}^T \boldsymbol{\Sigma^{-1}} \boldsymbol{NN}(s) \boldsymbol{V} \tag{3.22}$$

La distribution de la variable aléatoire $y(s)$ est donc la suivante.

$$\boldsymbol{y}(s) \sim \mathcal{N}(\boldsymbol{l_v}^{-1}(s) \boldsymbol{V}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{FF}(s), \boldsymbol{l_v}^{-1}(s)) \tag{3.23}$$

Et on peut estimer le *i-vector*.

$$\bar{\boldsymbol{y}}(s) = E[\boldsymbol{y}(s)] = \boldsymbol{l_v}^{-1}(s) \boldsymbol{V}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{FF}(s) \tag{3.24}$$

En cumulant des statistiques supplémentaires, on peut mettre à jour $\boldsymbol{V}$.

$$\boldsymbol{N_c} = \sum_s \boldsymbol{N_c}(s) \tag{3.25}$$

$$\boldsymbol{A_c} = \sum_s \boldsymbol{N_c}(s) \boldsymbol{l_v}^{-1}(s) \tag{3.26}$$

$$\boldsymbol{C_c} = \sum_s \boldsymbol{FF}(s) E[\boldsymbol{y}(s)]^T \tag{3.27}$$

$$\boldsymbol{NN} = \sum_s \boldsymbol{NN}(s) \tag{3.28}$$

Avec $\boldsymbol{A}$ et $\boldsymbol{C}$ matrices colonnes par bloc contenant respectivement les $\boldsymbol{A_c}$ et $\boldsymbol{C_c}$, la mise à jour de $\boldsymbol{V}$ et $\boldsymbol{\Sigma}$ se font selon l'équation suivante :

$$\boldsymbol{V} = \boldsymbol{A}^{-1}\boldsymbol{C} \tag{3.29}$$

$$\boldsymbol{\Sigma} = \boldsymbol{NN}^{-1}((\sum_s \boldsymbol{SS}(s)) - diag(\boldsymbol{CV}^T)) \tag{3.30}$$

Une fois $\boldsymbol{V}$ et $\boldsymbol{\Sigma}$ mises à jour, on peut répéter les étapes 3.22 à 3.30, jusqu'à satisfaire un critère de convergence ou un nombre d'itérations fixé.

3.2.5.2 Estimation des *i-vectors*

Une fois les matrices $\boldsymbol{V}$ et $\boldsymbol{\Sigma}$ estimées, pour une session s, on peut extraire le *i-vector* de la façon suivante :

$$\boldsymbol{\phi_s} = \bar{\boldsymbol{y}}(s) = E[\boldsymbol{y}(s)] = \boldsymbol{l_v}^{-1}(s)\boldsymbol{V}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{FF}(s) \tag{3.31}$$

3.2.5.3 Similarité cosine et compensation WCCN

Lorsqu'il s'agit de comparer des *i-vectors*, la similarité cosine est une méthode simple et dont l'efficacité est prouvée [Dehak et al., 2011]. Lorsque le score de similarité est proche de 1, les deux *i-vectors* sont très semblables, donc susceptibles de représenter le même locuteur, tandis que lorsqu'il est proche de -1, les *i-vectors* sont susceptibles de représenter des locuteurs différents.

$$s(\boldsymbol{\phi_1}, \boldsymbol{\phi_2}) = \frac{\boldsymbol{\phi_1}\boldsymbol{\phi_2}}{\|\boldsymbol{\phi_1}\|\|\boldsymbol{\phi_2}\|} \in [-1; 1] \tag{3.32}$$

Celle-ci peut se combiner avec la normalisation de la covariance intra-classe (Within Class Covariance Normalization ou WCCN [Dehak et al., 2011]). Considérant les n_i observations d'un locuteur i, de moyenne h_i, la matrice WCCN se calcule de la façon suivante :

$$\boldsymbol{W} = \frac{1}{S}\sum_{i=1}^{S}\frac{1}{n_i}\sum_{j=1}^{n_i}(\boldsymbol{\phi_{ij}} - \boldsymbol{h_i})(\boldsymbol{\phi_{ij}} - \boldsymbol{h_i})^T \tag{3.33}$$

Une fois la matrice de variabilité intra classe estimée, celle-ci peut être compensée par rotation des *i-vectors* selon la formule suivante :

$$\hat{\phi}_{ij} = \boldsymbol{L}\phi_{ij} \tag{3.34}$$

$\boldsymbol{L}$ est la décomposition de Cholesky de $\boldsymbol{W^{-1}}$, $\boldsymbol{W^{-1}} = \boldsymbol{LL^T}$.

3.2.6 Modélisation de la variabilité inter- et intra-locuteur : l'approche PLDA

D'abord pensée pour la reconnaissance de visages [Prince and Elder, 2007], la modélisation PLDA a par la suite été adaptée à la reconnaissance de locuteurs [Kenny, 2010]. Dans cette modélisation, les *i-vectors* sont considérés comme des observations d'un modèle génératif probabiliste. Chaque *i-vector* $\boldsymbol{\phi_{ij}}$, de dimension p, peut se décomposer de la manière suivante :

$$\boldsymbol{\phi_{ij}} = \boldsymbol{\mu} + \boldsymbol{\Phi h_i} + \boldsymbol{\Psi z} + \boldsymbol{\epsilon} \tag{3.35}$$

La partie $\boldsymbol{\mu} + \boldsymbol{\Phi h_i}$ ne dépend que du locuteur et la partie $\boldsymbol{\Psi z} + \boldsymbol{\epsilon}$ représente la variabilité canal (intra-locuteur). La matrice $\boldsymbol{\Phi}$ (les *eigenvoices* [Kuhn et al., 2000]) sert de base pour le sous-espace locuteur, alors que $\boldsymbol{\Psi}$ (les *eigenchannels*) sert de base pour le sous-espace canal. $\boldsymbol{h_i}$ et $\boldsymbol{z}$ suivent des lois normales standard, et $\boldsymbol{\epsilon}$ suit une loi de type $\mathcal{N}(0, \boldsymbol{\Lambda})$, $\boldsymbol{\Lambda}$ étant diagonale. Souvent, on ignore le facteur canal en l'intégrant dans $\boldsymbol{\epsilon}$, le modèle devient :

$$\boldsymbol{\phi_{ij}} = \boldsymbol{\mu} + \boldsymbol{\Phi h_i} + \boldsymbol{\epsilon} \tag{3.36}$$

$\boldsymbol{\Phi}$ est de taille $p \times r (r < p)$, r étant le nombre d'*eigenvoices* et $\boldsymbol{\epsilon}$ est de dimension p. Comme $\boldsymbol{h_i} \sim \mathcal{N}(0, \boldsymbol{I})$, on connaît la distribution des *i-vectors* : $\boldsymbol{\phi_{ij}} \sim \mathcal{N}(\mu, \boldsymbol{\Gamma} + \boldsymbol{\Lambda})$, avec $\boldsymbol{\Gamma} = \boldsymbol{\Phi\Phi^T}$. $\boldsymbol{\Gamma}$ est la matrice de variabilité inter-classe et $\boldsymbol{\Lambda}$ la matrice de variabilité intra-classe.

L'estimation des paramètres $\boldsymbol{\Phi}$ et $\boldsymbol{\Lambda}$ s'effectue avec l'algorithme EM, à partir d'un jeu d'apprentissage conséquent, annoté en locuteurs. L'entraînement s'effectue sur la base des locuteurs récurrents du corpus d'apprentissage. Par la suite, nous considérerons que les *i-vectors* sont centrés (i.e., $\boldsymbol{\mu} = 0$). L'apprentissage d'un modèle PLDA $\Theta = (\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ consiste à estimer les paramètres maximisant la vraisemblance :

$$L(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \frac{1}{N} \sum_{i=1}^{S} \sum_{j=1}^{n_i} log(p(\boldsymbol{\phi_{ij}} | \boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda})) \tag{3.37}$$

avec

$$p((\boldsymbol{\phi_{ij}}) | \boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \mathcal{N}((\boldsymbol{\phi_{ij}}); 0, \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T + \tilde{\boldsymbol{\Lambda}}) \tag{3.38}$$

Où $\tilde{\boldsymbol{\Phi}}$ est une matrice colonne par blocs, contenant N fois $\boldsymbol{\Phi}$ et $\tilde{\boldsymbol{\Lambda}}$ est une matrice diagonale par blocs contenant N fois $\boldsymbol{\Lambda}$, N étant le nombre total de *i-vectors*.

Estimation des paramètres PLDA Les étapes de l'algorithme EM sont les suivantes :

- étape E : estimation des probabilités *a posteriori* des variables locuteurs cachées $\boldsymbol{h_i}$, a partir des observations $\{\boldsymbol{\phi_{ij}}\}_{j=1}^{n_i}$ de ces locuteurs.

$$E[\boldsymbol{h_i}] = (N_i \boldsymbol{\Phi}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\Phi} + \boldsymbol{I})^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Lambda}^{-1} \sum_{j=1}^{n_i} \boldsymbol{\phi_{ij}} \tag{3.39}$$

$$E[\boldsymbol{h_i} \boldsymbol{h_i}^T] = (N_i \boldsymbol{\Phi}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\Phi} + \boldsymbol{I})^{-1} + E[\boldsymbol{h_i}] E[\boldsymbol{h_i}]^T \tag{3.40}$$

- étape M : a partir des estimations précédents, mise à jour du modèle.

$$\boldsymbol{\Phi_{new}} = \left(\sum_{i=1}^{S} \sum_{j=1}^{n_i} \boldsymbol{\phi_{ij}} E[\boldsymbol{h_i}]^T \right) \left(\sum_{i=1}^{S} N_i E[\boldsymbol{h_i} \boldsymbol{h_i}^T] \right)^{-1} \tag{3.41}$$

$$\boldsymbol{\Lambda_{new}} = \frac{1}{N} \sum_{i=1}^{S} \sum_{j=1}^{n_i} \left[\boldsymbol{\phi_{ij}} \boldsymbol{\phi_{ij}}^T - \boldsymbol{\Phi_{new}} E[\boldsymbol{h_i}] \boldsymbol{\phi_{ij}}^T \right] \tag{3.42}$$

Expression du rapport de vraisemblance Soient ϕ_i et ϕ_j deux *i-vectors*, pour lesquels on cherche à savoir s'ils représentent le même locuteur. Le score entre les deux *i-vectors* est le log-rapport de vraisemblance entre les hypothèses H_{tar} « ϕ_i et ϕ_j représentent le même locuteur » et H_{non} « ϕ_i et ϕ_j représentent des locuteurs différents ».

$$score(\boldsymbol{\phi_i}, \boldsymbol{\phi_j}) = log \frac{P(\boldsymbol{\phi_i}, \boldsymbol{\phi_j} | H_{tar})}{P(\boldsymbol{\phi_i}, \boldsymbol{\phi_j} | H_{non})} \tag{3.43}$$

Il s'agit d'évaluer si les deux vecteurs ont été généré à partir du même $\boldsymbol{h_i}$ ou non, et de résoudre les équations découlant des deux hypothèses. Sous l'hypothèse H_{tar}, le modèle génératif nous permet d'écrire :

$$\begin{bmatrix} \boldsymbol{\phi_i} \\ \boldsymbol{\phi_j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi} \end{bmatrix} \boldsymbol{h_{ij}} + \begin{bmatrix} \boldsymbol{\epsilon_i} \\ \boldsymbol{\epsilon_j} \end{bmatrix} \tag{3.44}$$

Tandis que sous l'hypothèse H_{non}, on a :

$$\begin{bmatrix} \boldsymbol{\phi_i} \\ \boldsymbol{\phi_j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & 0 \\ 0 & \boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{h_i} \\ \boldsymbol{h_j} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon_i} \\ \boldsymbol{\epsilon_j} \end{bmatrix} \tag{3.45}$$

D'où l'expression des probabilités conditionnelles :

$$P(\boldsymbol{\phi_i}, \boldsymbol{\phi_j} | H_{tar}) = \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\phi_i} \\ \boldsymbol{\phi_j} \end{bmatrix} | \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Lambda} & \boldsymbol{\Phi}\boldsymbol{\Phi}^T \\ \boldsymbol{\Phi}\boldsymbol{\Phi}^T & \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Lambda} \end{bmatrix} \right) \tag{3.46}$$

et

$$P(\boldsymbol{\phi_i}, \boldsymbol{\phi_j}|H_{non}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\phi_i}\\ \boldsymbol{\phi_j}\end{bmatrix} | \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda} & 0\\ 0 & \boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda}\end{bmatrix}\right) \tag{3.47}$$

Ce qui nous donne l'expression du log-rapport de vraisemblance :

$$score(\boldsymbol{\phi_i}, \boldsymbol{\phi_j}) = -\begin{bmatrix}\boldsymbol{\phi_i}\\ \boldsymbol{\phi_j}\end{bmatrix}^T (\boldsymbol{N_{tar}^{-1}} - \boldsymbol{N_{non}^{-1}}) \begin{bmatrix}\boldsymbol{\phi_i}\\ \boldsymbol{\phi_j}\end{bmatrix} \tag{3.48}$$

avec

$$\boldsymbol{N_{tar}} = \begin{bmatrix}\boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda} & \boldsymbol{\Phi\Phi}^T\\ \boldsymbol{\Phi\Phi}^T & \boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda}\end{bmatrix} \tag{3.49}$$

$$\boldsymbol{N_{non}} = \begin{bmatrix}\boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda} & 0\\ 0 & \boldsymbol{\Phi\Phi}^T + \boldsymbol{\Lambda}\end{bmatrix} \tag{3.50}$$

Normalisation Avant apprentissage ou calcul des scores PLDA, il est courant de normaliser les *i-vectors*. On peut distinguer la normalisation par la norme euclidienne [Garcia-Romero and Espy-Wilson, 2011], qui consiste à faire en sorte que tous les *i-vectors* sont répartis sur une sphère unitaire. Il est également possible de d'abord les centrer et normaliser leur variance, par les méthodes EFR (*Eigen Factor Radial*) ou SNN (*Spherical Nuisance Normalisation*) [Bousquet et al., 2012, 2011], des algorithmes itératifs.

Normalisation EFR L'approche EFR consiste à réduire la covariance totale des *i-vectors*. A partir d'un ensemble d'*i-vectors* d'apprentissage $\mathcal{T}$, le processus de normalisation est le suivant, n étant le nombre d'itérations.

Algorithme 1 : Algorithme de normalisation EFR

for i=1 **to** n **do**
- 1 : Calculer (μ_i, Σ_i) sur $\mathcal{T}$
- 2 : Mettre à jour les *i-vectors* :
- **for each** ϕ **in** $\mathcal{T}$ **do**
  - $\phi \leftarrow \frac{\Sigma_i^{-\frac{1}{2}}(\phi - \mu_i)}{||\Sigma_i^{-\frac{1}{2}}(\phi - \mu_i)||}$
- **end**

end

Lorsqu'on veut normaliser des *i-vectors* d'évaluation, on applique alors le même processus, avec les mêmes moyennes μ_i et matrices de covariance Σ_i estimées sur le corpus d'apprentissage.

Normalisation SNN La normalisation SNN suit le même principe, à l'exception du fait qu'on ne normalise plus selon la covariance totale, mais selon la covariance intra-classe. Dans l'ensemble de nos expériences, nous utiliserons la normalisation SNN avec la modélisation PLDA.

3.3 SRL intra-document

Dans cette section, nous abordons les outils généralement utilisés pour la tâche de SRL intra-document. Considérons un document audio dont nous souhaitons segmenter et regrouper les locuteurs. La sortie attendue du traitement est une liste de segments *[début, fin]* identifiés par des labels. Chaque label doit correspondre à un locuteur supposé du document. Comme le nombre de locuteurs et la variabilité canal sont en général limités au sein d'un même document, on privilégie l'utilisation de modèles à faible complexité. La SRL intra-document est en général réalisée en trois étapes : prétraitement, segmentation et regroupement (e.g. [Anguera et al., 2012; Bonastre et al., 2000; Tranter and Reynolds, 2006]).

3.3.1 Prétraitement

Le prétraitement classique consiste à extraire les paramètres acoustiques, avant d'effectuer une détection de parole pour éliminer bruits parasites et silences. Les paramètres acoustiques les plus utilisés sont les coefficients cepstraux (MFCC, voir section 3.2.1), agrémentés ou non de leurs Δ, $\Delta\Delta$ et de l'énergie, en fonction du traitement. Il s'agit de générer une succession de vecteurs décrivant l'acoustique, calculés sur des trames successives de durée fixe, avec recouvrement. La détection de parole s'effectue le plus souvent à l'aide de modèles à au moins deux Gaussiennes (une pour la parole, une pour le silence/bruit), et utilise l'algorithme de Viterbi pour le décodage. A la fin du prétraitement, les zones de parole sont bien identifiées, l'étape suivante consiste à détecter les changements de locuteur.

3.3.2 Segmentation en locuteurs

La segmentation en locuteurs vise à détecter les frontières des segments de parole, où le locuteur change. Le principe général est le suivant : comparer des segments consécutifs et décider s'ils ont été prononcés par le même locuteur, ou non. Plus un segment de parole est long, plus la technique qui permet de comparer deux segments peut être complexe, mais plus la probabilité qu'il soit homogène du point de vue locuteur est faible. Le résultat de la segmentation en locuteurs est donc un compromis entre longueur des segments et homogénéité.

Pour ce faire, on commence donc par comparer des segments courts, de durée fixe, qu'on regroupe hiérarchiquement à l'aide de métriques de plus en plus complexes, en

interdisant les regroupements de segments non consécutifs. L'approche se fait donc en plusieurs passes : une passe par type de modèle/méthode de comparaison.

L'approche de plus bas niveau consiste à comparer les segments courts, de durée fixe, à l'aide de deux fenêtres glissantes parcourant le document audio. Pour la comparaison, on peut utiliser la mesure GLR [Gish et al., 1991], la divergence de Kullback-Leibler [Delacourt et al., 1999; Siegler et al., 1997] ou encore la Divergence Gaussienne [Barras et al., 2006]. Cette étape permet d'obtenir un premier jeu de segments plus longs, de durées variables.

Pour comparer des segments de parole relativement courts mais de durées variables, nous avons vu à la section 3.2.3.1 que la mesure BIC [Schwarz et al., 1978] était une bonne candidate. Cette méthode nécessite de choisir empiriquement un seuil pour n'obtenir que des segments homogènes du point de vue locuteur. En effet, pour passer à des méthodes de représentation des segments-locuteurs plus complexes, telles que le GMM ou le *i-vector*, on a besoin de segments homogènes et suffisamment longs. C'est pour cela que la segmentation BIC est souvent un préalable à des méthodes plus complexes.

3.3.3 Regroupement intra-document

Une fois la segmentation en locuteurs effectuée, le document audio est découpé en segments homogènes correspondant à différents locuteurs, mais ne pouvant pas dépasser un tour de parole. La tâche de regroupement consiste à regrouper (lier) les segments par locuteur, à l'échelle du document audio. Plusieurs approches sont possibles, et peuvent être utilisées de façon complémentaire. L'approche commune consiste à calculer une matrice de similarité entre toutes les paires de segments, à l'échelle du document. Les similarités dépendent de la représentation utilisée : Gaussienne/BIC (e.g. [Dupuy et al., 2012a]), GMM/CLR (e.g. [Barras et al., 2006; Ghaemmaghami et al., 2013]), ou *i-vector*/cosine ou PLDA (e.g. [Dupuy et al., 2012a]). En général, l'approche Gaussienne/BIC est un prérequis aux deux approches suivantes, car pour estimer de manière précise un modèle GMM ou *i-vector*, la quantité de parole sous-jacente doit être supérieure à un tour de parole.

A partir de la matrice de similarités, des approches classiques de regroupement non supervisé sont envisageables, le plus utilisé étant le regroupement hiérarchique ascendant [Chen and Gopalakrishnan, 1998; Gish et al., 1991; Siegler et al., 1997; Siu et al., 1992; Solomonoff et al., 1998; **?**], mais on trouve dans la littérature d'autres méthodes de regroupement comme le k-moyennes (*k-means*) [Shum et al., 2011a], qui impose de spécifier le nombre de classes souhaité, les regroupements basés sur des représentations sous forme de graphes [Shum et al., 2013a] ou encore le regroupement ILP [Rouvier and Meignier, 2012] (pour *Integer Linear Programming*).

Après le regroupement BIC, les frontières des segments sont parfois affinées par l'algorithme de Viterbi (e.g. [Dupuy et al., 2012a; Meignier et al., 2000; Shum et al.,

2013b]) : des GMMs à quelques gaussiennes sont appris de manière itérative pour chaque classe-locuteur, et le décodage de Viterbi ajuste les frontières jusqu'à répondre à un critère de convergence. Notons que les méthodes de regroupement précitées travaillent avec la notion de distance, c'est-à-dire une métrique analogue à l'opposée d'une similarité.

Dans les paragraphes suivants, nous détaillons les méthodes de regroupement hiérarchique ascendant et basées sur la théorie des graphes, utilisées dans les expériences, ainsi que les différents modèles possibles pour le calcul de similarités.

3.3.3.1 Regroupement hiérarchique ascendant

Le regroupement hiérarchique ascendant (*Hierarchical Agglomerative Clustering*, ou HAC) [Lance and Williams, 1967] consiste à regrouper successivement des vecteurs ou des classes (ici, les segments) en fonction d'une mesure de distance qui caractérise leur similarité. C'est un procédé itératif, partant de n classes initiales. A chaque itération, les deux classes les plus proches sont regroupées en une, et la distance avec les autres classes est mise à jour. L'algorithme s'arrête lorsqu'il ne reste plus qu'une classe ou lorsqu'un critère d'arrêt est atteint (par exemple, lorsque la distance décidant du prochain regroupement atteint un seuil λ, ou lorsqu'un nombre de classes donné est atteint, si on connaît le nombre total de locuteurs du document). L'algorithme est illustré par la figure 3.1

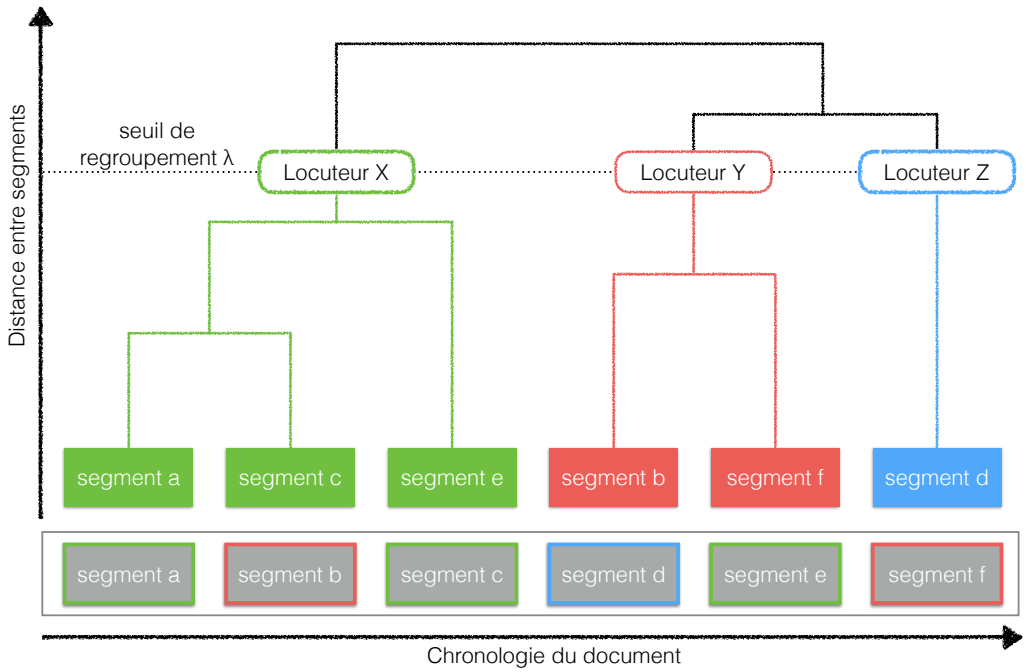

FIGURE 3.1 – Principe du regroupement hiérarchique ascendant (HAC) appliqué à un document audio pré-segmenté.

Concernant la mise à jour des distances, elle peut se faire par mise à jour du modèle représentant la classe fusionnée [Gish et al., 1991], puis calcul des nouvelles dis-

tances. Il existe également des critères standard de mise à jour de distances, comme le saut maximum (*complete-linkage*) [Defays, 1977], minimum (*single-linkage*) [Sibson, 1973] ou le critère de Ward (*Ward-linkage*) [Ward Jr, 1963], où les modèles ne sont pas ré-estimés. Dans les travaux présentés, nous utiliserons le saut minimum ou maximum.

Regroupement à saut minimum Pour ce critère, lors de la fusion de deux classes en une seule, la distance entre la nouvelle classe et n'importe quelle autre classe est égale à la plus petite distance entre les segments de la nouvelle classe et de l'autre classe. Lors d'un regroupement, soit $\boldsymbol{U}$ la nouvelle classe issue de celui-ci, constituée d'éléments (ou vecteurs) ($\boldsymbol{u_i}$), et une autre classe $\boldsymbol{V}$, constituée des éléments ($\boldsymbol{v_j}$), la mise à jour des distances se fait selon l'équation qui suit.

$$dist(\boldsymbol{U}, \boldsymbol{V}) = min_{i,j}(dist(\boldsymbol{u_i}, \boldsymbol{v_j})) \tag{3.51}$$

Le regroupement par saut minimum est un regroupement de proche en proche.

Regroupement à saut maximum Pour le regroupement à saut maximum, la mise à jour des distances se fait selon les *maxima*, comme illustré par l'équation suivante.

$$dist(\boldsymbol{U}, \boldsymbol{V}) = max_{i,j}(dist(\boldsymbol{u_i}, \boldsymbol{v_j})) \tag{3.52}$$

L'utilisation du saut maximum est une approche conservatrice, qui permet de regrouper les segments en classes homogènes, dont le diamètre ne peut dépasser la valeur du seuil de regroupement λ.

3.3.3.2 Lien avec la théorie des graphes

Chacune des deux méthodes de regroupement hiérarchique précitées peut avoir une interprétation en théorie des graphes. Si l'on considère la matrice de distances comme un graphe, chaque index de la matrice correspond un sommet du graphe (qui symbolise un *i-vector*), et à chaque distance entre *i-vectors* correspond une arête. On appelle composante connexe tout sous-graphe maximal dont, pour chaque paire de sommets, il existe une suite d'arêtes permettant de les relier.

Toute classe issue d'un regroupement à saut maximum a un diamètre maximal λ, c'est-à-dire que tous les sommets du sous-graphe représentant cette classe sont reliés entre eux par des arêtes de poids maximal λ. On parle alors de composante connexe complètement connectée.

La philosophie du regroupement hiérarchique ascendant à saut minimum consiste à dire que si deux sommets (ou vecteurs) ϕ_i et ϕ_j sont reliés par une arête de poids inférieur à λ (ou éloignés d'une distance inférieure à λ), alors ils appartiennent à la

même classe. Les classes se forment donc par association, ou chaînage. Le résultat de ce regroupement est alors une composante connexe : pour toute paire de vecteurs de la classe, il existe un chemin d'arêtes de poids inférieur à λ permettant de les relier. La figure 3.2 illustre la méthode de regroupement.

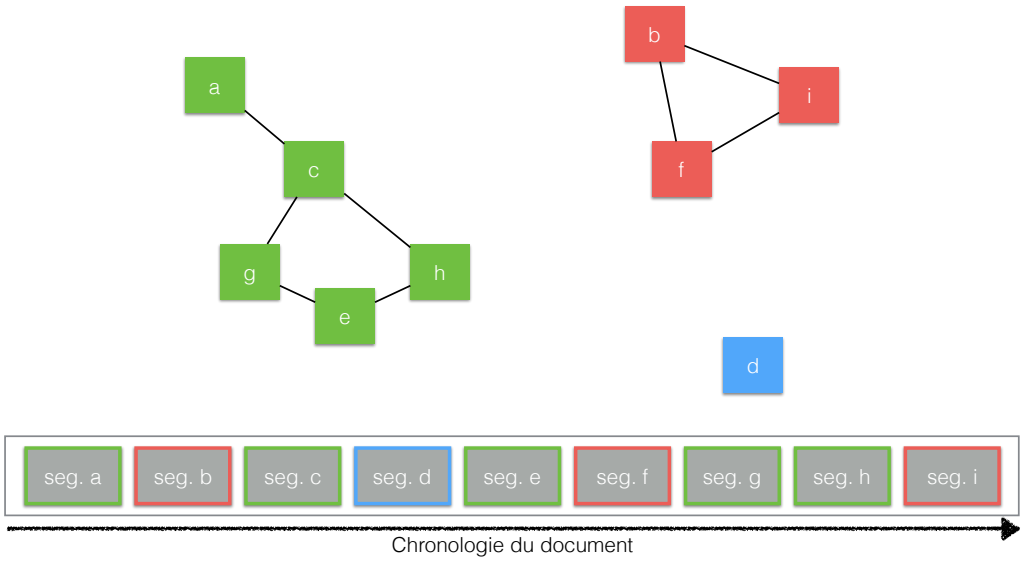

FIGURE 3.2 – Principe du regroupement en composantes connexes (CC) appliqué à un document audio pré-segmenté.

Terminologie Ces dernières années, la méthode de regroupement ILP a été proposée dans le cadre du regroupement en locuteurs [Dupuy et al., 2014b; Rouvier and Meignier, 2012], en remplacement du regroupement HAC à saut maximum (généralement appelé simplement HAC). Le regroupement ILP est un problème d'optimisation sous contraintes, qui vise à générer un nombre minimal de classes, tout en minimisant la dispersion au sein de chaque classe. Il s'agit d'un algorithme qui ne s'exécute pas en temps polynomial (contrairement aux regroupements hiérarchiques), ce qui pose problème lorsqu'on cherche à traiter des collections volumineuses. C'est pourquoi nous nous limiterons, dans ce manuscrit, à mentionner l'existence de la méthode.

Afin de limiter le temps de calcul de la résolution du problème ILP, les auteurs de [Dupuy et al., 2014b] ont proposé de décomposer le problème de regroupement ILP global en sous problèmes indépendants. En se basant sur la théorie des graphes, ils ont proposé de formaliser le problème du regroupement comme celui de la division d'un graphe complètement connecté en sous graphes. Le graphe complètement connecté représente la matrice globale des similarités. A partir de ce graphe global, ils ont proposé d'en extraire des sous graphes sous la forme de composantes connexes. Chaque sous graphe devient alors un problème de regroupement indépendant, et seules les composantes connexes dites "complexes" (c'est-à-dire n'ayant pas un élément central connecté à tous les autres) sont résolues par ILP. Dans l'article

en question [Dupuy et al., 2014b], la génération des composantes connexes étant vue comme une approche divisive, le lien n'est pas établi avec le regroupement agglomératif hiérarchique ascendant à saut minimum.

Ce lien entre regroupement dit "en composantes connexes" (CC) et regroupement HAC à saut minimum a seulement été établi à la fin de cette thèse. Ainsi, dans la suite du manuscrit, nous parlerons de regroupement **HAC** pour le regroupement hiérarchique ascendant à saut **maximum** et de regroupement **CC** pour le regroupement hiérarchique ascendant à saut **minimum**, c'est-à-dire en Composantes Connexes, tout en gardant à l'esprit que le regroupement HAC génère en réalité aussi des composantes connexes (complètement connectées).

3.3.3.3 Modèles utilisés

Quelle que soit la méthode de regroupement utilisée, celle-ci exploite la notion de similarité ou de distance entre segments. La mesure de similarité dépend donc de la représentation utilisée pour modéliser les segments.

Modèle mono-gaussien Une première architecture de regroupement simple se base sur la mesure ΔBIC [Chen and Gopalakrishnan, 1998]. Elle reprend la modélisation mono-gaussienne pour chaque segment et consiste à calculer les similarités entre tous les segments. Cette méthode est généralement un prérequis aux deux suivantes, car elle permet de fusionner les segments en sessions-locuteurs suffisamment longues pour utiliser des modélisations plus complexes. En effet, le coefficient de pénalité BIC est généralement choisi de façon à minimiser le taux de manqués, ce qui a pour conséquence de produire des segments purs au sens du locuteur. Le regroupement BIC aura donc tendance à générer plusieurs segments consécutifs pour un même locuteur.

Nous faisons une distinction entre segments et sessions-locuteurs : un segment est une sous-partie continue d'un document audio. Lorsque nous regroupons des segments à l'échelle d'un document, nous ne pouvons plus parler de segments, d'où le terme session-locuteur. Dans le contexte du regroupement inter-document, nous pourrons également parler de classe-locuteur : une classe regroupant des segments ou des sessions-locuteurs.

Modèle GMM La modélisation à mélange de gaussiennes est utilisée par [Barras et al., 2006; Ghaemmaghami et al., 2013; **?**], le regroupement est basé sur le score CLR (voir 3.2.4.3). Un corpus d'apprentissage est nécessaire pour estimer un GMM-UBM, et chaque session-locuteur est modélisée par adaptation MAP (voir 3.2.4.2).

Modèle *i-vector* Chaque session-locuteur du document traité peut également être modélisée par un *i-vector*, qui est la représentation à l'état de l'art. Pour ce

faire (cf. section 3.2.5), les modèles GMM/UBM (cf. section 3.2.4) et TV doivent être préalablement entraînés sur un corpus d'apprentissage. Les *i-vectors* peuvent être normalisés selon différentes méthodes (WCCN, EFR, SNN), avant le calcul de similarités. La similarité la plus intuitive est la similarité cosine, mais le calcul de scores PLDA (cf. section 3.2.6) est également possible, en ayant préalablement estimé les matrices $\mathbf{\Phi}$ et $\mathbf{\Lambda}$ sur un corpus d'apprentissage annoté en locuteurs. Les scores PLDA sont généralement plus précis, car ils tiennent compte de la variabilité inter- et intra-locuteur. La modélisation *i-vector* en SRL a par exemple été utilisée par [Rouvier et al., 2013; Shum et al., 2011b; Silovsky and Prazak, 2012].

3.4 SRL de collection

Dans ce mémoire, nous nous intéressons tout particulièrement à la tâche de SRL appliquée à des collections de documents audiovisuels. Il ne s'agit pas seulement de regrouper les segments de parole de chaque locuteur au sein d'un document audio, mais au niveau d'une collection de documents. A ce titre, les locuteurs récurrents doivent être identifiés par le même label dans chaque document de la collection.

Dans la littérature, le problème de la SRL de collection est considéré comme une étape supplémentaire de la tâche de SRL intra-document. Même si la terminologie diffère (*Speaker Linking* chez [Bourlard et al., 2013; Ferràs and Bourlard, 2012; Ghaemmaghami et al., 2013; Meignier et al., 2002; Van Leeuwen, 2010], *Cross-Show Speaker Diarization* pour [Tran et al., 2011b; Yang et al., 2011b]), elle a tendance à se normaliser en ce sens (*Speaker Diarization and Linking* dans [Ferras et al., 2016a; Ghaemmaghami et al., 2015]). L'idée générale est de traiter chaque document indépendamment, puis de regrouper les classes-locuteurs produites par le traitement intra-document à l'échelle de la collection, sans remettre en cause la segmentation.

3.4.1 La question de la variabilité inter-document

C'est donc le regroupement inter-document qui caractérise la SRL de collection. Par conséquent les méthodes de SRL déjà utilisées pour le regroupement intra-document ont été adaptées au regroupement inter-document, que ce soit au niveau de la modélisation ou du regroupement, avec deux grandes différences. D'abord le nombre de segments à regrouper est d'un à plusieurs ordres de grandeur supérieur. Ensuite, la variabilité inter-document est plus importante que la variabilité intra-document. Certains locuteurs peuvent en effet parler dans un environnement acoustique différent selon les émissions (micro-trottoir vs. interview en studio, par exemple). Par conséquent, la compensation de la variabilité intra-locuteur/inter-documents est bien plus importante que lorsqu'on travaille sur un seul document. Enfin, la chronologie de certaines collections peut faire que l'âge des locuteurs aug-

mente la variabilité intra-locuteur [Doddington, 2012; Matveev, 2013].

Du côté des méthodes employées, on pourra citer la modélisation mono gaussienne avec BIC [Yang et al., 2011b], GMM/CLR avec regroupement hiérarchique [Barras et al., 2006; Tran et al., 2011b], *i-vector* avec partition de graphe [Dupuy et al., 2012b; Shum et al., 2013a] et regroupement hiérarchique, ILP [Dupuy et al., 2012b]. Cependant, comme la variabilité intra-locuteur/inter-document est plus importante que dans le cas intra-document, les méthodes basées sur la représentation *i-vector* et utilisant des techniques de compensation de variabilité telles que la WCCN ou la PLDA sont plébiscitées dans la littérature récente. A l'échelle de la collection, on distingue deux architectures de regroupement : le regroupement global (e.g. [Dupuy et al., 2012b]) et incrémental (e.g. [Dupuy et al., 2014a]), qui sont discutées par [Van Leeuwen, 2010].

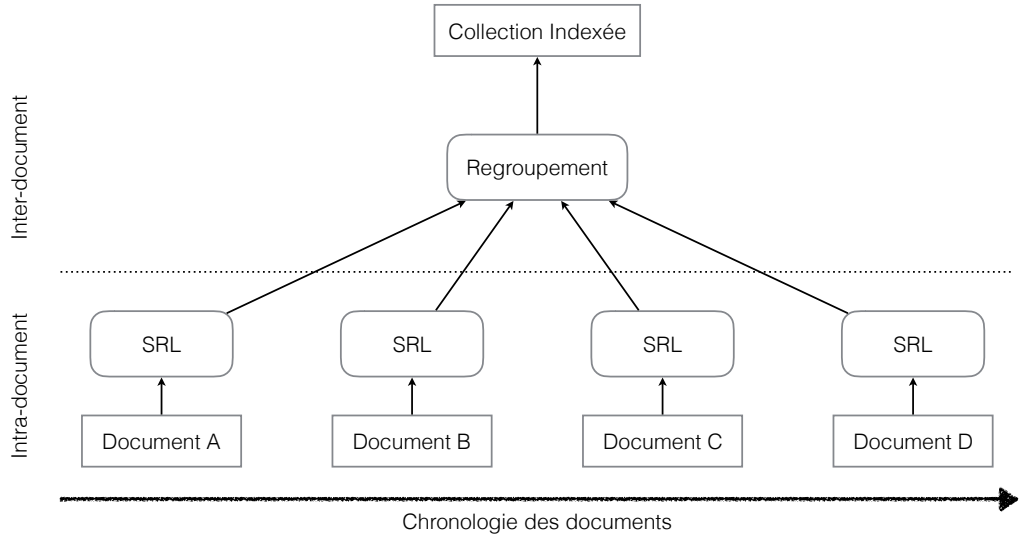

FIGURE 3.3 – Principe du regroupement global pour la SRL de collection.

3.4.2 Regroupement Global

Le regroupement global consiste à traiter la collection comme un tout, et de permettre des regroupements inter-document indépendamment de l'ordre chronologique des documents. Les premières approches sur le regroupement global de collection [Tran et al., 2011a; Yang et al., 2011a] consistaient à considérer la collection comme un document unique. Le principe était de d'abord traiter chaque document séparément, et d'arrêter le traitement intra-document à l'étape de la segmentation en locuteurs. Ensuite, les auteurs concatènent les documents de la collection en un seul et considèrent le problème comme un cas classique de regroupement intra-document. Dès lors, les méthodes appliquées sont les méthodes classiques de regroupement intra-document, décrites à la section 3.3.3 : Gaussienne/BIC suivie de GMM/CLR, selon un regroupement hiérarchique ascendant. La seule limite de

l'approche est la complexité du regroupement hiérarchique ascendant, qui est quadratique, ce qui implique qu'il existe une taille de collection limite pouvant être traitée selon cette approche.

Pour alléger la combinatoire du regroupement inter-document (celui-ci implique de calculer les distances entre toutes les paires de segments possibles), les auteurs ont également testé une architecture hybride, qui consiste à effectuer le regroupement Gaussienne/BIC au sein de chaque document séparément, afin de réduire le nombre de classes à comparer à l'échelle de la collection. Dans ce cas, seul le regroupement GMM/CLR est effectué sur l'ensemble des documents.

Cette architecture de regroupement global, illustrée par le figure 3.3 a été étudiée ces dernières années avec plusieurs variantes [Bourlard et al., 2013; Dupuy et al., 2012a; Ferràs and Bourlard, 2012; Ghaemmaghami et al., 2013] portant principalement sur les critères de regroupement (hiérarchique Ward ou à saut maximum, ILP) et les modèles de représentation des locuteurs (modèles JFA, *i-vector*), de calcul de scores (divergence KL, cosine, PLDA).

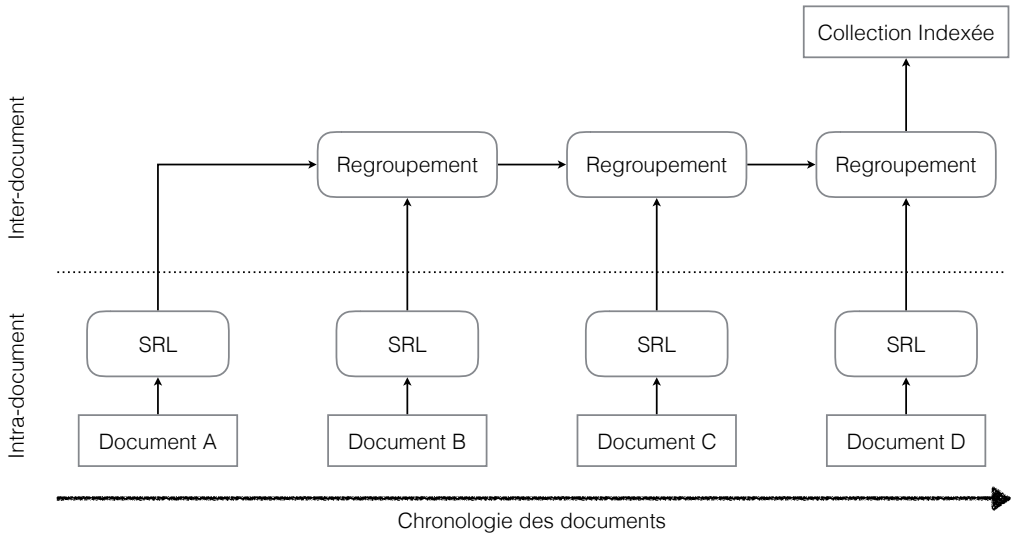

FIGURE 3.4 – Principe du regroupement incrémental pour la SRL de collection.

3.4.3 Regroupement Incremental

L'approche incrémentale répond à un besoin applicatif courant, où la collection s'enrichit avec le temps. En effet, l'approche globale pose deux problèmes majeurs. D'une part, elle est très consommatrice de ressources, puisqu'à chaque nouvel épisode, le regroupement est refait de manière globale, jusqu'à un point où la quantité de données à traiter est trop importante pour les ressources disponibles. D'autre part, à chaque nouvel épisode, le regroupement global peut remettre en question les regroupements entre les épisodes précédents. Applicativement, pour un utilisateur humain, il est difficile d'attribuer une identité à un label qui serait susceptible de changer à chaque nouvel épisode traité, c'est déroutant. Pour pallier le problème,

l'approche incrémentale consiste à ne pas remettre en question le passé à chaque nouvel épisode, et à n'effectuer le regroupement inter-document qu'entre le dernier document et les classes-locuteurs issues des regroupements entre les documents antérieurs. Le principe est illustré par la figure 3.4.

Initialement discuté et testé par [Tran et al., 2011a; Van Leeuwen, 2010; Yang et al., 2011a], c'était une des contraintes imposées du challenge *Multi Genre Broadcast* (MGB) [Bell et al., 2015], dont une des tâches évaluées était la SRL appliquée à des données télévisuelles de la BBC. La contrainte de traitement chronologique des documents présente l'avantage de limiter la combinatoire des regroupements inter-document possibles, et permet donc, comparativement à un système avec regroupement global, de gérer des collections de taille bien supérieure. En effet, lors de l'ajout d'un document, les seules distances à calculer sont entre les classes-locuteurs du document en question et celles issues des regroupements entre documents antérieurs.

Dans un système de regroupement incrémental ne faisant pas d'erreurs, la complexité est donc quadratique en nombre de locuteurs de la collection. Soit une collection de K épisodes, contenant N locuteurs au total. Dans le pire cas, les N locuteurs sont présents dans chaque document. La complexité du regroupement incrémental est alors de $\mathcal{O}(K \times N^2)$ contre $\mathcal{O}(K^2 \times N^2)$ pour le regroupement global, on gagne donc un ordre de grandeur. En contrepartie, le regroupement incrémental limite la combinatoire des regroupements possibles, ce qui peut donner un résultat final sous-optimal et donc dégrader les performances par rapport à un regroupement global.

3.5 Evaluation de la structuration des collections

3.5.1 Taux d'erreur de SRL

La métrique classique pour mesurer la performance de SRL est le taux d'erreur de SRL, en anglais Diarization Error Rate (DER). Il est calculé à partir d'une référence, annotée manuellement, et d'une hypothèse générée par la machine. Cette métrique a été introduite par le *National Institute of Standards and Technology* (NIST), lors de la campagne d'évaluation [NIST, 2000] pour la tâche de segmentation en locuteurs. A partir de la meilleure correspondance possible entre les locuteurs de la référence et les locuteurs de l'hypothèse, le taux d'erreur correspond à la somme de la durée des segments de parole manquée, de fausse alarme et de mauvaise attribution des locuteurs, sur la durée de parole totale (voir équation 3.53).

$$DER = \frac{err_{locuteur} + err_{FA} + err_{Miss}}{durée_{totale}} \tag{3.53}$$

Lorsqu'on traite des collections, on distingue généralement les taux d'erreur intra-document et inter-document. Le taux d'erreur intra-document sert à évaluer la per-

formance au sein de chaque document traité (dans notre cas, chaque épisode d'une émission), et peut être moyenné sur la collection (par exemple, l'émission). Le taux d'erreur inter-document, quant à lui, permet d'évaluer la qualité du regroupement en locuteurs à travers la collection. Dans ce cas, le système doit identifier chaque locuteur récurrent de la collection par la même étiquette dans tous les documents de ladite collection. Dans le cadre de la structuration des collections, la minimisation du taux d'erreur de SRL est donc un critère de premier choix.

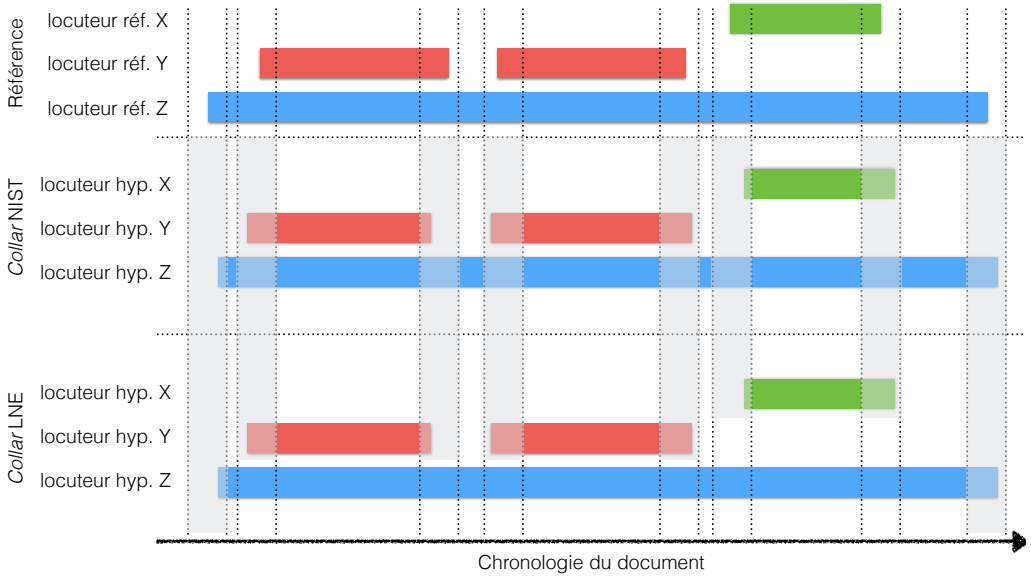

FIGURE 3.5 – Effet de la tolérance aux frontières sur la parole superposée, selon le choix de la méthode (NIST ou LNE).

Influence du *collar* Comme les références sont annotées par des humains et qu'il est difficile de définir empiriquement ce qui constitue le début ou la fin d'un segment, il est commun d'autoriser une marge d'erreur (ou *collar*) pour le calcul du DER, autour des frontières des segments de référence. Historiquement, par la méthode NIST, le *collar* exclut du calcul du DER les zones frontières de tout segment de référence. Cependant, cela provoque un effet de bord important lorsqu'on cherche à évaluer des zones de parole superposée. Proposée plus récemment par le Laboratoire National de Métrologie et d'Essais (LNE) [Galibert, 2013], une autre méthode d'application du *collar* permet de gérer ce problème. Premièrement, le *collar* n'est plus vu comme servant à exclure des zones de calcul du DER, mais sert vraiment de marge de tolérance aux frontières des segments de référence. Ainsi, si deux segments hypothèse consécutifs sont associés à deux segments de référence consécutifs qui n'ont pas exactement la même frontière, aucune erreur n'est décomptée. Ensuite, il ne s'applique que sur les segments hypothèse ayant été associés à une référence. Sur de la parole superposée, il n'est donc pas automatiquement décompté deux fois. La méthode a également des limites, puisque plus elle effectue d'appariements, plus elle applique le *collar*, et donc plus elle retire de la parole de l'évaluation. L'application

du *collar* sur la parole superposée selon chaque méthode est illustrée par la figure 3.5 [1]. Pour toutes les expériences présentées par la suite, l'outil de calcul du DER sera celui du LNE.

3.5.2 Analyse en locuteurs

La tâche de SRL de collection étant assez récente, on trouve peu de métriques inhérentes à l'analyse en locuteurs dans la littérature. On peut tout de même citer les travaux de [Van Leeuwen, 2010], qui ont proposé les notions d'impureté et d'entropie, et de [Ferràs and Bourlard, 2012], qui utilisent les métriques plus classiques de pureté et couverture, utilisées auparavant pour évaluer la SRL intra-document [Gauvain et al., 1998].

3.5.2.1 Impureté et Entropie

Les auteurs de [Van Leeuwen, 2010] ne travaillaient pas sur la tâche de SRL de collection à proprement parler, mais seulement sur une tâche d'appariement en locuteurs, qu'on peut rapprocher de l'étape de regroupement inter-document de la SRL de collection. La principale différence est que les données sur lesquelles travaillaient les auteurs étaient des segments de parole mono-locuteur, le but étant d'apparier (regrouper) les segments prononcés par un même locuteur. Dans notre cas, les segments qu'on cherche à regrouper lors de l'étape inter-document ne sont pas nécessairement mono-locuteur et peuvent contenir la voix de plusieurs individus. Pour évaluer la qualité de l'appariement, les auteurs ont proposé les notions d'impureté et d'entropie de classe, et d'impureté et d'entropie en locuteurs. L'idée derrière ces métriques était de s'affranchir de la notion de durée de parole, l'homogénéité en locuteur des segments à apparier étant parfaite.

Soient $\mathcal{C}_i$, $i \in [1..C]$, l'ensemble des classes-locuteurs résultant du regroupement en locuteurs, on considère que chaque classe-locuteur correspond au locuteur de référence lui ayant apporté le plus de segments. L'impureté moyenne de classe est définie comme :

$$I^C = 1 - \frac{1}{N} \sum_i f_{i1} \tag{3.54}$$

N est le nombre total de segments du corpus d'évaluation, et f_{i1} correspond à la fréquence du locuteur de référence principal dans la classe-locuteur i. Par exemple, si une classe-locuteur a agrégé 5 segments, dont 3 appartiennent au même locuteur de référence, $f_{i1} = 3$. L'entropie est alors exprimée comme :

$$H^C = 1 - \frac{1}{N} \sum_i n_i {H_i}^C \tag{3.55}$$

1. Cette figure a été inspirée de la thèse [Delgado et al., 2015]

avec

$$H_i{}^C = -\sum_k p_{ik} log_2(p_{ik}) \tag{3.56}$$

$p_{ik} = f_{ik}/n_i$, n_i étant le nombre de segments total de la classe-locuteur i, f_{ik} la fréquence du locuteur de référence k. L'impureté et l'entropie de classe permettent donc d'évaluer la pureté des classes formées. L'impureté et l'entropie en locuteurs se définissent de manière analogue en inversant les notions de classe et de locuteur. Elles permettent d'évaluer la qualité du regroupement des locuteurs de référence. Dans l'article, les auteurs choisissent alors le système donnant le meilleur compromis entre pureté des classes et qualité du regroupement des locuteurs de référence.

3.5.2.2 Pureté et couverture

De façon analogue, mais en tenant compte de la durée de parole, les auteurs de [Ferràs and Bourlard, 2012], qui évaluaient la tâche de SRL de collection appliquée à des enregistrements de réunion, ont utilisé les notions de pureté et couverture de classes-locuteurs [Gauvain et al., 1998]. Pour une classe-locuteur donnée, la pureté est définie comme le ratio entre la durée de parole issue de son locuteur majoritaire sur la durée de parole totale contenue dans la classe-locuteur en question.

$$p_i = \frac{d_{i1}}{\sum_j d_{ij}} \tag{3.57}$$

Inversement, pour un locuteur donné, le taux de couverture est égal au rapport de sa plus importante durée de parole associée à une classe-locuteur sur la durée de parole totale du locuteur.

$$c_j = \frac{d_{1j}}{\sum_i d_{ij}} \tag{3.58}$$

d_{ij} correspond à la durée de parole du locuteur j associé à la classe i, l'indice 1 indiquant le locuteur le plus important d'une classe, ou l'inverse. Les performances du système sont alors décrites avec des valeurs de pureté et couverture moyennes.

$$p = \frac{\sum_i (p_i \sum_j d_{ij})}{\sum_{ij} d_{ij}} \tag{3.59}$$

$$c = \frac{\sum_j (c_j \sum_i d_{ij})}{\sum_{ij} d_{ij}} \tag{3.60}$$

Pour ne retenir qu'une valeur, on utilise parfois la moyenne géométrique, comme l'on fait les auteurs [Valente and Wellekens, 2004] (même s'ils ne définissent pas la pureté de la même façon).

$$K = \sqrt{pc} \tag{3.61}$$

A l'extrême, si on ne regroupe pas du tout, la pureté des classes est de 1, tandis que si on regroupe tous les segments en une seule classe, c'est le taux de couverture moyen qui est de 1.

3.6 Bilan

Dans ce chapitre, nous avons vu les méthodes classiques de modélisation des segments de parole du point de vue locuteur. Elles nous ont permis d'étudier les architectures classiques de SRL intra-document et de SRL de collection. Pour la SRL de collection, nous avons notamment souligné l'importance de la compensation de la variabilité intra-locuteur/inter-document, via des méthodes telles que la WCCN ou la PLDA appliquées à la représentation *i-vector*. Nous avons notamment détaillé les méthodes de regroupement HAC ou CC, ainsi que les deux architectures classiques de regroupement inter-document : global ou incrémental, qui répondent à des besoins applicatifs différents. Enfin, nous avons discuté des métriques d'évaluation de la qualité d'un système de SRL de collection. L'ensemble de ces concepts vont nous permettre de construire notre système de SRL de collection, que nous présentons et évaluons dans le chapitre qui suit.

Chapitre 4

Analyse d'un système de SRL à l'état de l'art

Résumé

Ce chapitre traite de l'analyse d'un système de SRL à l'état de l'art. Le système présenté consiste en une approche en deux étapes (SRL intra-document, puis regroupement inter-document). Il utilise la modélisation i-vector comme représentation de plus haut niveau et l'architecture de regroupement est de type global. Après avoir présenté le protocole d'évaluation, on évalue de façon exhaustive les performances du système. On quantifie d'abord les performances maximales atteignables en étudiant le comportement de la métrique DER, puis on compare différentes versions du système, utilisant différentes méthodes de calcul de similarités (cosine, cosine/WCCN, PLDA) et de regroupement (HAC ou CC). Les résultats montrent que le système le plus performant utilise la PLDA et le regroupement HAC, avec un DER inter-document de 15.7% sur BFM et 18.1% sur LCP. Le chapitre se termine par l'analyse en locuteurs des performances, à l'aide de nouvelles métriques : le taux d'erreur nominal et le taux d'erreur classe, en faisant le lien avec les types de locuteurs identifiés au chapitre 2. L'analyse montre les limites des systèmes à l'état de l'art, qui peinent à regrouper correctement les invités récurrents : l'écart de performances entre SRL intra- et inter-document (du simple au double) montre que la question de la modélisation de la variabilité intra-locuteur/inter-document n'est pas complètement résolue et constitue un axe de recherche pertinent.

4.1 Présentation du système *baseline*

Pour répondre à la problématique de la SRL de collection, nous proposons le système *baseline* suivant, illustré par la figure 4.1 et détaillé ci-après. Il a été développé avec la librairie Python SIDEKIT [Larcher et al., 2016] et repose sur une

architecture de regroupement global pour traiter la collection. Dans l'ensemble des chapitres qui suivent, nous utiliserons cette architecture de regroupement global. La question du regroupement incrémental fera l'objet d'un chapitre dédié (chapitre 8).

Le système présenté est une variante de celui du LIUM [Dupuy et al., 2014b, 2012a], où le regroupement ILP est remplacé par un regroupement HAC ou CC. Ce système traite la collection en deux étapes : d'abord la SRL intra-document, réalisée séparément pour chaque document, puis un regroupement inter-document global, réalisé sur l'ensemble des sorties des documents de la collection. A titre de comparaison, les systèmes de SRL les plus récents, proposés pour les campagnes d'évaluation REPERE [Galibert and Kahn, 2013b], tels que celui du LIMSI [Bredin et al., 2013], du LIUM [Dupuy et al., 2014b] ou d'Orange [Favre et al., 2013], ou pour le Challenge MGB [Bell et al., 2015], tel que celui proposé par [Karanasou et al., 2015], reposent sur cette architecture en deux passes, avec des variantes sur la méthode de segmentation, de représentation des segments acoustiques ou de mesure de similarités. Pour la modélisation des segments-locuteurs, on utilise la représentation *i-vector*, qui est la représentation à l'état de l'art. Elle est par exemple employée par [Ferras et al., 2016b; Villalba et al., 2015] dans le cadre du Challenge MGB.

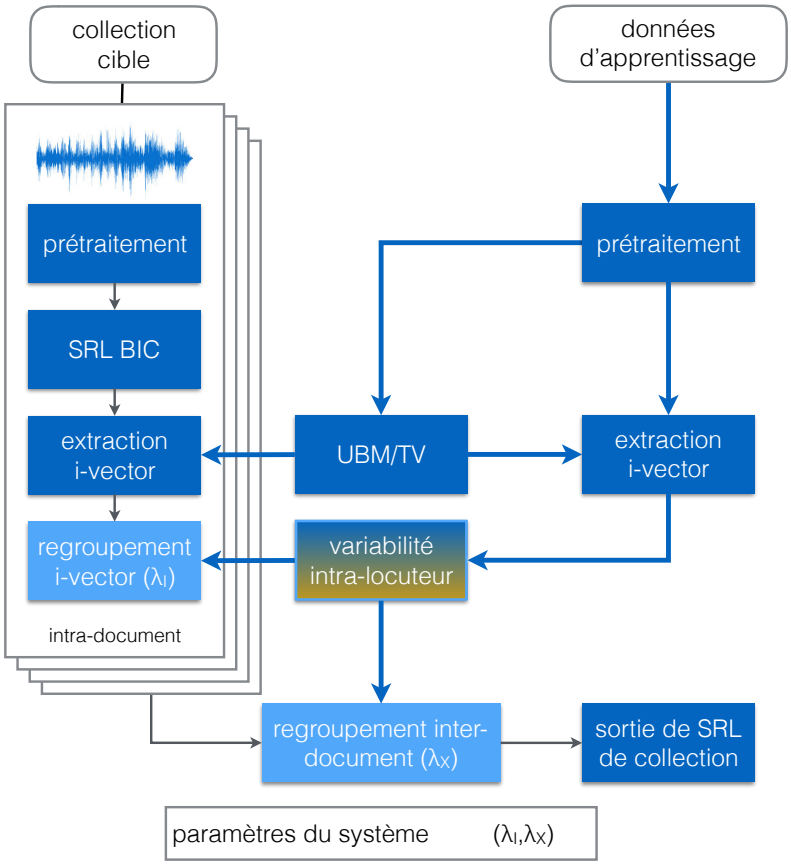

FIGURE 4.1 – Schéma du système de SRL de collection *baseline*.

4.1.1 SRL intra-document

Le prétraitement consiste en l'extraction de 42 descripteurs (13 MFCCs auxquels s'ajoute un coefficient pour l'énergie, complétés des Δ et $\Delta\Delta$), suivi d'une détection de parole à 8 Gaussiennes, utilisant l'algorithme de Viterbi. Pour la segmentation, les frontières sont d'abord estimées par une segmentation GLR standard, utilisant une fenêtre glissante de 20ms avec recouvrement. Elle est ensuite affinée par une segmentation gaussienne/BIC. Enfin deux regroupements hiérarchiques à saut maximum sont appliqués successivement pour fusionner/annoter les segments au sein de chaque document : d'abord un regroupement utilisant les similarités BIC (inclus dans le processus *SRL BIC*), puis un regroupement basé sur des similarités entre *i-vectors*. Avant l'extraction des *i-vectors*, les vecteurs acoustiques sont centrés et réduits par classe-locuteur issue du regroupement BIC. Le seuil de regroupement HAC BIC est choisi empiriquement pour générer des classes-locuteurs homogènes, afin de permettre l'extraction de *i-vectors* représentant bien un seul locuteur. Le seuil du regroupement HAC appliqué aux *i-vectors* est noté λ_I. A la fin de la SRL intra-enregistrement, chaque classe-locuteur est réduite à un seul *i-vector* : il s'agit de la moyenne des *i-vectors* la constituant.

La représentation *i-vector* utilisée dans les expériences suivantes est apprise sur les données d'apprentissage, à partir d'un GMM/UBM de covariance diagonale. La modélisation de la variabilité intra-locuteur peut prendre trois formes. Il peut s'agir de :

- Aucune modélisation lorsque seule la représentation *i-vector* est utilisée, auquel cas la similarité est la cosine entre *i-vectors*.
- La matrice WCCN, utilisée pour normaliser les *i-vectors*. Dans ce cas, la similarité est aussi la cosine.
- Les matrices PLDA, utilisées pour calculer un logarithme de rapport de vraisemblance, qui constitue la mesure de similarité.

4.1.2 Regroupement inter-document

Pour le regroupement inter-document, une matrice de similarités est calculée entre tous les *i-vectors* issus de chaque document, représentant chacun une classe-locuteur intra-document. Cette matrice de similarités est utilisée pour réaliser le regroupement global des *i-vectors* en classes-locuteurs à l'échelle de la collection. Le seuil de regroupement inter-document est noté λ_X.

Si pour le regroupement intra-document, nous utilisons uniquement un regroupement hiérarchique à saut maximum (HAC), nous comparerons, pour le regroupement inter-document, le regroupement HAC et en composantes connexes (CC). De plus, lors du regroupement inter-document, nous n'interdisons pas les regroupements intra-document. Par conséquent, selon le type de regroupement inter-document uti-

lisé, ou si $\lambda_X > \lambda_I$, des classes-locuteurs issues d'un même document peuvent fusionner.

Dans la suite de ce manuscrit, nous nous concentrerons principalement sur les regroupements *i-vector* intra- et inter-document et le calcul de similarités entre *i-vectors*, à partir d'une segmentation BIC donnée.

4.2 Evaluation du système *baseline*

4.2.1 Expériences *oracle*

Avant de tester les différents paramètres pour la tâche de SRL de collection, on s'intéresse aux performances d'un système de regroupement *i-vector* idéal. Le module de SRL BIC est configuré pour générer des sessions-locuteurs pures, afin que les *i-vectors* extraits à partir de chaque session-locuteur représentent bien un unique locuteur. Cependant, dans les faits, certaines sessions-locuteurs peuvent être impures. A partir de cette première passe de SRL BIC, on considère qu'un système de regroupement *i-vector* idéal regrouperait les sessions selon leur locuteur majoritaire. Cette notion de regroupement idéal nous permet de quantifier les performances maximales atteignables du système de regroupement inter- et intra-document compte-tenu des erreurs de segmentation (parole et locuteur).

4.2.1.1 Génération des références

Les références sont des fichiers qui listent les tours de parole des documents à évaluer, ainsi que l'identité du locuteur associé. Elles sont produites à partir d'annotations effectuées par des annotateurs humains et servent à calculer le DER.

Dans ce manuscrit, comme nous nous intéressons particulièrement à la tâche de SRL de collection, nous avons décidé d'exclure la parole superposée pour générer les références (hormis pour une expérience contrastive à la section 4.2.1.3). Le traitement de la parole superposée constitue un problème spécifique de la segmentation en locuteurs. Cette parole, exclue des références, représente 10 minutes de signal pour la collection LCP et 33 minutes pour la collection BFM. Dans les expériences, la parole superposée sera donc traitée par le système de SRL mais sera considérée comme du bruit lors de l'évaluation : si le système y détecte de la parole, elle sera comptée comme de la fausse alarme.

4.2.1.2 Influence du *collar*

Comme nous l'avons vu à la section 3.5.1, lors du calcul du DER, il est courant d'utiliser une tolérance aux frontières des segments, ou *collar*. Dans la table 4.1, on présente les taux d'erreur de SRL pour les deux collections cibles, après la segmentation automatique et un regroupement intra- et inter-document idéal (*oracle*), pour

différentes durées de tolérance aux frontières. Notons que plus le *collar* est élevé, plus les taux d'erreur diminuent, car la tolérance aux frontières des segments est plus grande. Pour la suite des expériences, nous travaillerons avec une tolérance de 250 millisecondes.

Collar (ms)	LCP		BFM	
	DER-I	DER-X	DER-I	DER-X
0	8.9	8.9	10.2	10.2
250	5.9	6.2	7.8	7.8
500	4.7	5.1	6.5	6.5

TABLE 4.1: Performances de SRL oracle, pour les deux collections, à différents *collars*

La table permet de constater qu'avec un *collar* de 250 millisecondes, après regroupement inter- puis intra-document idéal, basé sur les *i-vectors*, le DER intra-document (DER-I) atteint 5.9% pour LCP et 7.8% pour BFM. On note que le DER inter-document (DER-X) est quasiment identique, à 6.2% pour LCP et 7.8% pour BFM. Ces taux d'erreur sont donc les meilleurs taux atteignables sans remettre en question tout ce qui précède l'extraction des *i-vectors* dans la chaîne de traitement. Il existe donc une part d'erreur non soluble par notre regroupement *i-vector*.

4.2.1.3 Cas particulier de la parole superposée

La table 4.2 présente les résultats de deux expériences comparatives où, dans le premier cas, la parole superposée est considérée telle quelle (incluse dans les références), et dans le second comme du bruit (exclue des références). L'inclusion de la parole superposée dans l'évaluation nous permet de mesurer son impact sur le DER. Le DER étant composé de trois types d'erreurs (fausse alarme, parole manquée et erreur d'attribution), on les détaille dans la table, pour une tolérance aux frontières de 250 millisecondes.

Que la parole superposée soit comptée ou non, les taux d'erreur de SRL intra-document sont quasiment identiques. Cependant, le détail des erreurs montre des différences. On peut remarquer que lorsque la parole superposée est considérée comme du bruit, environ 60% du DER est dû à une mauvaise détection de parole (majoritairement de la fausse alarme), tandis que les 40% restants sont de l'erreur d'attribution de segments.

Les résultats montrent que lorsque la parole superposée est considérée comme telle (incluse dans les références), la fausse alarme est plus faible et la parole manquée plus élevée. En effet, comme le système ne gère pas la parole superposée (il ne génère qu'un locuteur hypothèse là où il devrait en générer deux), l'outil d'évaluation considère l'absence d'une deuxième hypothèse comme de la parole manquée. Quand au contraire, la parole superposée est considérée comme du bruit, le système la traite quand même et y détecte de la parole, d'où l'écart de fausse alarme.

Parole superposée	Type d'erreur	LCP	BFM
considérée comme telle	DER-I *oracle*	5.9	7.6
	fausse alarme	1.5	1.2
	parole manquée	1.8	2.7
	erreur d'attribution	2.6	3.6
considérée comme du bruit	DER-I *oracle*	**5.9**	**7.8**
	fausse alarme	3.0	3.7
	parole manquée	0.7	1.0
	erreur d'attribution	2.2	3.0

TABLE 4.2: Détail des taux d'erreur intra-document (DER-I) pour un collar de 250 ms.

Quelle que soit la stratégie d'évaluation (avec ou sans parole superposée), l'erreur d'attribution est une erreur résiduelle et non soluble par le regroupement. Elle est due au module de segmentation et regroupement BIC : certains *i-vectors* sont extraits à partir de segments qui contiennent la voix de plus d'un locuteur. Pour la suite des expériences décrites dans ce manuscrit, la parole superposée sera considérée comme du bruit.

4.2.2 Expériences *baseline*

4.2.2.1 Protocole d'évaluation

Dorénavant, nous effectuons des expériences comparatives sur les données en conditions réelles, avec différentes versions du système *baseline* (le regroupement *i-vector* n'est plus considéré idéal). Nous comparons différentes méthodes de calcul de similarité (cosine avec ou sans WCCN, PLDA) et deux méthodes de regroupement : le regroupement hiérarchique ascendant à saut maximum (HAC) et le regroupement en composantes connexes (CC).

Chaque expérience dépend d'une paire de seuils de regroupement λ_I (intra-document) et λ_X (inter-document). Comme les deux méthodes de regroupement utilisent la notion de distance, les seuils λ_I et λ_X doivent s'appliquer sur des valeurs analogues à des distances (bien qu'en réalité une distance ne peut pas être négative). Or lorsqu'on compare des *i-vectors*, on calcule des mesures de similarité. Si l'on note s la similarité entre deux *i-vectors* et d la distance, on propose donc les conversions suivantes :

- pour la PLDA, on considère les valeurs opposées aux rapports de vraisemblance.

$$d_{PLDA} = -s_{PLDA} \tag{4.1}$$

- pour la cosine (que ce soit avec compensation WCCN ou non), on utilise une transformation affine.

$$d_{cosine} = -200 \times s_{cosine} + 100 \tag{4.2}$$

La transformation affine choisie pour la similarité cosine permet de placer les seuils de regroupements d'intérêt dans le même intervalle que ceux de la PLDA. Les plages de valeurs de distance explorées sont les suivantes : de -90 à 30, avec un pas de 10 pour λ_I et λ_X. Pour la similarité cosine, avant transformation, cela revient à placer les seuils de regroupement dans un intervalle de similarités $[0.4, 0.9]$, avec un pas de 0.05.

Comme nous ne disposons que de deux collections pour les expériences, nous proposons d'évaluer la calibration en validation croisée. Pour une paire de seuils (λ_I, λ_X) qui optimise la tâche sur une collection, nous mesurons les performances de la même paire de seuils sur l'autre collection. On parle alors de configuration *dédiée*. De plus, nous présentons une configuration de seuils qui optimise la moyenne des performances sur les deux collections, on parle alors de configuration *commune*.

Enfin, pour permettre une analyse fine des résultats, nous décidons d'exclure du traitement certaines zones non annotées des collections cibles. C'est-à-dire que nous traitons le signal compris entre le début du premier segment annoté et la fin du dernier segment annoté. Dans la majorité des documents, l'annotation a été effectuée à partir de la fin du générique de début, jusqu'au début du générique de fin. Sur la collection BFM, sont également exclues du traitement les coupures publicitaires, au nombre de 3 par document. Ceci permet d'éviter des regroupements entre zones de parole annotée et zones de parole non annotée, qui rendraient difficile l'analyse des résultats : on ne peut évaluer la qualité du système que si on connait les tours de parole et l'identité des locuteurs traités.

4.2.2.2 Résultats

Le choix des dimensions des paramètres du système *baseline* (UBM, TV, PLDA) fait l'objet d'une annexe dédiée (voir annexe A). Pour les expériences présentées dans la suite de ce manuscrit, le système *baseline* est figé avec les paramètres suivants : GMM/UBM à 256 composantes, matrice TV de dimension 200 et matrice PLDA de dimension 100. Ce choix a été arrêté après les expériences préliminaires qui portaient uniquement sur le regroupement HAC. En effet, la configuration retenue est la plus performante avec les mesures de similarité WCCN et PLDA en regroupement inter-document HAC.

La table 4.3 résume les performances intra- et inter-document des systèmes en fonction du type de regroupement et du type de mesure de similarités, aux dimensions choisies (256 pour l'UBM, 200 pour la matrice TV et 100 pour la PLDA). Globalement, les résultats montrent que sur la collection BFM, l'utilisation des méthodes de compensation de variabilité WCCN et PLDA améliore efficacement les

résultats inter-document, de 19.3% à 13.4% dans le meilleur des cas (configuration *CC/dédiée*), tandis que pour la collection LCP, le gain est plus limité, de 19.5% à 18.1% (configuration *HAC/dédiée*), peut-être en raison d'une plus grande différence avec le corpus d'apprentissage.

collection	regroupement	configuration	cosine		cosine/WCCN		PLDA	
			DER_I	DER_X	DER_I	DER_X	DER_I	DER_X
BFM	HAC	*dédiée*	14.4	22.2	13.3	17.6	10.6	15.7
		commune	13.6	23.8	13.0	19.0	10.6	15.7
	CC	*dédiée*	12.4	19.3	10.8	15.0	**9.6**	**13.4**
		commune	12.4	19.3	11.2	15.4	9.9	13.6
LCP	HAC	*dédiée*	8.5	**19.5**	9.6	19.7	**8.3**	**18.1**
		commune	8.5	19.5	9.9	20.6	10.0	19.1
	CC	*dédiée*	8.5	22.6	8.9	22.6	8.4	20.1
		commune	8.5	22.6	9.3	22.9	8.7	21.2

TABLE 4.3: Résumé des performances intra- et inter-document des systèmes *baseline*.

On note également que dans le meilleur des cas, le DER intra-document atteint 9.6% sur BFM et 8.3% sur LCP. A la section 4.2.1.2, nous avions quantifié le DER intra-document minimum atteignable à 7.8% sur BFM et 5.9% sur LCP. Au vu de ces résultats, on peut constater que l'écart entre les performances du meilleur système et le meilleur DER atteignable est faible puisque d'environ 2 points. Par ailleurs, dans toutes les configurations, on observe une dégradation plus importante entre les performances intra- et inter-document (dans le meilleur cas de 3.7 points pour BFM et de 9.1 points pour LCP). Ceci nous amène à deux constats :

- Sur chaque collection, plus de 70% du meilleur DER intra-document est constitué d'erreur irréductible, due à des briques de traitement antérieures au regroupement *i-vector* intra-document. Il reste donc des progrès à faire sur ces briques de pré-traitement.
- L'écart important entre performances intra- et inter-document montre que la compensation de la variabilité inter-document n'est pas parfaite et qu'elle est améliorable.

Dans les chapitres suivants, nous nous consacrerons au deuxième point, à savoir l'amélioration du regroupement inter-document, par deux méthodes différentes : la compensation neuronale de variabilité ou l'adaptation au domaine.

Lien avec des travaux antérieurs Les DER inter-document renseignés dans la table 4.3 sont légèrement plus élevés de ceux obtenus par [Dupuy, 2015], dont la thèse a porté sur la SRL de collection, sur des corpus approchant et des systèmes semblables. Ainsi, sur l'émission *BFM Story*, l'auteur obtient dans la meilleure configuration un DER de 13.9% avec un système ILP/PLDA à regroupement global (14.1% avec un système HAC/PLDA). Sur l'émission *LCP Info*, il obtient dans le meilleur cas 17.7% avec le système HAC/PLDA et 19.8% avec le système ILP/PLDA.

La différence constatée (près de 2 points sur le corpus BFM en configuration

HAC/PLDA) s'explique probablement par le fait que dans ce manuscrit, nous avons choisi, pour des raisons qui seront détaillées au chapitre 6, d'apprendre nos modèles (UBM,TV,WCCN et PLDA) sur des données issues d'émissions radiophoniques uniquement, tandis que les données d'évaluation sont issues d'émissions télévisées. Dans les travaux de [Dupuy, 2015], une partie des données d'apprentissage est constituée d'émissions télévisées, en particulier de quelques épisodes de l'émission *BFM Story*, ce qui justifie l'écart observé sur BFM.

4.2.2.3 DER et seuil de regroupement

Maintenant que les performances des différents systèmes *baseline* sont connues, nous vérifions la significativité du pas de seuil choisi pour les expériences (de 10 en 10). Exceptionnellement, nous nous permettons de calculer le DER inter-document à chaque regroupement pour le HAC, et aux seuils correspondants pour le CC. Ceci va nous permettre d'avoir une idée de l'écart entre le minimum réel de DER et le minimum estimé, lorsqu'on échantillonne la mesure. On pourra également avoir une idée de la dynamique du DER au fil des regroupements : le DER décroit-il de façon continue jusqu'à son minimum pour remonter de façon continue, ou existe-t-il des minima locaux ?

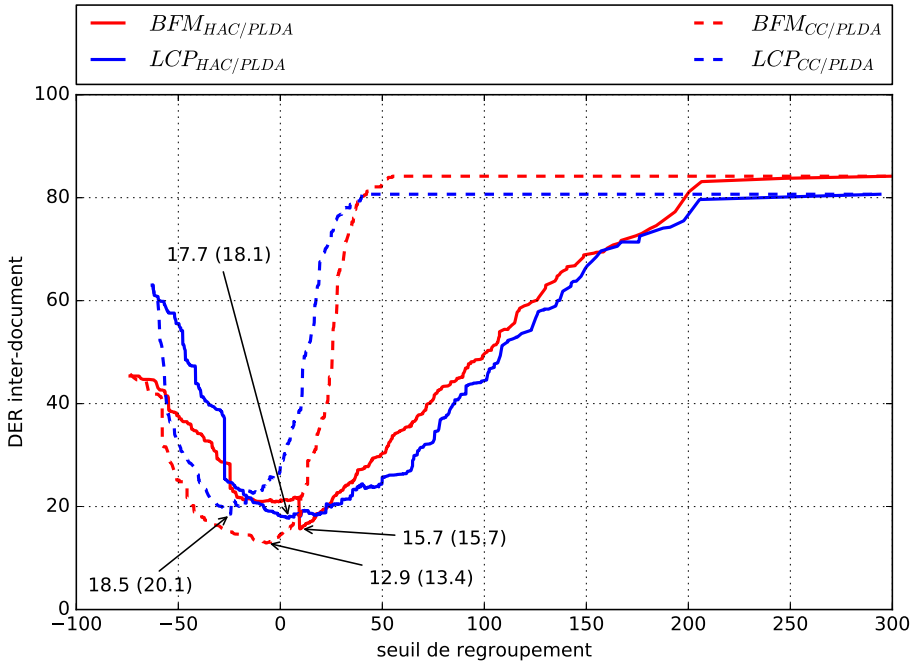

FIGURE 4.2 – Evolution du DER inter-document, regroupement par regroupement, dans les configurations HAC/PLDA et CC/PLDA pour les deux collections cibles. Les valeurs affichées sont les valeurs minimales réelles suivies, entre parenthèses, des valeurs minimales estimées avec un pas de seuil de 10.

Sur la figure 4.2, nous avons représenté le DER inter-document en fonction du seuil de regroupement, pour les configuration HAC/PLDA et CC/PLDA, sur les deux collections. Pour chaque courbe, nous affichons le minimum réel de DER, suivi, entre parenthèses, du minimum estimé avec un pas de seuil de 10. On constate dans le pire cas un écart de 1.6 points (CC/PLDA sur LCP) et dans le meilleur cas un écart nul (HAC/PLDA sur BFM). On note également que l'intervalle de seuils utilisables est plus faible avec le regroupement CC qu'avec le regroupement HAC. En effet, le regroupement CC étant associatif, dès que de mauvais regroupements sont effectués, le taux d'erreur peut très vite augmenter (sur la figure, dès que le seuil est supérieur à 0). Le regroupement HAC, conservatif, est plus robuste au choix du seuil. Pour chacune des deux collections, avec ce regroupement, on remarque de fortes ruptures de pente, qui traduisent la fusion de deux classes-locuteurs importantes.

4.3 Analyse en locuteurs

Dans la problématique de la structuration et l'indexation des collections, se pose la question de l'existence ou non de locuteurs plus importants que d'autres, pour lesquels une erreur serait plus pénalisante. Dans la section 2.2.1.1, nous avons mis en évidence des distributions de temps de parole variables selon le rôle des locuteurs. Or, dans le cadre de la structuration de collections, selon le contexte applicatif, il peut être pertinent que le système sache bien étiqueter les prises de parole d'un homme politique, plutôt que les prises de paroles du présentateur d'une émission, ou l'inverse. Or, si le présentateur d'une émission parle pour 50% du temps de parole total, la formule de calcul du DER montre que le taux d'erreur est très dépendant des performances du système pour ce locuteur particulier.

Une attention particulière doit donc être accordée aux locuteurs ayant un temps de parole plus faible, mais dont le rôle dans la structuration de la collection est important. Typiquement, il peut s'agir de locuteurs récurrents rentrant dans la catégorie des invités. Or, les durées de prise de parole des locuteurs invités sont en moyenne plus faibles que pour les journalistes, le DER ne peut donc pas bien refléter les performances sur ces locuteurs importants.

Dans cette section, nous proposons d'étudier les performances des systèmes *baseline*, aux dimensions de 256 pour l'UBM, 200 pour la matrice TV et 100 pour la PLDA, en nous intéressant aux impacts du regroupement sur les différents types de locuteurs. Pour ce faire, nous proposons d'analyser l'appariement entre locuteurs hypothèses et locuteurs de référence, effectué lors du calcul du DER. Nous définissons d'abord deux métriques, le taux d'erreur nominal et le taux d'erreur classe, avant de passer à une analyse détaillée.

4.3.1 Définition des métriques

Dans la table suivante (4.4), nous rappelons certaines statistiques concernant les locuteurs des deux collections, détaillées à la section 2.2. Les locuteurs sont divisés en quatre classes, selon deux caractéristiques : journaliste ou invité, ponctuel ou récurrent. Dans la collection BFM, le temps de parole total est équitablement réparti entre les invités et journalistes, tandis que le temps de parole moyen par épisode des locuteurs récurrents, à environ 3 minutes, est deux fois supérieur au temps de parole des ponctuels, qui est de l'ordre d'1m30s. Concernant la collection LCP, le temps de parole total est aussi réparti de manière équilibrée entre journalistes et invités. Cependant, si le temps de parole moyen par épisode des journalistes récurrents est toujours le double de celui des locuteurs ponctuels, les invités récurrents parlent en moyenne 37 secondes par épisode, une durée relativement courte.

		journalistes ponctuels	invités ponctuels	journalistes récurrents	invités récurrents
BFM	#locuteurs	48	297	42	35
	temps de parole moyen	80s	90s	186s	184s
	%temps de parole total	5.4%	39.2%	45.8%	9.6%
LCP	#locuteurs	5	126	24	73
	temps de parole moyen	62s	52s	100s	37s
	%temps de parole total	0.9%	18.3%	51.2%	29.7%

TABLE 4.4: Statistiques des différentes classes de locuteurs dans les deux collections.

Pour effectuer l'analyse en locuteurs, nous proposons de définir deux métriques : le taux d'erreur nominal et le taux d'erreur classe.

4.3.1.1 Taux d'erreur nominal

Pour calculer le DER, dont nous avons donné la définition à la section 3.5.1, la première étape de l'outil d'évaluation est l'estimation d'un appariement optimal entre locuteurs de références et classes-locuteurs hypothèses. Ensuite, le calcul tient compte de la parole manquée, faussement détectée ou attribuée à un mauvais locuteur, pour tous les locuteurs de référence de la collection.

Lors de l'appariement, chaque classe-locuteur ne peut-être appariée qu'à un locuteur de référence (identifié par son nom et prénom). Ainsi, si le processus de regroupement génère moins de classes-locuteurs qu'il existe de locuteurs de référence dans la collection, certains locuteurs vont être "laissés pour compte". Cet appariement permet d'analyser en détails, locuteur par locuteur, la répartition du temps de parole : un locuteur de référence A est-il associé à une classe hypothèse ? Son temps de parole est-il réparti entre plusieurs classes ? Le cas échéant, est-il pour autant le locuteur à qui la classe est appariée ?

Nous proposons donc de quantifier pour chaque locuteur de référence, dans quelle mesure celui-ci contribue au DER. On définit le taux d'erreur nominal d'un locuteur

comme sa proportion de temps de parole qui contribue au DER, relativement à sa durée totale de parole dans la référence. C'est donc une mesure proche de la couverture, qui correspond à :

$$taux^{err}_{nominale} = 1 - couverture_{nominale} \tag{4.3}$$

Ainsi, si un locuteur A est associé à une classe, mais qu'une partie de son temps de parole appartient à une autre classe, ce temps de parole est décompté comme de l'erreur d'attribution. En revanche, si aucune classe-locuteur ne lui est appariée, l'intégralité de son temps de parole contribue au DER, son taux d'erreur nominal est alors de 100%. A l'aide de cette métrique, on peut étudier, pour l'ensemble des locuteurs d'un type particulier (par exemple, les invités récurrents), la répartition des taux d'erreur nominaux. En fonction des besoins applicatifs, on peut vouloir ne rater aucun locuteur, c'est-à-dire essayer d'apparier tous les locuteurs de référence, même si cela ne correspond pas au DER optimal.

4.3.1.2 Taux d'erreur classe

Le taux d'erreur classe suit la même logique, mais au niveau d'une classe (ou type) de locuteurs (par exemple les invités ponctuels). Cela correspond à la somme des erreurs des locuteurs de la classe sur la durée de parole totale de la classe. Dans les collections cibles que nous traitons, nous avons défini quatre classes de locuteurs, l'idée est donc de calculer quatre taux d'erreurs pour analyser la performance de SRL de collection. Ceci nous permet d'analyser les performances du système selon les types de locuteurs. Par exemple, lors des regroupements inter-document, les locuteurs ponctuels ne doivent pas être regroupés : par définition, il ne parlent que dans un seul document. Analyser leur taux d'erreur classe permet donc de quantifier dans quelle mesure le regroupement se fait au détriment des locuteurs ponctuels.

Soit une classe $\mathcal{C}$ contenant c locuteurs i, de durée totale d_i^{tot}, dont une partie est de la durée d'erreur d_i^{err} (car mal appariée ou non appariée du tout), on définit le taux d'erreur classe :

$$taux^{err}_{\mathcal{C}} = 1 - couverture^{moyenne}_{\mathcal{C}} = \frac{\sum_i^c d_i^{err}}{\sum_i^c d_i^{tot}} \tag{4.4}$$

4.3.2 Ecart de performances intra-/inter-document

Commençons par étudier l'effet du regroupement inter-document, en le comparant avec les performances intra-document. Par exemple, dans la configuration *dédiée* sur la collection BFM, avec la similarité PLDA et un regroupement HAC, on obtient un DER intra-document de 10.6% mais un DER inter-document de 15.7%, comme indiqué dans la table 4.3. Sur la collection LCP, le DER intra-document

est à 8.3% contre 18.1% en inter-document. Etudions les performances sur ces deux collections en taux d'erreur nominal et taux d'erreur classe.

4.3.2.1 Cas de la collection BFM

Nous avons vu précédemment que nous sommes capables, pour chacun des locuteurs de la référence, de quantifier dans quelle mesure il contribue au DER. La figure 4.3 présente une analyse différenciée par type de locuteur, avant et après regroupement inter-document. Les graphes de la première colonne concernent le regroupement intra-document, ceux de la seconde le regroupement inter-document, tandis que chaque ligne correspond à un type de locuteur.

Chaque graphe est un histogramme où une barre représente le taux d'erreur nominal d'un locuteur. Le degré de coloration de chaque barre de l'histogramme est fonction de la durée totale de parole du locuteur dans la collection. Ainsi, chez les journalistes récurrents (graphes de la $3^{ème}$ ligne), il existe une seule barre très colorée : elle représente le locuteur principal de l'émission. La méthode de représentation permet donc d'avoir une idée des contributions réelles au DER : un locuteur qui parle 10 minutes au total dans la collection (barre foncée) et qui présente un taux d'erreur de 20% contribuera plus au DER qu'un locuteur qui parle 1 minute (barre claire) avec un taux d'erreur de 100%. Elle permet également d'étudier nominalement les performances et par exemple de quantifier le nombre de locuteurs affichant un taux d'erreur de 100%. En fonction des besoins applicatifs, on peut vouloir ne rater aucun locuteur, même si cela ne correspond pas au DER optimal.

Dans chaque graphe, les locuteurs sont triés par taux d'erreur nominal croissant : l'index d'un locuteur dans un graphe de la première colonne n'est pas le même dans celui de la seconde. Pour aider à la lecture, nous renseignons pour chaque graphe (ie. type de locuteur) la moyenne et la médiane des taux d'erreur nominaux, ainsi que le taux d'erreur de la classe (ie. la somme des erreurs des locuteurs de la classe sur la durée de parole totale de la classe). Ainsi, le premier graphe (première ligne, première colonne) concerne les taux d'erreur nominaux des journalistes ponctuels en évaluation intra-document, on y lit que la médiane est à 0.3%, la moyenne à 5.7% et l'erreur de la classe à 3.5%.

Rappelons que dans le cas du calcul du DER intra-document, l'appariement est effectué pour chaque document séparément. C'est seulement parce qu'on connait l'identité des locuteurs de référence à travers la collection qu'on peut calculer les taux d'erreur nominaux sur l'ensemble de la collection. Contrairement à l'appariement pour le calcul du DER inter-document, il faut décompter les erreurs document par document, pour chaque locuteur.

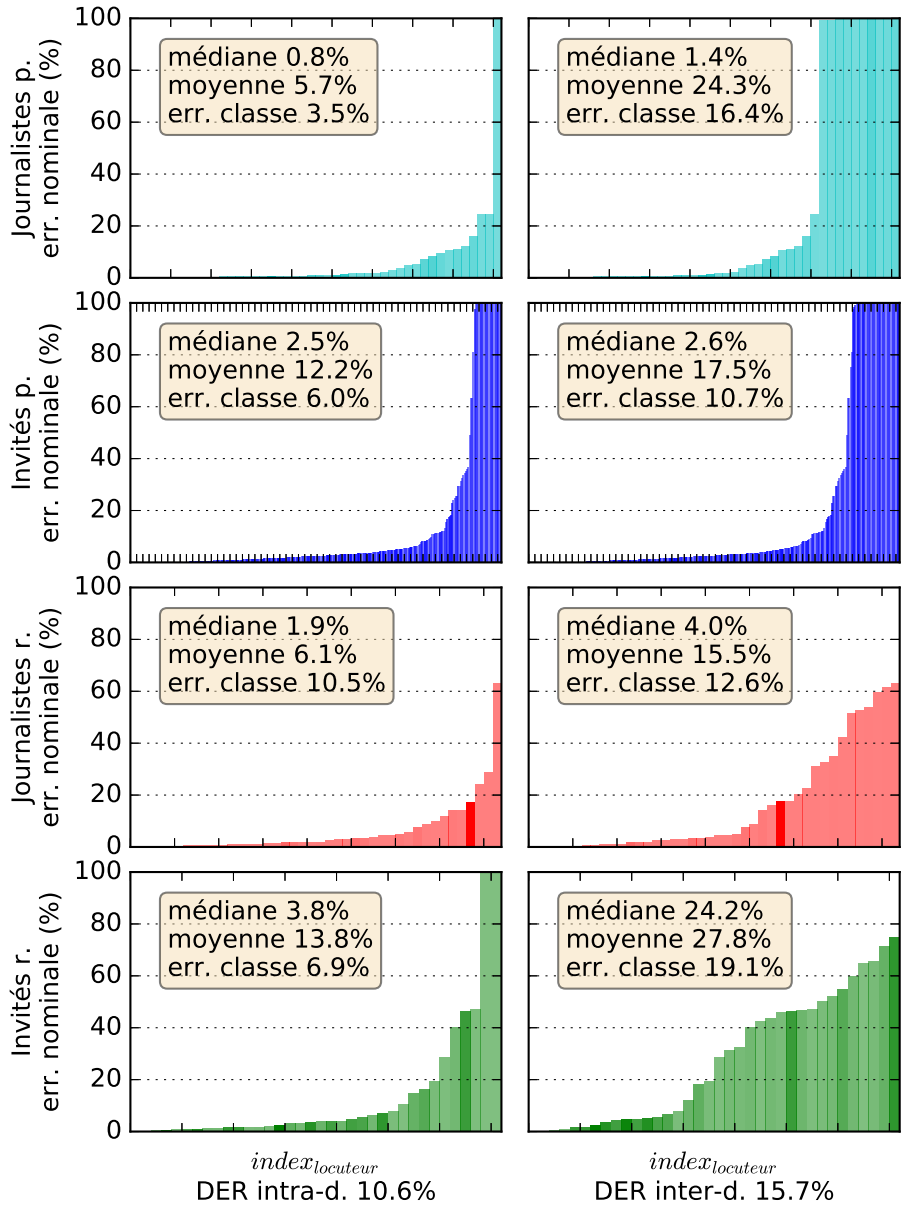

FIGURE 4.3 – Analyse par type de locuteur des différences de taux d'erreur nominaux entre les regroupements intra- et inter-document pour la collection BFM, avec le système HAC/PLDA *baseline*. Les locuteurs peuvent être **ponctuels** (p.) ou **récurrents** (r.).

Du côté des performances intra-document (colonne de gauche), les performances sont assez homogènes entre les classes, avec une médiane variant de 0.8% pour les journalistes ponctuels à 3.8% pour les invités récurrents. Dans chaque graphe, on constate cependant en queue de distribution quelques locuteurs affichant un taux d'erreur de 100% (non appariés). Sachant que la collection contient 310 invités ponctuels, on en dénombre tout de même une trentaine qui n'ont pas été appariés. Ceci s'explique par la distribution du temps de parole de la classe. En effet, nous avons vu à la section 2.2 qu'elle était bi-modale, différenciant les individus intervenant sur un temps très court (micro-trottoir) de ceux participant à échange plus long (débat).

Or ici, sur la trentaine d'invités ponctuels non appariés, on en dénombre seulement 4 avec un temps de parole supérieur à 30 secondes. Concernant la métrique erreur par classe, la classe au taux d'erreur le plus élevé est celle des journalistes récurrents avec 10.5%.

Lorsqu'on passe au regroupement inter-document, on observe des performances assez stables chez les invités ponctuels et les journalistes récurrents, avec une augmentation du taux d'erreur par classe de 6.0% à 10.7% et de 10.5% à 12.6% respectivement, les médianes ne dépassant pas les 4.0%. Chez les journalistes ponctuels, en revanche, une dizaine de locuteurs voient leur taux nominal passer à 100% d'erreur, ce qui multiplie le taux d'erreur classe par 5, à 16.4%. Ces locuteurs ont été regroupés avec des locuteurs plus importants en inter-document. Pourtant, leur temps de parole moyen se situe autour de la minute. Enfin, pour les invités récurrents, on observe une augmentation importante de la médiane, et donc un triplement du taux d'erreur classe à 19.1%. Les invités récurrents sont ceux pour lesquels la variabilité inter-document est la plus forte : ils peuvent être interviewés à différents moments dans des environnements variés. Lorsqu'on compare avec les journalistes récurrents, la variation entre les performances intra- et inter-document est moins importante, car les conditions acoustiques sont en général plus stables d'un épisode à l'autre.

Dans la collection BFM, les deux classes prépondérantes (invités ponctuels et journalistes récurrents) représentent 85% du temps de parole total. Or ce sont celles pour lesquelles l'augmentation de l'erreur classe est la plus limitée. Ceci permet d'expliquer l'augmentation du DER d'environ 5 points entre les regroupements intra- et inter-document, là où pour LCP on constate une dégradation de 10 points dans la même configuration.

4.3.2.2 Cas de la collection LCP

Intéressons nous maintenant à la collection LCP, toujours avec la similarité PLDA et le regroupement HAC. La représentation est la même, présentée à la figure 4.4. Concernant le regroupement intra-document (histogrammes de gauche), on constate chez les journalistes, qu'ils soient récurrents ou ponctuels, des performances correctes, avec un taux d'erreur de classe autour de 6%. Du côté des invités, les performances sont plus disparates selon les individus. Chez les invités ponctuels, on constate une médiane à 2.3% et les performances sont excellentes pour 80% des locuteurs. En revanche, la queue de distribution montre pour une quinzaine de locuteurs un taux d'erreur nominal de 100%, d'où l'erreur nominale moyenne à 17.6%. Quand on regarde le détail des résultats, on constate que parmi ces 15 locuteurs, aucun ne parle plus de 10 secondes. Du côté des invités récurrents, le phénomène est semblable, avec une médiane légèrement plus élevée à 5.7%. On note une dégradation progressive des performances pour le dernier quart de locuteurs, parmi lesquels 2 affichent un taux d'erreur de 100%. Ce sont des locuteurs qui parlent très peu

(moins de 5 secondes chacun). L'erreur de la classe est alors à 9.3% et la moyenne à 19.2%.

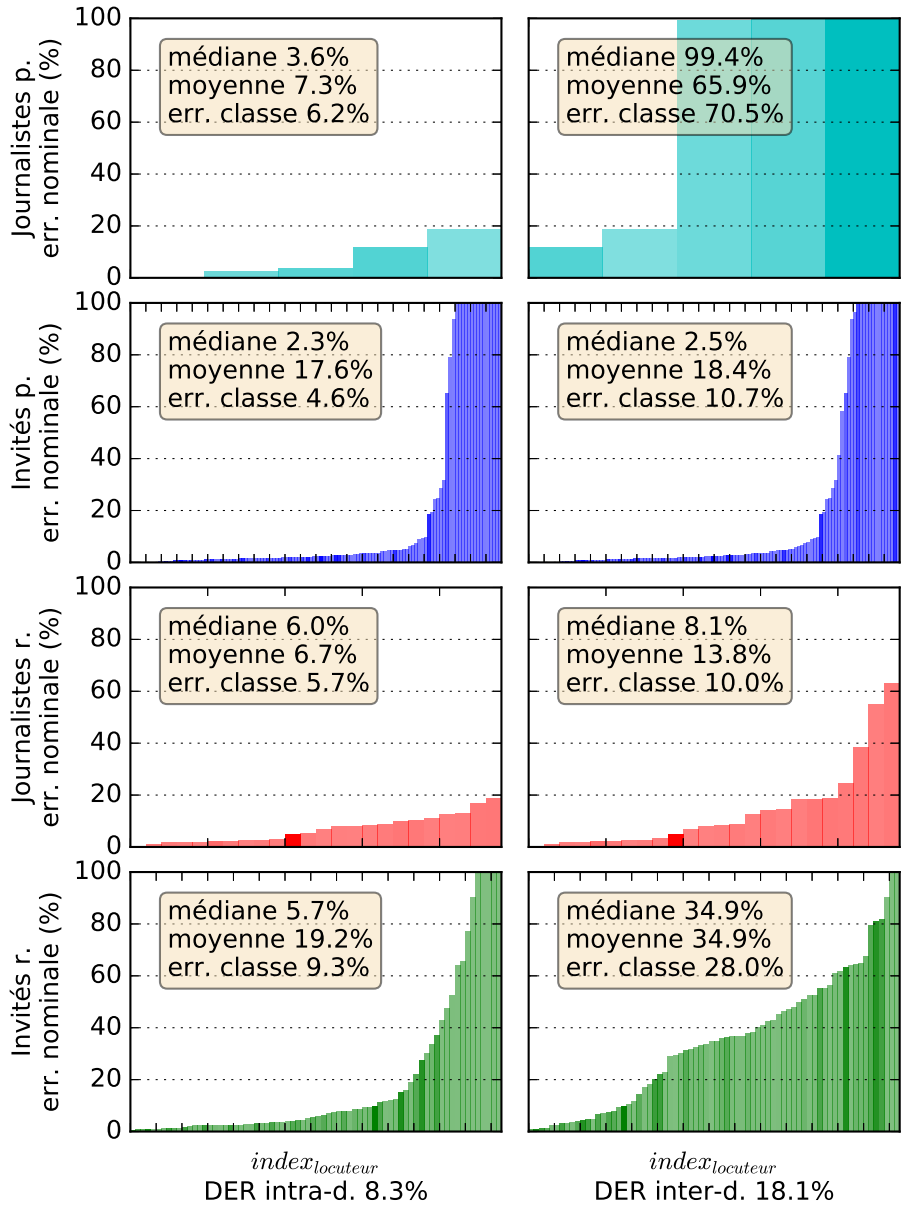

FIGURE 4.4 – Analyse par type de locuteur des différences de taux d'erreur nominaux entre les regroupements intra- et inter-document pour la collection LCP, avec le système HAC/PLDA *baseline*. Les locuteurs peuvent être **ponctuels** (p.) ou **récurrents** (r.).

Lorsqu'on compare avec les performances inter-document (colonne de droite), on observe des comportements différents selon les types de locuteurs. Chez les journalistes ponctuels (5 individus seulement), on observe une dégradation significative avec trois locuteurs passant à 100% d'erreur, leur temps de parole moyen dépassant la minute. Du côté des invités ponctuels et journalistes récurrents, les performances varient peu quand on regarde l'allure des histogrammes, avec une médiane qui augmente légèrement. En revanche, le taux d'erreur de chaque classe est environ doublé :

les locuteurs pour lesquels le taux d'erreur augmente représentent un temps de parole important, ce qui impacte plus fortement la métrique. Par exemple, parmi les invités ponctuels, on compte 2 locuteurs parlant plus de 2 minutes qui n'ont pas été appariés.

La plus grosse perte de performances se situe au niveau des invités récurrents, où on observe un triplement du taux d'erreur classe, à 28.0%, et où la médiane passe de 5.7% à 34.9%. On observe pour une grande partie des locuteurs une dégradation. En particulier, on remarque un certain nombre de locuteurs importants (barres foncées) pour lesquels le taux d'erreur nominal avoisine les 80%. Ces mauvais résultats chez les invités récurrents sont à mettre en relation avec leur durée de parole moyenne par épisode : 37 secondes, la plus courte des quatre types de locuteurs. Lorsqu'on compare avec les journalistes récurrents, la variation des performances intra- et inter-document est moins importante et peut s'expliquer par leur durée de parole moyenne par épisode, qui avoisine les 100 secondes.

On sait que la plus grosse proportion du temps de parole total de la collection LCP concerne les journalistes récurrents (environ 50%), puis les invités récurrents (environ 30%). Par conséquent, une dégradation des performances sur ces deux classes de locuteurs peut avoir un effet négatif important sur le DER.

4.3.2.3 Bilan

En résumé, l'enseignement principal de cette étude est que sur les deux collections, pour la tâche de regroupement inter-document, la classe de locuteurs la plus difficile à regrouper est celle des invités récurrents.

4.3.3 Influence du seuil de regroupement

Dans cette section, nous proposons d'étudier l'influence du seuil de regroupement au voisinage du seuil qui optimise le DER inter-document. Nous utiliserons le même type d'analyse détaillée qu'à la section précédente, mais cette fois dans la configuration de regroupement CC, toujours avec la similarité PLDA.

4.3.3.1 Cas de la collection LCP

L'expérience est présentée dans la figure 4.5. Cette fois, la figure comprend trois colonnes, la centrale correspondant au seuil optimal ($\lambda_X = -30$), celle de gauche (respectivement de droite) à un seuil de deux pas inférieur (resp. supérieur) à l'optimal. Visuellement, un premier constat est que plus le seuil est élevé ("plus on regroupe"), plus le nombre de locuteurs dont le taux d'erreur nominal atteint 100% est élevé. Ceci est logique, car à trop regrouper, on génère un nombre de classes-locuteurs inférieur au nombre réel de locuteurs : certains ne peuvent pas être appariés

lors du calcul du DER. A contrario, si le nombre de classes est trop élevé, les locuteurs ne sont pas suffisamment regroupés, et la majorité de leur temps de parole est compté comme de l'erreur. C'est particulièrement visible dans le premier graphe de la troisième ligne, qui concerne les journalistes récurrents. C'est la classe la plus représentée, sur laquelle le regroupement inter-document a beaucoup d'importance.

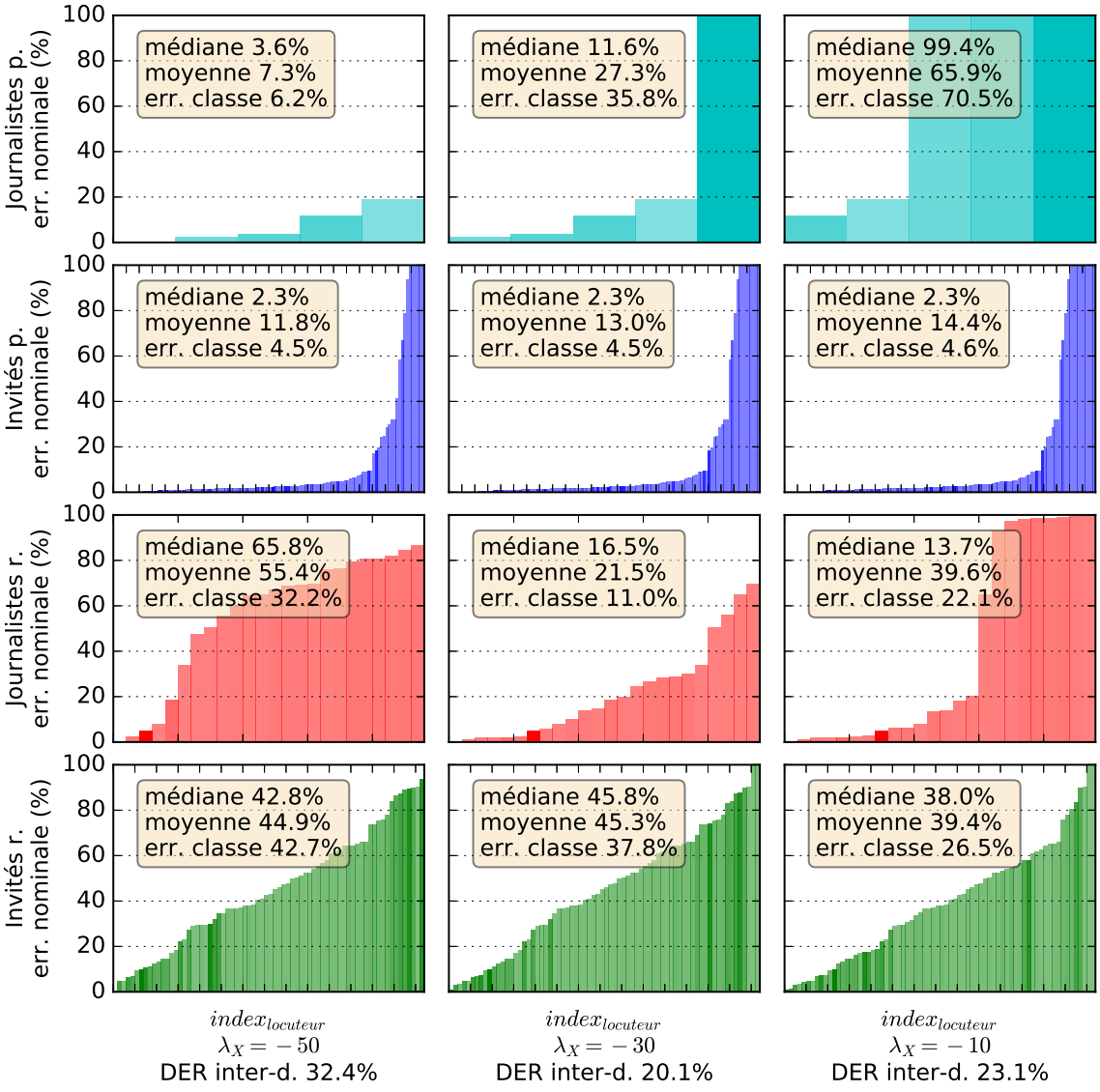

FIGURE 4.5 – Analyse en locuteurs des différences de taux d'erreur nominale selon le seuil de regroupement inter-document, pour la collection LCP, avec le système CC/PLDA *baseline*. Les locuteurs peuvent être **ponctuels** (p.) ou **récurrents** (r.).

Au niveau des performances par classe, on constate pour les locuteurs ponctuels des allures d'histogrammes assez similaires à travers les seuils, la seule différence étant les quelques locuteurs dont le taux d'erreur passe à 100%. Quant au taux d'erreur classe, il augmente à mesure qu'on regroupe. Ceci est logique, les locuteurs ponctuels ne peuvent rien gagner à un regroupement inter-document, puisqu'ils n'apparaissent que dans un épisode. Pour préserver ces locuteurs, il faut donc veiller à

ne pas trop regrouper. Concernant les journalistes récurrents, la classe la plus importante en durée de parole, on observe de grosses variations en fonction du seuil, avec une valeur optimale à 11.0% pour l'erreur classe, obtenue lorsque le seuil de regroupement optimise le DER. Il est logique que les deux soient corrélés, la classe la plus représentative ayant le plus d'impact sur le DER.

Enfin, chez les invités récurrents, l'erreur classe diminue au fil des regroupements, de 42.7% à 26.5%. Le seuil optimal pour cette classe se situe au-delà du seuil qui optimise le DER. L'ensemble des observations montre donc des comportements différents selon les classes et les seuils de regroupement considérés. En fonction du besoin final de la tâche de SRL, le seuil de regroupement optimal peut varier.

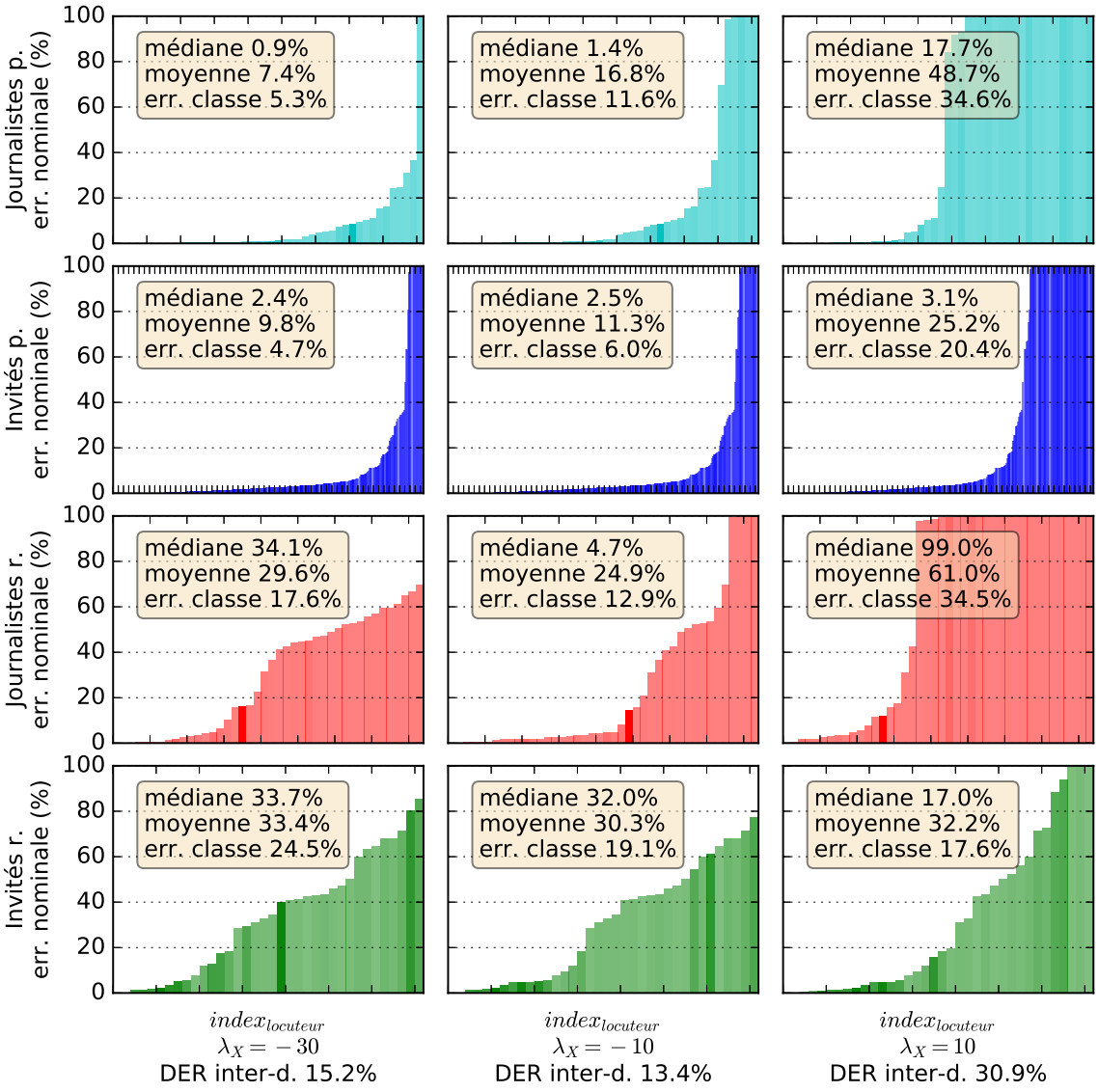

FIGURE 4.6 – Analyse en locuteurs des différences de taux d'erreur nominale selon le seuil de regroupement inter-document, pour la collection BFM. Les locuteurs peuvent être **ponctuels** (p.) ou **récurrents** (r.).

4.3.3.2 Cas de la collection BFM

Concernant la collection BFM, l'expérience est présentée à la figure 4.6, pour le voisinage du seuil optimal $\lambda = -10$. Les observations sont similaires à celles de LCP. Plus le seuil de regroupement est élevé, plus le nombre de locuteurs passant à 100% d'erreur est important. Pour les locuteurs ponctuels, leur taux d'erreur classe augmente à mesure des regroupements, a contrario des locuteurs récurrents. On constate encore une fois que le seuil minimisant le taux d'erreur classe varie selon la classe considérée, et ce dans le même ordre que sur la collection LCP : au premier seuil, l'erreur est minimale pour les locuteurs ponctuels, au suivant, on l'optimise pour les journalistes récurrents, et au troisième, ce sont les invités récurrents qui sont favorisés. Enfin, on sait que les classes les plus représentées sont les invités ponctuels et journalistes récurrents, ce qui explique que le DER est minimal au seuil intermédiaire, où le taux d'erreur classe des premiers est à 6.0% et des seconds à 12.9%.

Quant à l'histogramme des invités récurrents dans le meilleur cas (quatrième ligne, troisième colonne), on observe deux paliers dans la distribution des taux d'erreurs, avec une médiane à 17.0%. Si les trois histogrammes de la quatrième ligne ont des allures assez similaires, le taux d'erreur classe diminue grâce aux gains obtenus sur des locuteurs importants lors des différents regroupements. Pour preuve, la présence de barres foncées avec des taux d'erreur assez élevés dans le premier histogramme, dont la position rétrograde successivement dans les deux histogrammes suivants.

4.3.3.3 Bilan

L'ensemble des observations met en évidence des comportements différents selon les classes et les seuils de regroupement considérés. En fonction du besoin final de la tâche de SRL, le seuil de regroupement optimal peut varier.

4.3.4 Etude comparative des différentes *baseline*

Comme nous avons pu le constater dans la section précédente, optimiser le DER n'optimise pas nécessairement le taux d'erreur sur les différents types de locuteurs. Ainsi, nous proposons de comparer les différents systèmes *baseline* (cosine, cosine/WCCN et PLDA, avec regroupement HAC ou CC) à la configuration *dédiée* de chaque collection, qui optimise le DER. Pour chaque configuration, nous proposons d'étudier les taux d'erreur pour chaque type de locuteur, pour voir si tel ou tel système serait plus adapté à tel type de locuteur.

Le comparatif est présenté à la figure 4.7, sous forme de paquets d'histogrammes. La partie supérieure concerne BFM tandis que l'inférieure concerne LCP. Dans la figure, chaque paquet d'histogrammes correspond à la configuration *dédiée* d'un

des systèmes *baseline*, explicité en abscisse. Chacun d'entre eux contient 5 barres représentant les différents taux d'erreur : une par type de locuteur et la dernière pour le DER. Enfin, la transparence des quatre premières barres est fonction de la proportion du type de locuteur dans le temps de parole total : c'est-à-dire l'influence sur le DER. Par exemple, dans la partie supérieure, les barres les plus foncées sont la bleue et la rouge, qui représentent respectivement les invités ponctuels et les journalistes récurrents, soit 85% du temps de parole total de la collection BFM.

Concernant la collection BFM, on sait que le meilleur système *baseline* est le système CC/PLDA, avec un DER inter-document à 13.4%. Derrière, le second système le plus performant est le HAC/PLDA, avec un DER à 15.7%. Le détail des performances des deux systèmes par type de locuteur montre qu'elles sont quasiment identiques sur les locuteurs récurrents. C'est sur les locuteurs ponctuels que le regroupement HAC montre ses limites. En revanche, pour optimiser le taux d'erreur sur les locuteurs ponctuels, il faut privilégier le système CC/WCCN, où le DER est à 15.0%. On voit donc que pour des DER proches (15.7% contre 15.0%), le comportement des systèmes peut varier sur les différents types de locuteurs.

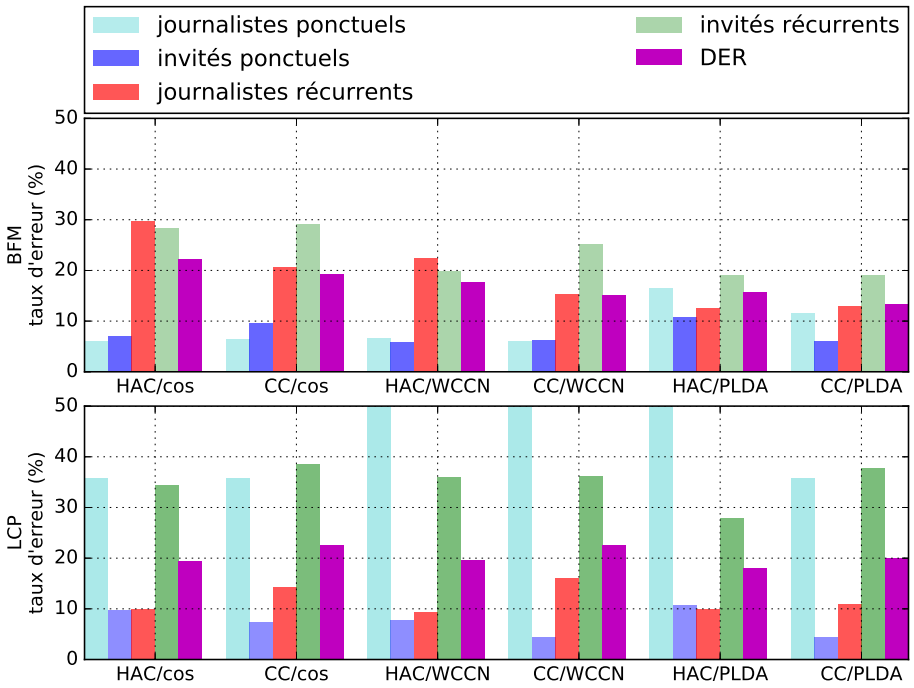

FIGURE 4.7 – Détails des taux d'erreur classe et du DER inter-document de chaque système de SRL *baseline* au DER minimal, pour les deux collections.

La collection LCP affiche de son côté un DER minimal à 18.1% avec le système HAC/PLDA. Pour cette collection, on remarque la variabilité importante des taux d'erreur des journalistes ponctuels, jusqu'à 70.5%, même si leur présence est marginale. Pour le reste, les systèmes donnent des performances assez similaires, même si on peut remarquer que le regroupement CC a tendance à être meilleur sur les invités ponctuels. Globalement, le DER des différents systèmes *baseline* va-

rie peu, les méthodes de compensation de variabilité intra-locuteur/inter-document sont peu efficaces, ce qui explique les taux assez constants chez les différents types de locuteurs.

4.3.5 Sensibilité au seuil, pour chaque classe de locuteurs

Dans les deux sections précédentes (4.3.3 et 4.3.4), nous avons constaté qu'optimisation du DER ne signifiait pas nécessairement optimisation sur tous les types de locuteurs, d'une part, et que des différences existaient entre les différents systèmes *baseline*, d'autre part. Dans cette section, nous proposons d'étudier plus largement la dynamique des différents taux d'erreur en fonction du seuil de regroupement inter-document λ_X, pour les systèmes *baseline* HAC/cosine et CC/cosine, sans normalisation. Les figures représentatives des autres systèmes sont disponibles en annexe B.

4.3.5.1 HAC/Cosine

La figure 4.8 représente l'évolution des taux d'erreur classe en fonction du seuil de regroupement, pour le système HAC/Cosine, sans normalisation. Le DER est aussi illustré par la courbe mauve. Pour ce système de SRL, on observe clairement des optima différents pour chaque type de locuteur. Par définition, les taux d'erreur sur les locuteurs ponctuels sont bas pour des seuils bas. Parfois, on observe une faible diminution, par exemple pour les journalistes ponctuels de BFM, entre les seuils -40 et -30 : il s'agit d'un regroupement intra-document valide (ie. entre deux classes qui représentent le même locuteur), qui a lieu pendant la phase de regroupement inter-document (les regroupements intra-document n'étant pas interdits). Sur la figure, on peut considérer que malgré de faibles variations, les taux d'erreur des locuteurs ponctuels sont minimum pour les seuils inférieurs à $\lambda_X = -20$. Au-delà, les taux ne peuvent qu'augmenter.

Concernant les locuteurs récurrents, la dynamique est différente, les taux d'erreur au seuil le plus bas ($\lambda_X = -90$) sont élevés, ce qui est normal car aucun regroupement inter-document n'a eu lieu. Ensuite, les regroupements successifs font baisser les taux pour atteindre un minimum pour λ_X variant de 20 à 40. Ainsi, le DER est optimal pour $\lambda_X = 0$, sur les deux collections : c'est un compromis entre les deux zones d'optimalité. Il est à 23.8% pour BFM et à 19.5% pour LCP.

4.3.5.2 CC/Cosine

La figure 4.9 présente la dynamique des taux d'erreur pour le système CC/Cosine. Cette fois-ci, les courbes ont la même allure selon les types de locuteurs, les seuils optimaux pour les différents types de locuteurs sont plus rapprochés, voire identiques. Par exemple, pour la collection BFM, le seuil λ_X est quasiment optimal pour

les quatre types de locuteurs, avec un DER de 19.3%. Pour la collection LCP, la classe des journalistes ponctuels, même si elle est peu représentative, voit son taux d'erreur augmenter rapidement, à partir de -60. Si les optima des trois autres classes de locuteurs sont légèrement différents (de -40 à -20 selon la classe), un seuil de -30 donne un DER de 22.6%.

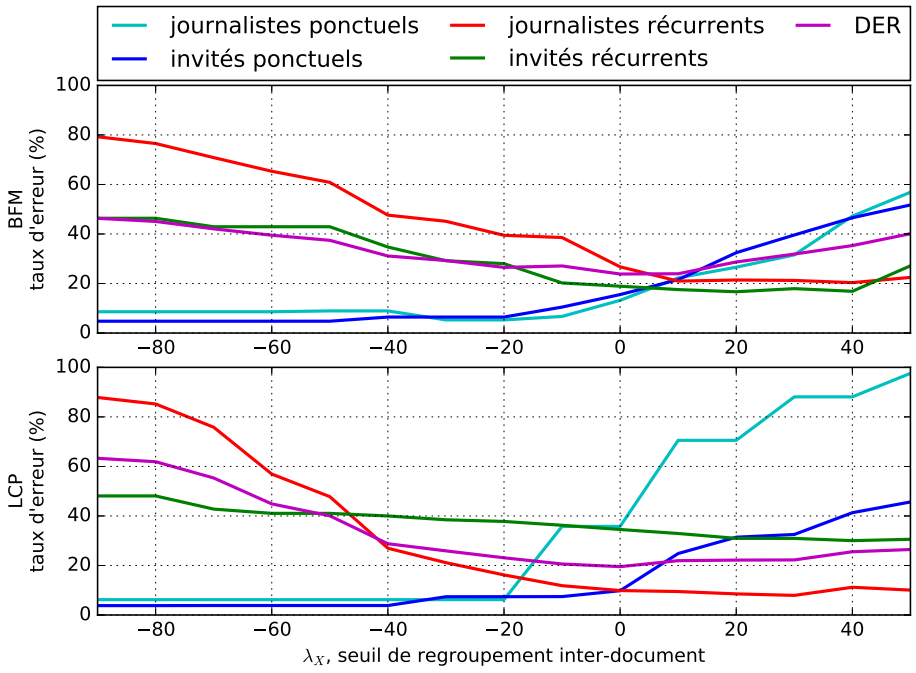

FIGURE 4.8 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement HAC λ_X, avec la similarité cosine seule ($\lambda_I = -10$).

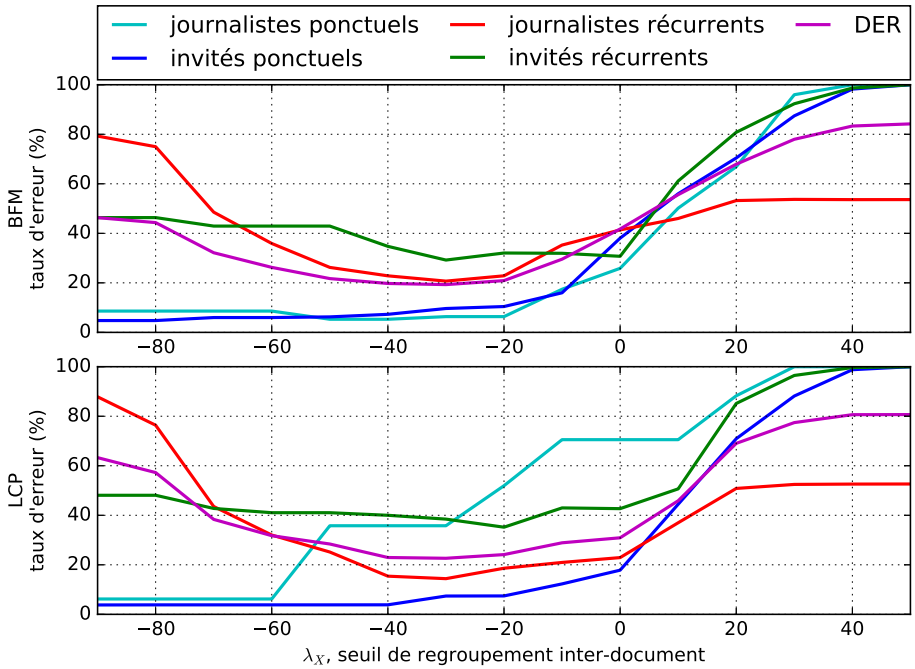

FIGURE 4.9 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement CC λ_X, avec la similarité cosine seule ($\lambda_I = -10$).

Ces deux figures illustrent donc des dynamiques de taux d'erreur différentes selon les types de locuteurs. Idéalement, un regroupement qui optimise le taux d'erreur des quatre classes pour un même seuil favorisera un DER bas. S'il est tentant de conclure que le regroupement CC favorise ce phénomène, des contre-exemples existent, par exemple le système CC/WCCN (figure B.2 de l'annexe B).

4.4 Bilan

Dans ce chapitre, nous avons présenté notre système de SRL à l'état de l'art, à architecture de regroupement global, que nous avons évalué sur deux collections distinctes d'une quarantaine d'épisodes. Les résultats montrent que le système le plus compétitif utilise la PLDA pour compenser la variabilité intra-locuteur/inter-document, que ce soit avec un regroupement HAC ou CC. L'importance de l'erreur irréductible due aux briques de pré-traitement, relativement à l'erreur intra-document totale, indique que des progrès restent à faire au niveau de la détection de parole et de la segmentation en locuteurs. Par ailleurs, les écarts importants entre DER intra- et inter-document nous poussent à étudier de façon plus approfondie la question du regroupement inter-document dans les chapitres suivants.

L'analyse détaillée des performances des différents systèmes proposés, par type de locuteur, révèle des comportements différents selon les types de locuteurs. Ainsi, c'est sur les invités récurrents que le module de regroupement inter-document fait le plus d'erreurs, à cause de la grande variabilité inter-document de ces locuteurs, d'une

part, et de la faible durée de parole par document de certains d'entre eux, d'autre part. Selon le type de locuteur (et indirectement le type d'application envisagé), le point de fonctionnement optimal du système pourra donc varier.

Chapitre 5

Compensation neuronale de variabilité

Résumé

Dans ce chapitre, qui constitue une des contributions principales de la thèse [Le Lan et al., 2017], nous proposons une nouvelle méthode de compensation de variabilité intra-locuteur/inter-document. Dans le système de SRL de collection présenté au chapitre précédent, on remplace les méthodes à l'état de l'art telles que WCCN ou PLDA par une approche basée sur un réseau de neurones pour le calcul des similarités. Le concept clé est l'utilisation de la triplet loss qui permet de concentrer l'apprentissage du réseau sur les exemples les plus difficiles. Le réseau de neurones projette les i-vectors dans un nouvel espace qui optimise la séparation des locuteurs. La calibration de l'apprentissage du réseau de neurones est effectuée sur une tâche de vérification du locuteur, à l'aide des métriques EER et minDCF. Les résultats montrent que la méthode proposée est plus performante que la PLDA, en utilisant un regroupement CC, avec une amélioration relative de 16% du DER inter-document.

5.1 Introduction

Ces dernières années, les réseaux de neurones profonds sont devenus très populaires dans un grand nombre de domaines tels que le traitement de la langue, la reconnaissance d'image ou le traitement de la parole. Au sein de la communauté du traitement de la parole, le *deep-learning* a été récemment utilisé pour des tâches telles que la transcription automatique (e.g. [Dahl et al., 2012]), la vérification de locuteurs (e.g. [Richardson et al., 2015]) ou la segmentation en locuteurs (e.g. [Bredin, 2016]). Pour la transcription de la parole (e.g. [Dahl et al., 2012]), les réseaux de neurones profonds sont principalement utilisés pour prédire des triphones (séquences de

trois phonèmes) et répondent donc à une problématique de classification (on cherche à prédire un triphone parmi plusieurs milliers possibles). Les séquences de triphones prédits sont alors utilisés pour décoder ce qui a été prononcé. Par la suite, on s'est aperçu que la capacité des réseaux de neurones à prédire les triphones traduisait leur capacité à représenter l'information phonétique. C'est ce qui a inspiré les *Bottleneck Features* (e.g. [Richardson et al., 2015]), utilisés pour la vérification du locuteur. Ils sont alors utilisés en remplacement ou en complément des coefficients MFCC. Plus récemment, des réseaux de neurones à base de cellules LSTM (pour *Long Short Term Memory*) ont été utilisés pour la segmentation en locuteurs [Bredin, 2016], en remplacement des approches classiques à base de modèles gaussiens exploitant la mesure BIC ou GLR. Les travaux présentés dans ce chapitre, effectués en fin de thèse, s'inspirent de l'approche proposée par [Bredin, 2016], mais pour répondre au problème du regroupement en locuteurs dans la tâche de SRL.

Dans ce chapitre, nous proposons une nouvelle méthode de calcul de similarités entre *i-vectors* pour le regroupement en locuteurs, en remplaçant les méthodes classiques telles que la similarité cosine ou la PLDA par une approche basée sur un réseau de neurones. Des approches neuronales s'appuyant directement sur la représentation *i-vector* ont été proposées récemment [Bhattacharya et al., 2016] et ont montré qu'elles pouvaient égaler la PLDA sur la tâche de vérification du locuteur. Dans cet article, les auteurs entraînent un réseau de neurones à classer des *i-vectors* selon le locuteur qu'ils représentent, dans le but de compenser la variabilité intra-locuteur. Notre approche a le même but, mais n'utilise pas la même stratégie. Elle s'inspire de [Schroff et al., 2015] et [Bredin, 2016], qui ont proposé l'utilisation de plongements neuronaux optimisés pour la reconnaissance et le regroupement de visages ou de locuteurs, en utilisant la *triplet loss* [Wang et al., 2014] pour l'apprentissage. La principale différence de notre approche est qu'elle utilise les *i-vectors*, plutôt que le signal brut (image ou coefficients cepstraux). C'est pourquoi nous la voyons comme une alternative aux méthodes conventionnelles de calcul de similarités pour les *i-vectors*, de la même manière que les auteurs de [Bhattacharya et al., 2016].

5.2 Définition de la méthode neuronale

Dans notre système de SRL, dont on rappelle l'architecture à la figure 5.1, la méthode neuronale proposée prend la place de la brique compensation de variabilité intra-locuteur/inter-document (intitulée "variabilité intra-locuteur" sur la figure 5.1). Elle a notamment pour but de compenser la variabilité intra-locuteur, à la manière de la PLDA. L'idée est de projeter de façon non linéaire les *i-vectors* sur une sphère qui sépare plus efficacement les classes-locuteurs du point de vue de la similarité cosine. Pour ce faire, on entraîne un réseau de neurones f de type *feed-forward*,

optimisé pour ce problème grâce au paradigme *triplet loss*. Par cette nouvelle méthode, la similarité entre deux *i-vectors* (ϕ_1, ϕ_2) correspond à la similarité cosine entre les deux projections $(f(\phi_1), f(\phi_2))$. Dans la suite du document, nous appellerons les similarités générés par cette approche similarités ou scores TR.

Soit un jeu (ϕ_i) de *i-vectors* d'apprentissage représentant différents locuteurs, on échantillonne des triplets (ϕ_a, ϕ_p, ϕ_n) tels que ϕ_a (appelé *anchor*) et ϕ_p (appelé *positive*) représentent le même locuteur et ϕ_n (appelé *negative*) un locuteur différent. La *triplet loss* vise à mieux séparer les classes-locuteurs dans l'espace de projection en maximisant la similarité *anchor-positive*, tout en minimisant la similarité *anchor-negative*. Pour le jeu de tous les N triplets possibles $\mathcal{T} = (\phi_a^i, \phi_p^i, \phi_n^i)_{i\in[1..N]}$, la *loss* est définie par :

$$\mathcal{L}(\mathcal{T}) = \Sigma_i^N max(0, \Delta_i + \beta) \tag{5.1}$$

$$\Delta_i = -\frac{f(\phi_a^i)f(\phi_p^i)^T}{\|f(\phi_a^i)\|\|f(\phi_p^i)\|} + \frac{f(\phi_a^i)f(\phi_n^i)^T}{\|f(\phi_a^i)\|\|f(\phi_n^i)\|} \tag{5.2}$$

β est une marge visant à forcer une séparation des classes-locuteurs. Idéalement, on souhaite qu'à la fin de l'apprentissage, pour tout triplet i, $\Delta_i + \beta < 0$. Pour optimiser l'apprentissage, il est plus rapide de ne choisir que des triplets qui contribuent à la fonction de *loss*, c'est-à-dire des triplets où le *negative* est plus proche de l'*anchor* que le *positive*, à β près. Dans l'article original [Schroff et al., 2015], deux stratégies de sélection de triplets sont comparées : *hard-selection* consiste à choisir tous les triplets contribuant à la loss (ie. $0 < \Delta_i + \beta$), tandis que *soft-selection* consiste à exclure les triplets les plus difficiles (ie. on ne choisit que ceux situés dans la marge, tels que $0 < \Delta_i + \beta < \beta$).

A chaque itération d'apprentissage (ou époque), on propose la stratégie suivante, similaire à celle de [Bredin, 2016] et [Schroff et al., 2015], mais adaptée à l'utilisation des *i-vectors* :

- Pour chaque classe-locuteur qui comprend au moins 3 *i-vectors*, on choisit aléatoirement des paires de *i-vectors* (ϕ_a^i, ϕ_p^i). Le choix des classes à 3 *i-vectors* ou plus est identique à la stratégie employée pour l'apprentissage de la PLDA.
- Parmi les k plus proches voisins (k-PP) de chaque *anchor* $f(\phi_a^i)$, un *negative* ϕ_n^i est tiré aléatoirement parmi ceux qui satisfont à la contrainte $0 < \Delta_i + \beta$ ($< \beta$, selon la stratégie de sélection).
- Ensuite, tous les triplets générés sont utilisés pour mettre à jour les poids et gradients du réseau de neurones.

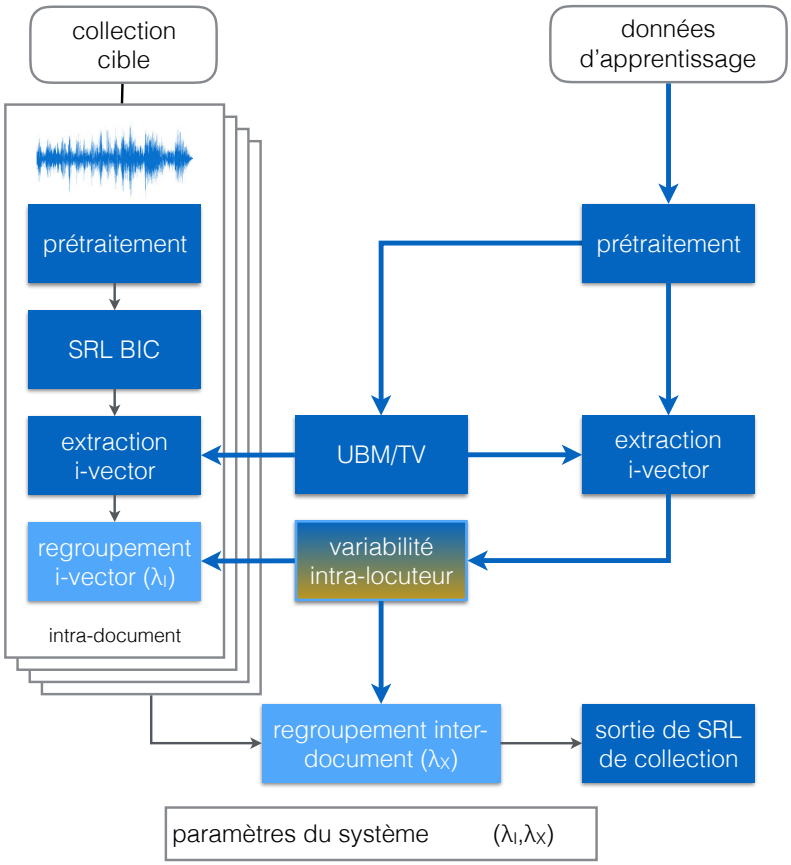

FIGURE 5.1 – Schéma du système de SRL de collection *baseline*. La méthode proposée consiste à remplacer la brique de compensation de variabilité inter-locuteur/inter-document.

5.3 Apprentissage du réseau de neurones

5.3.1 Protocole

Avant d'évaluer la méthode de calcul de similarité TR sur la tâche de segmentation et regroupement en locuteurs, on la teste d'abord sur une tâche de vérification du locuteur, en utilisant les *i-vectors* de référence, c'est-à-dire extraits à partir de la segmentation de référence des collections. Sur l'ensemble de ces *i-vectors* de référence, on calcule le score de similarité de toutes les paires possibles, et on prend une décision sur le fait qu'une paire corresponde au même locuteur ou non, selon que la similarité est supérieure ou inférieure à un seuil d'acceptation/rejet λ. Comme ce sont les *i-vectors* de référence, on connait l'identité des locuteurs représentés par chaque *i-vector* et on peut évaluer la qualité du système.

Les métriques utilisées sont le taux d'égale erreur (*Equal Error Rate* ou EER) et le minimum de la fonction de coût (*minimum Detection Cost Function* (minDCF), avec un a priori de 1%. Ce sont les métriques usuelles dans la communauté (e.g. [Van Leeuwen and Brümmer, 2007]). Comme la tâche de vérification du locuteur est

une tâche classique de classification, on peut explorer plus rapidement les différentes configurations possibles du réseau de neurones que dans le cadre de l'évaluation d'un système de SRL.

5.3.1.1 Taux d'égale erreur

La tâche de vérification du locuteur est une tâche de classification ou le système cherche à estimer si deux *i-vectors* caractérisent le même locuteur ou non. Dans cette configuration, on distingue deux types d'erreurs :

- La fausse acceptation (ou fausse alarme), quand le système estime que les deux *i-vectors* correspondent à un même locuteur alors que ce n'est pas le cas.
- Le faux rejet (ou manqué), lorsque le système estime que les deux *i-vectors* correspondent à deux locuteurs différents alors qu'il s'agit en réalité d'un même locuteur.

Un tel système peut alors être évalué pour une configuration donnée (fonction d'un seuil d'acceptation/rejet λ), par le taux de fausse acceptation et le taux de faux rejet. Plus on essaie de minimiser le taux d'erreur d'un type donné, plus l'autre taux d'erreur risque d'augmenter. Pour caractériser un tel système, on peut utiliser le taux d'égale erreur (*Equal Error Rate* ou EER), qui correspond à la configuration du système où les deux taux sont égaux. Dans un système de classification, plus le taux d'égale erreur est bas, meilleur est le système.

5.3.1.2 Minimum de la fonction de coût

Une alternative à l'EER est le minimum de la fonction de coût, elle permet de pénaliser plus fortement un type d'erreur par rapport à l'autre. On définit la fonction de coût C_{det} de la façon suivante :

$$C_{det}(P_{miss}, P_{FA}) = C_{miss}P_{miss}P_{FA} + C_{FA}P_{FA}(1 - P_{tar}) \tag{5.3}$$

P_{miss} et P_{FA} sont respectivement les taux de faux rejet et de fausse acceptation. On définit P_{tar} comme la probabilité a priori que deux *i-vectors* représentent un même locuteur, dans notre cas, positionnée à 1%. Les paramètres C_{miss} et C_{FA} représentent les coûts respectifs des deux types d'erreur. Dans notre problème ils sont positionnés à 1.

Le minimum de la fonction de coût correspond au minimum de la fonction C_{det}, obtenu en faisant varier le seuil de décision d'acceptation/rejet du système. C'est une mesure de qualité : plus la valeur du minDCF est basse, meilleur est le système.

5.3.2 Implémentation

Le réseau de neurones consiste en une couche *feed-forward* complètement connectée de dimension 200 (identique à la dimension des *i-vectors*), suivie d'une couche d'activation *tanh*. L'apprentissage se fait par descente de gradient avec l'algorithme Adadelta [Zeiler, 2012] et le réseau est implémenté avec Keras/Theano [Chollet, 2017]. Dans les prochains paragraphes, nous allons étudier quelques aspects clés de la configuration du réseau : le choix de la marge, le nombre de plus proches voisins et la représentativité des classes. Dans toutes les expériences, nous décidons d'utiliser la méthode *soft-selection* pour le choix des triplets, car nos expériences préliminaires n'ont pas montré de différences significatives avec la méthode *hard-selection*.

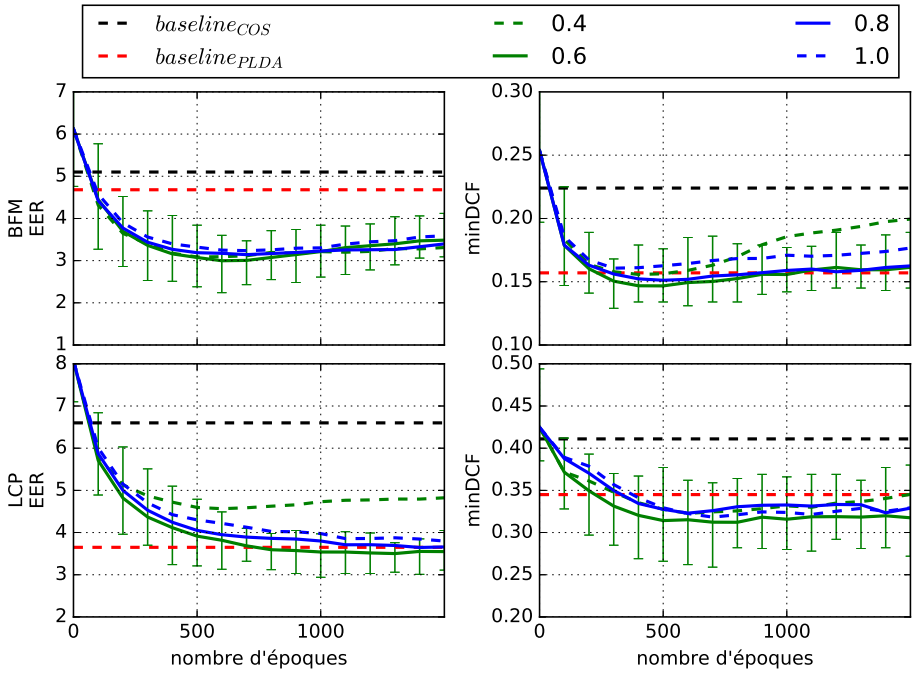

FIGURE 5.2 – Taux d'égale erreur (EER) et minDCF moyens sur les deux collections cibles, en fonction du nombre d'époques, en utilisant différentes marges pour l'apprentissage. Chaque expérience est répétée 20 fois.

5.3.3 Choix de la marge

La marge constitue une notion de distance, ainsi, nous définissons la relation entre distance d et similarité cosine s de la façon suivante : $d = 1 - s$. La figure 5.2 présente les résultats de la tâche de reconnaissance du locuteur sur les *i-vectors* de référence des deux collections cibles, en fonction du nombre d'époques. Comme l'initialisation du réseau se fait aléatoirement, l'influence des paramètres est étudiée pour une initialisation donnée. Différentes valeurs de marge sont comparées, de 0.4 à 1.0 avec un pas de 0.1, à partir de 20 initialisations différentes. A chaque époque, un triplet par classe-locuteur est présenté, le nombre de plus proches voisins étant

fixé à 100. Pour une meilleure lisibilité, seule la moitié des courbes est présentée (un pas de 0.2).

Les résultats montrent que l'approche proposée fait mieux que l'approche cosine seule pour toutes les marges testées et pour les deux métriques. La marge donnant les meilleures performances est de 0.6 et est présentée avec l'intervalle $[min, max]$ de performances sur les 20 expériences réalisées. Si on compare avec l'approche PLDA, on constate que la méthode proposée, pour une marge de 0.6, donne de meilleures résultats en moyenne pour les deux métriques. Cependant, elle reste proche quant au taux d'égale erreur pour LCP et au minDCF pour BFM.

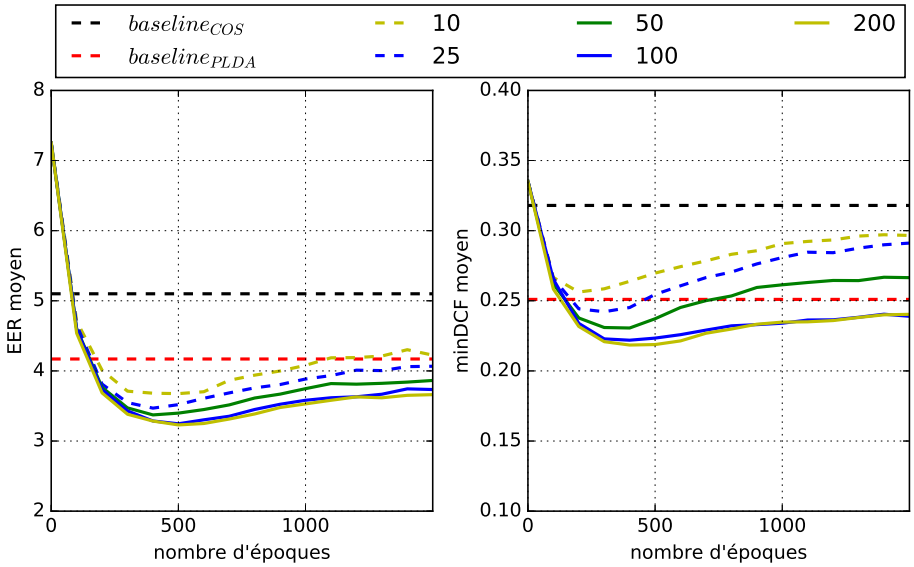

FIGURE 5.3 – Taux d'égale erreur (EER) et minDCF moyens sur les deux collections cibles, en fonction du nombre d'époques, avec différents k-PP pour la sélection des *négatifs*. Chaque expérience est répétée 20 fois.

5.3.4 Nombre de plus proches voisins

En pratique, l'utilisation des plus proches voisins pour la sélection des *negatives* est une façon d'accélérer l'apprentissage. En effet, la majorité des *negatives* qui respectent la contrainte de marge est en général dans un voisinage restreint de l'*anchor*, et il ne sert à rien d'aller tester des *negatives* situés trop loin. On choisit donc de tester seulement parmi les plus proches voisins. En pratique, c'est le fait qu'on ne recalcule pas les plus proches voisins à chaque époque qui fait gagner du temps de calcul. On considère que sur plusieurs époques successives, les plus proches voisins ne changent pas beaucoup.

Dans la figure 5.3, on explore le choix du nombre de plus proches voisins et leur influence sur l'EER et le minDCF. Les k-PPs sont mis à jour toutes les 50 époques. Le protocole d'évaluation de la section précédente est réutilisé : évaluation contrastive des paramètres à partir d'un même réseau initial, répétée pour 20 initialisations

différentes et calcul des performances moyennes sur les 20 expériences.

Les résultats montrent que plus le nombre de proches voisins est élevé, plus les performances sont bonnes, mais au-delà de 100, la progression est très faible. Nous supposons donc que la majorité des bons candidats *negatives* est localisée parmi les 100 plus proches voisins des *anchors*. Cette configuration de 100 plus proches voisins sera conservée pour les expériences suivantes, c'est un bon équilibre entre performances et temps de calcul.

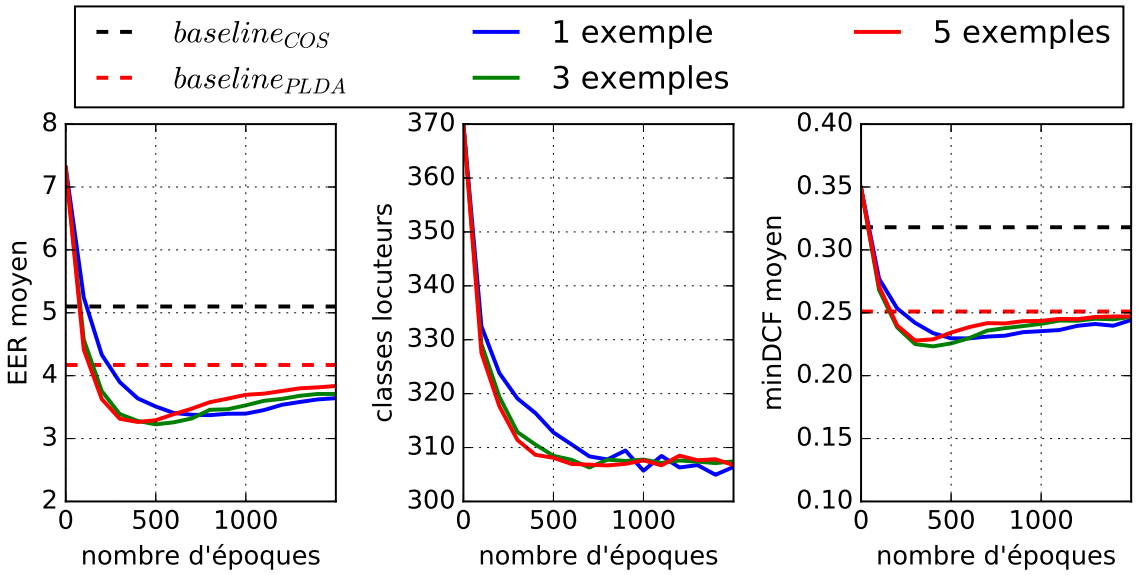

FIGURE 5.4 – Influence du nombre d'exemples fournis par locuteur à l'apprentissage, sur le taux d'égale erreur et le minDCF moyen des deux collections cibles, et sur le nombre de locuteurs contribuant à la *loss*, en fonction du nombre d'époques.

5.3.5 Représentativité des classes

Dans la littérature [Bredin, 2016; Schroff et al., 2015], les réseaux de neurones utilisant la *triplet loss* sont entraînés sur des données proches du signal brut (images ou coefficients cepstraux pour l'audio), en présentant 40 triplets par classes à chaque époque. Or nos données d'apprentissage sont constituées de *i-vectors*, et le nombre d'exemples par classe est très limité (le corpus d'apprentissage contient entre 3 et 59 *i-vectors* par classe, la moitié des classes ne contenant que 5 *i-vectors* ou moins). Puisque nous ne pouvons pas apporter beaucoup de diversité pour les paires de *i-vectors* (3 exemples dans une classe signifie seulement 6 paires *anchor-positive* possibles), on explore l'influence du nombre de triplets par classe présentés à chaque époque. Si on fournit trop de triplets par classe, l'apprentissage pourrait être biaisé par le fait qu'on présente toujours les mêmes paires *anchor-positive* pour certaines classes, même si la diversité est assurée par la présence des *negatives*. Par ailleurs, on souhaite quand même fournir plusieurs triplets par classe, afin d'apprendre au réseau à compenser la variabilité intra-locuteur, et pour compenser un effet observé

lors de nos premières expériences : au fil des époques, certaines classes ne disposent plus de *negatives* présents dans la marge, ce qui fait qu'elles arrêtent de contribuer à l'apprentissage. Les résultats sont présentés dans la figure 5.4, où on compare les performances de réseaux auxquels on présente 1, 3 ou 5 triplets par classe, chaque expérience étant répétée 20 fois. On note aussi le nombre de classes contribuant à la *loss* au fil des époques.

Les résultats montrent qu'utiliser plus d'un triplet par classe-locuteur fonctionne mieux, ce qui nous laisse donc penser que le réseau apprend à compenser la variabilité intra-locuteur/inter-document. Pour les trois configurations testées, on observe également une diminution d'environ 17% du nombre de classes-locuteurs contribuant à l'apprentissage. Le choix de 3 triplets par classe donne les meilleurs EER et minDCF, nous choisissons donc cette configuration pour les expériences suivantes.

5.4 Evaluation des Performances de SRL

La configuration du réseau de neurones étant fixée (marge de 0.6, 100 plus proches voisins, 3 exemples par classe), nous étudions maintenant les performances sur la tâche de SRL, en utilisant le DER intra- et inter-document comme métrique d'évaluation. Nous arrêtons d'utiliser les *i-vectors* de référence et utilisons ceux extraits à partir de la segmentation en locuteur. La table 5.1 rappelle les performances des méthodes contrastives, avec un regroupement hiérarchique complet (HAC) ou en composantes connexes (CC). Les seuils de regroupement sont communs aux deux collections cibles.

scoring	clust.	λ_I	λ_X	LCP DER		BFM DER	
				intra-.	inter-.	intra-.	inter-.
cosine	HAC	-10	0	8.5	19.5	13.6	23.8
	CC	-10	-30	8.5	22.6	12.4	19.3
PLDA	HAC	10	10	10.0	19.1	10.6	15.7
	CC	-10	-20	8.7	21.2	9.9	13.6
TR_{avg}	CC	-10	-30	8.0	16.6	9.8	13.3
TR_{best}				**7.9**	**16.1**	**9.6**	**13.1**

TABLE 5.1: Performances *baseline* des systèmes contrastifs et proposés, pour les regroupements HAC et CC, exprimées en DER intra- et inter-document.

La figure 5.5 montre les DER moyens sur les deux collections, le regroupement étant effectué sur la base de scores TR. Les moyennes sont calculées à partir des résultats obtenus pour 20 réseaux de neurones différents. Les intervalles (min, max) sont aussi présentés toutes les 100 époques. Les résultats montrent que pour les deux collections, après 1300 époques, la méthode proposée avec un regroupement CC (courbe bleue pleine) bat la PLDA avec un regroupement HAC (courbe rouge en

pointillés), en moyenne. En effet, les DER inter-document moyens à 1300 époques, reportés dans la table 5.1 (ligne TR_{avg}) sont de 16.6% (respectivement 13.3%) pour LCP (resp. BFM), tandis que la PLDA avec un regroupement HAC donnait un DER de 19.1% (resp. 15.7%). On note même dans la table 5.1 qu'en choisissant le meilleur réseau parmi les 20 testés (ligne TR_{best}), le DER atteint 16.1% (resp. 13.1%).

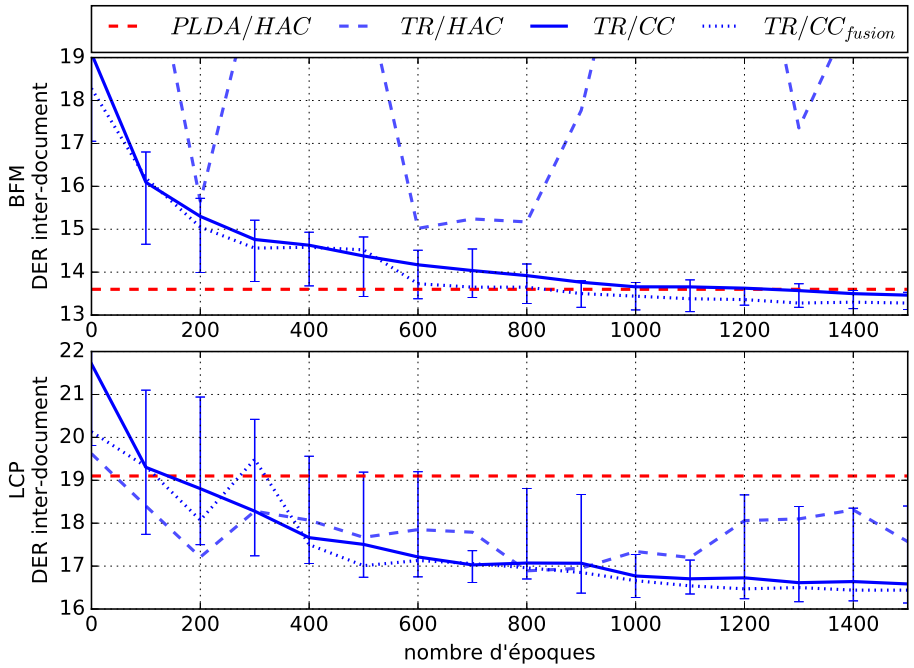

FIGURE 5.5 – Evolution du DER inter-document moyen sur les deux collections cibles, avec la similarité TR et un regroupement CC ou HAC, en fonction du nombre d'époques. La configuration du réseau de neurones est avec une marge de 0.6, la stratégie *soft selection* et 3 triplets par classe à chaque époque. La configuration du regroupement CC est ($\lambda_I = -10, \lambda_X = -30$).

Dans le chapitre précédent, nous avons également travaillé avec le regroupement HAC, mais nos résultats préliminaires avec les scores TR donnaient de mauvais résultats (variations importantes du DER entre différentes époques), sans qu'on puisse y trouver une explication. Comme présenté sur le graphe supérieur de la figure 5.5, pour BFM, le HAC moyen (courbe bleue en pointillés longs) donne des variations assez fortes de DER inter-document, en fonction des époques. Cependant, nous avons remarqué que l'utilisation du regroupement CC était plus adéquate, avec une réduction régulière du DER à travers les époques. En effet, le réseau fait en sorte de séparer les classes par une marge donnée, tandis que le regroupement CC consiste à décider que deux éléments appartiennent à la même classe s'ils sont éloignés d'une distance inférieure à un certain seuil. Par conception, les deux approches se complètent assez bien : si le réseau fonctionne de façon idéale, on ne peut regrouper que des éléments appartenant à la même classe. Ce n'est pas le cas ici, sur la figure, la configuration de regroupement présentée est la même que la *baseline* cosine

simple ($\lambda_I = -10, \lambda_X = -30$), ce qui, en distance cosine, revient à se placer à ($\lambda_I = 0.45, \lambda_X = 0.35$), la marge étant positionnée à 0.6.

Il est également intéressant de noter que les DER inter-document minimaux sont atteints pour plus de 1300 époques, tandis que sur les expériences de reconnaissance du locuteur, les performances avaient tendance à légèrement se dégrader dans la même zone, tout particulièrement pour BFM. Ceci peut s'expliquer par le fait que les *i-vectors* de référence, utilisés pour évaluer la tâche de reconnaissance, sont différents des *i-vectors* utilisés pour évaluer la tâche de SRL. Pour BFM, nous avons aussi remarqué dans la section 5.3.3 que les performances des scores TR étaient proches de la PLDA en minDCF, contrairement à LCP. La conclusion est la même quand on regarde le DER, et pour cause : le minDCF évalue un point de fonctionnement du classifieur où pour un *i-vector* donné, il y a plus d'imposteurs possibles que de candidats de la même classe. C'est ce qui caractérise la tâche de regroupement inter-documents !

Enfin, la figure 5.5 présente également les performances du système qui consiste à exploiter les 20 réseaux pour effectuer le regroupement (courbe bleue en pointillés fins) : on concatène la sortie des 20 réseaux pour en faire un supervecteur, qui est ensuite utilisé pour calculer les similarités cosine. Il est intéressant de noter que les performances d'un tel système sont légèrement meilleures que le système moyen. En pratique, on choisit généralement un seul réseau sur la base de ses bonnes performances sur un corpus de développement, mais il est tout de même utile de savoir qu'en ne faisant aucune sélection et qu'en exploitant tous les réseaux, on obtient des performances dans la moyenne.

5.5 Analyse en locuteurs

A l'aide des méthodes de visualisation proposées au chapitre 3, nous pouvons comparer les performances, par type de locuteur, du système de SRL utilisant la compensation neuronale de variabilité (TR) avec les systèmes *baseline*.

5.5.1 Analyse d'erreur

5.5.1.1 Sur la collection LCP

La figure 5.6 compare les performances nominales par type de locuteur, entre les systèmes HAC/PLDA et CC/TR, pour la collection LCP. Le système HAC/PLDA est, parmi tous les systèmes proposés au chapitre 3, le plus performant sur cette collection, et est battu par le système CC/TR, en moyenne. Le réseau de neurones pour lequel les performances sont présentées est un réseau avec des performances proches de la moyenne : le DER inter-document est à 16.5% (moyenne à 16.6%), tandis que le système de SRL HAC/PLDA affiche un DER de 18.1%.

A la comparaison, le système utilisant la compensation neuronale améliore les taux d'erreur sur tous les types de locuteurs sauf les invités récurrents. Chez les locuteurs ponctuels, le taux d'erreur classe est divisé par deux, passant de 70.5% à 35.8% et de 10.7% à 4.3%, et on note une diminution des locuteurs n'étant attribués à aucune classe-locuteur (affichant un taux d'erreur à 100%). Chez les journalistes récurrents, le taux d'erreur classe diminue légèrement de 10.0% à 9.2%, même si médiane et moyenne sont légèrement plus élevées. Cependant, on constate pour les invités récurrents une légère augmentation du taux d'erreur classe, de 28.0% à 28.6%, la médiane dépassant les 40%.

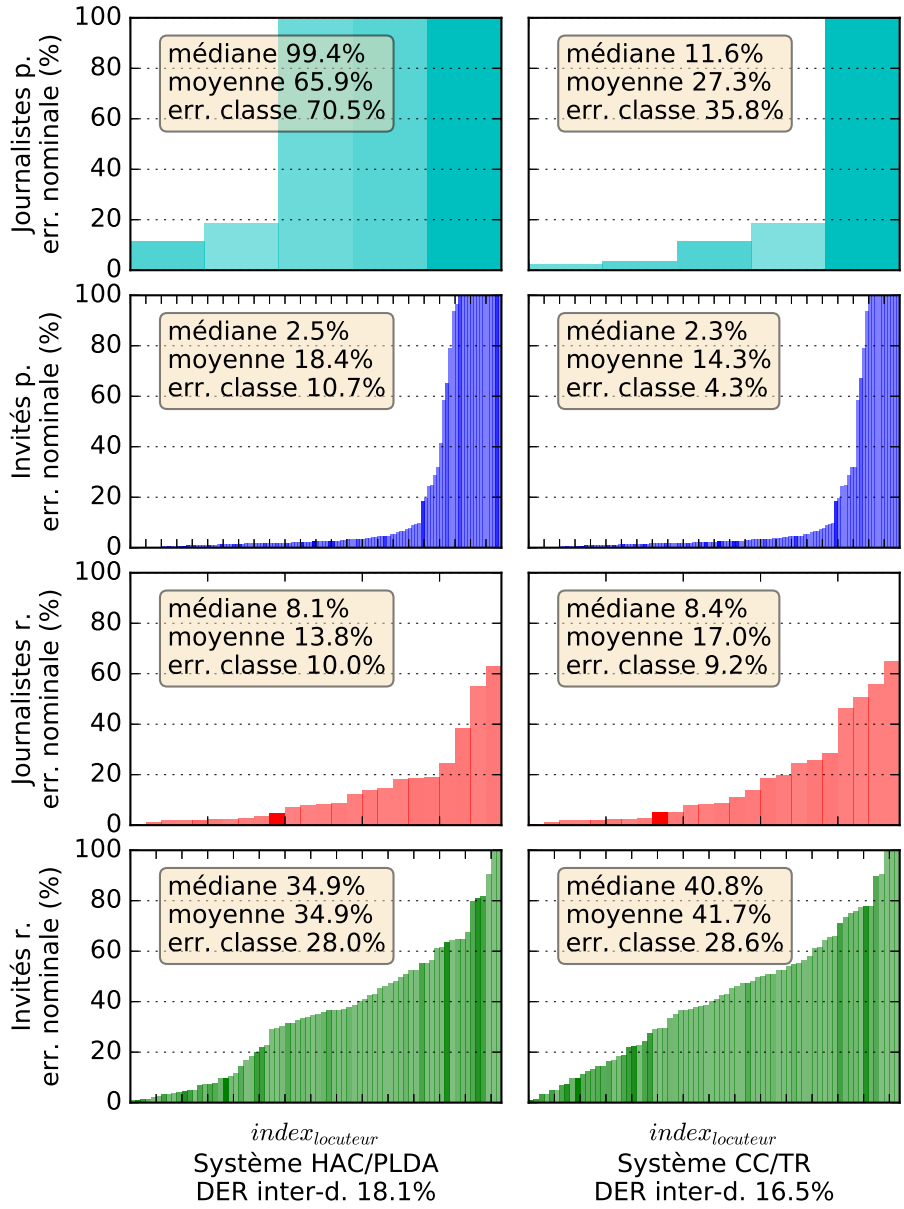

FIGURE 5.6 – Analyse d'erreur comparative entre les systèmes HAC/PLDA ($\lambda_I = -10, \lambda_X = 10$) et CC/TR ($\lambda_I = -10, \lambda_X = -30$), dans la configuration minimisant le DER, pour la collection LCP.

5.5.1.2 Sur la collection BFM

Sur la figure 5.7, on effectue la même analyse comparative que précédemment, entre les systèmes HAC/PLDA et CC/TR, sur la collection BFM. Le système HAC/PLDA donne un DER inter-document de 15.7% contre 13.4% pour le système CC/TR.

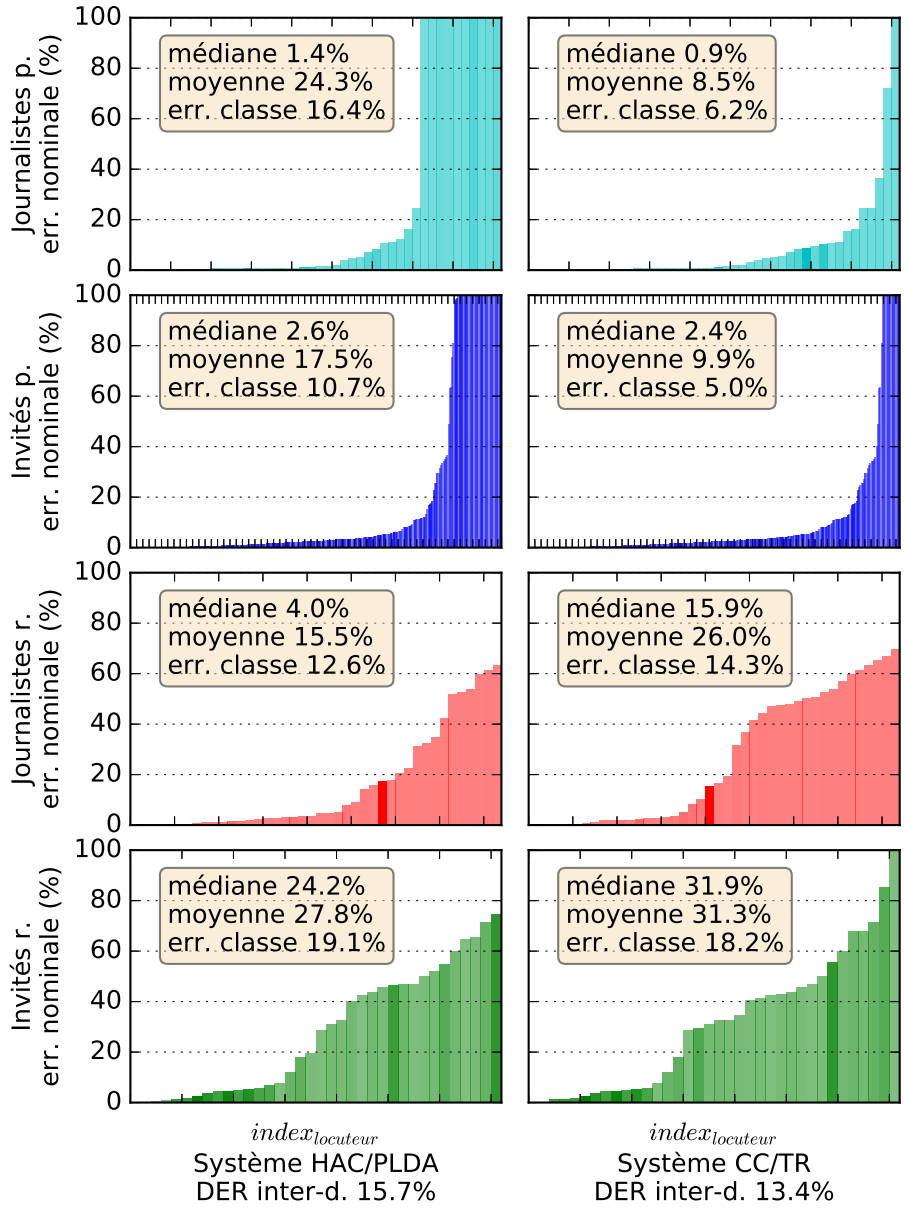

FIGURE 5.7 – Analyse d'erreur comparative entre les systèmes HAC/PLDA ($\lambda_I = 10, \lambda_X = 10$) et CC/TR ($\lambda_I = -10, \lambda_X = -30$), dans la configuration minimisant le DER, pour la collection BFM.

Cette fois, c'est sur les journalistes récurrents que l'approche neuronale fait augmenter le taux d'erreur classe. Il passe de 12.6% à 14.3%, et surtout, la médiane est multipliée par 4, passant de 4.0% à 15.9%. En effet, avec l'approche TR, on note un plus grand nombre de locuteurs pour lesquels le taux d'erreur classe dépasse les 40%.

En revanche, on peut encore constater les bonnes performances du système CC/TR sur les journalistes et invités ponctuels, le taux d'erreur classe étant divisé respectivement par 3 et 2. On compte bien moins de locuteurs n'ayant pas été appariés à une classe-locuteur. Chez les invités récurrents, le taux d'erreur classe diminue également, passant de 19.1% à 18.2%, même si médiane et moyenne augmentent légèrement.

5.5.1.3 Bilan

Sur les deux collections, on constate donc que l'approche CC/TR comparée à l'approche HAC/PLDA améliore les performances sur les locuteurs ponctuels. Concernant les locuteurs récurrents, l'amélioration est plus faible. Le taux d'erreur classe diminue légèrement pour un type de locuteur récurrent (journalistes sur LCP, invités sur BFM), mais on observe de façon générale une augmentation des médiane et moyenne chez les récurrents. Ces résultats sont à mettre en relation avec la philosophie du regroupement en composantes connexes, un regroupement associatif moins robuste que le regroupement HAC.

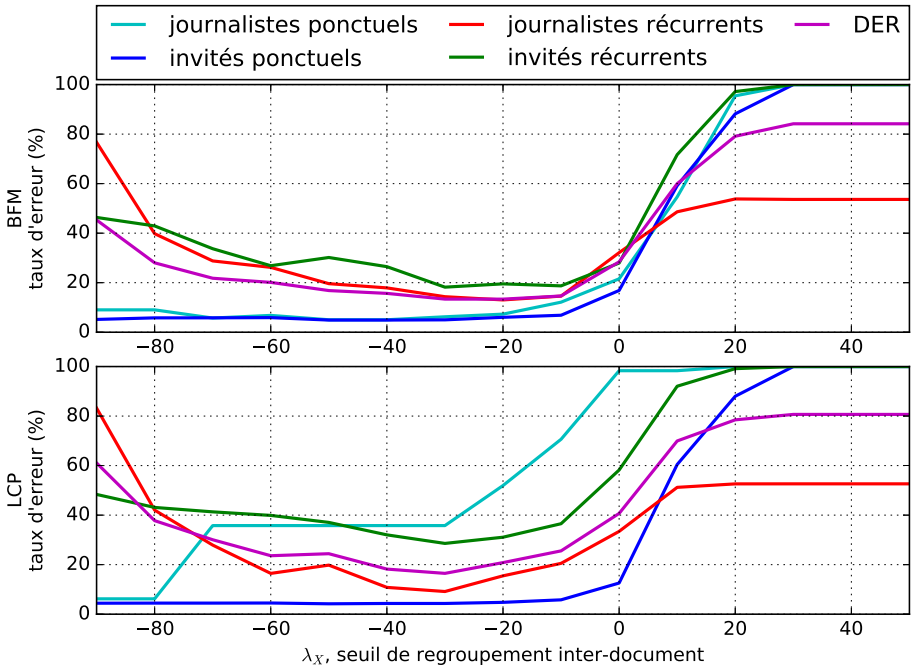

FIGURE 5.8 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement CC λ_X, avec la similarité TR.

5.5.2 Dynamique des taux d'erreur

A la section 4.3.5, nous avons étudié l'évolution des taux d'erreur par type de locuteur en fonction du seuil de regroupement. Pour le système CC/TR, nous utilisons le même type de représentation à la figure 5.8, où nous visualisons l'évolution

des quatre taux d'erreur classe et du DER en fonction du seuil de regroupement inter-document. Sur les deux collections, le DER est minimal pour $\lambda_X = -30$, et on peut noter qu'à ce seuil le taux d'erreur pour chaque classe est très proche de son minimum (hormis pour les journalistes ponctuels de LCP, classe marginale dans la collection). Cet alignement des optima peut expliquer les DER particulièrement bons obtenus avec ce système de SRL (pour rappel, 13.4% pour BFM et 16.5% pour LCP).

5.6 Conclusions sur l'approche neuronale

Dans ce chapitre, nous avons proposé une nouvelle méthode de mesure de similarités entre *i-vectors*. Un réseau de neurones projette les *i-vectors*, de façon non linéaire, dans un espace qui optimise la séparation des classes-locuteurs du point de vue de la similarité cosine. Pour entraîner le réseau, on utilise la *triplet loss*, qui permet de focaliser l'apprentissage sur les exemples les plus difficiles. Les seuils de regroupement optimisant le DER dans le nouvel espace sont les mêmes que dans l'espace de variabilité totale initial, avec la similarité cosine et un regroupement CC. Comme le nombre d'exemples d'apprentissage est relativement faible, seulement 3 triplets par classe-locuteur sont utilisés à chaque époque, et certaines classes arrêtent de contribuer à l'apprentissage après quelques époques.

Les performances de SRL sur les deux collections cibles montrent que l'approche proposée est compétitive avec les méthodes de calcul de similarités à l'état de l'art, telles que la PLDA, que ce soit en DER intra- ou inter-document. La principale différence réside dans le nombre de paramètres du réseau de neurones à optimiser : ils sont bien plus nombreux que ceux de la PLDA et nécessitent donc beaucoup d'expériences d'apprentissage avant d'obtenir une configuration optimale. Cependant, une fois le réseau entraîné, le calcul de similarités est aussi rapide qu'avec la PLDA.

On a également remarqué que le choix du regroupement HAC était incompatible avec l'approche TR, qui doit donc être utilisé avec un regroupement CC pour être optimal. C'est une différence notable avec la PLDA, qui fonctionne mieux avec un regroupement HAC. Or le regroupement HAC est conservatif et donc moins sensible aux erreurs que le regroupement CC, associatif, ce qui constitue un avantage pour l'approche PLDA/HAC comparée à l'approche TR/CC.

Chaque système a donc ses avantages et ses inconvénients. Si la méthode TR/CC donne les meilleurs résultats, on pourrait lui préférer la méthode PLDA/HAC pour le caractère conservatif du regroupement et le faible nombre de paramètres à optimiser pour l'apprentissage. Les résultats obtenus restent à être confirmés sur la tâche de reconnaissance du locuteur, où le nombre de locuteurs disponibles pour l'apprentissage est bien plus élevé (du millier à la centaine de milliers). Cependant, l'approche Triplet Ranking semble prometteuse pour les futurs systèmes de SRL.

Deuxième partie

Adaptation au domaine d'un système de SRL

Chapitre 6

Adaptation au domaine

Résumé

Ce chapitre, qui constitue une des contributions de la thèse [Le Lan et al., 2016c], est consacré à la question de l'adaptation au domaine pour la SRL de collection. Après un état de l'art portant sur l'adaptation au domaine pour la tâche de reconnaissance du locuteur, nous proposons une stratégie d'adaptation pour la SRL de collection. L'idée sous-jacente est d'utiliser les classes-locuteurs produites par une première passe de SRL de collection pour mettre à jour les modèles de calcul de similarités. Ainsi, une nouvelle passe de regroupement inter-document peut être effectuée avec des modèles mis à jour et adaptés à la collection traitée. Le processus, vertueux, peut être répété plusieurs fois pour alternativement améliorer les modèles et la qualité des classes-locuteurs produites. Nous proposons donc une méthode d'adaptation itérative pour chaque type de compensation de variabilité (WCCN, PLDA, TR), en utilisant une pondération des données d'apprentissage initial et de la collection. Les évaluations montrent que le processus d'adaptation itérative permet de réduire le DER pour la majorité des configurations testées, et fonctionne particulièrement bien avec le regroupement HAC. De plus, il est robuste au choix du seuil de regroupement inter-document. L'analyse en locuteurs met en évidence le fait que le taux d'erreur nominal de plusieurs locuteurs ponctuels diminue lors de la première itération d'adaptation, tandis que les itérations supplémentaires permettent de réduire l'erreur des locuteurs récurrents.

6.1 Introduction

Dans la première partie, nous avons présenté notre système de SRL de collection à l'état de l'art (au chapitre 4), et avons proposé une nouvelle méthode de compensation neuronale de variabilité intra-locuteur/inter-document (TR) (au chapitre 5), qui s'est révélée plus performante que les approches classiques telles que la PLDA.

Quelle que soit la méthode de compensation de variabilité utilisée, on a pu noter un écart significatif de performances entre DER intra- et DER inter-document, pouvant varier du simple au double. L'adaptation au domaine pourrait être un moyen de réduire cet écart.

Dans les applications classiques de reconnaissance vocale, des modèles acoustiques sont estimés sur des données d'apprentissage, pour traiter des données cibles. Souvent, lorsque les données d'apprentissage et cibles ne sont pas du même domaine, les performances sur les données cibles sont dégradées : les modèles n'ont pas été appris pour ce type de données et fonctionnent mal. C'est le champ d'application de l'adaptation au domaine, qui vise à compenser la différence de conditions acoustiques entre domaine source et domaine cible. En général, un corpus de développement, du domaine cible, sert à adapter les modèles.

Pour les évaluations présentées dans ce manuscrit, nous ne disposons pas d'un tel corpus de développement puisque nous ne possédons que deux collections. Nous proposons donc d'utiliser directement les collections cibles pour l'adaptation. Ce n'est pas une contrainte étant donné la tâche à laquelle on s'intéresse : comme on considère les collections dans leur ensemble pour réaliser la SRL, on peut très bien les utiliser pour réaliser l'adaptation.

Un système de SRL de collection à l'état de l'art repose sur trois modèles nécessitant un apprentissage supervisé, avec des niveaux de supervision (ie. d'annotation) différents selon les modèles :

- Le modèle du monde (GMM/UBM) requiert des segments de parole non bruitée pour être appris.
- La matrice de Variabilité Totale (TV) doit être entraînée sur des segments de parole contenant chacun la voix d'un seul locuteur.
- Les modèles de compensation de variabilité intra-locuteur/inter-document (WCCN, PLDA, TR) sont ceux nécessitant le plus haut niveau d'annotation : pour un grand nombre de locuteurs, on a besoin de plusieurs segments de parole ne contenant que leur voix, dans des environnements acoustiques variés.

Lorsqu'on veut effectuer un traitement de SRL sur une nouvelle collection, il y a trois possibilités :

1. Si la collection cible est acoustiquement et/ou structurellement proche du jeu d'entraînement disponible (par exemple de nouveaux épisodes d'une émission déjà présente dans le corpus d'apprentissage), on peut entraîner le système sur les données d'entraînement, de façon supervisée.
2. Lorsque les différences d'environnement acoustique sont importantes, on peut utiliser les données cibles (sans annotations) pour adapter les paramètres du système source, de manière non supervisée.
3. Enfin, si l'on en a les moyens, on peut annoter manuellement des données

du domaine cible pour apprendre et/ou adapter les paramètres de manière supervisée.

Les travaux de ce chapitre sont dédiés à la deuxième approche. Dans les sections suivantes, nous commencerons par faire un état de l'art de l'adaptation au domaine pour la tâche de vérification du locuteur, comme c'est une problématique qui n'a jamais été abordée pour la tâche de SRL. Nous proposerons ensuite une stratégie d'adaptation itérative pour la SRL de collection, puis évaluerons la stratégie proposée sur les différents systèmes de SRL à l'état de l'art, à travers la mesure du DER et l'analyse en locuteur des résultats.

6.2 Etat de l'art en reconnaissance du locuteur

L'apprentissage d'un système de SRL à l'état de l'art requiert plusieurs dizaines d'heures de parole annotées pour estimer la variabilité intra- et inter-locuteur. Lorsque qu'on ne dispose pas de suffisamment de données issues du domaine cible, la solution consiste à adapter des modèles appris sur des données du domaine source. Pour la tâche de vérification du locuteur, dans [Shum et al., 2014b], il a été montré que le composant le plus important pour l'adaptation au domaine d'un système *i-vector/PLDA* est la PLDA. Par conséquent, nous allons principalement nous intéresser aux méthodes d'adaptation au domaine agissant sur la covariance intra-classe [Glembek et al., 2014] ou la PLDA [Garcia-Romero and McCree, 2014; Shum et al., 2014b], en mentionnant que d'autres approches existent, se concentrant sur l'adaptation de l'espace de variabilité totale [Aronowitz, 2014; Chen et al., 2015; Kanagasundaram et al., 2015].

6.2.1 Adaptation de la covariance intra-classe

Dans [Glembek et al., 2014], les auteurs proposent une méthode d'adaptation de la covariance intra-locuteur. Le système de comparaison de locuteurs utilise le paradigme *i-vector*, combiné à une analyse discriminante linéaire (LDA) pour réduire le nombre de dimensions et à la PLDA pour le calcul de scores. Comme la LDA repose sur le calcul des covariances intra- et inter-classe, la proposition consiste à adapter la variabilité intra-locuteur en ajoutant la variabilité inter-domaine.

$$\boldsymbol{W_{new}} = \boldsymbol{W} + \alpha \boldsymbol{W_{BD}} \tag{6.1}$$

Le facteur α permet d'exagérer la variabilité inter-domaine, qui ne dépend pas du locuteur, ce qui a pour effet de mieux la compenser lors de la LDA.

6.2.2 Adaptation de la PLDA

Deux méthodes pour l'adaptation de la PLDA sont présentées. Le choix de la méthode dépend de la quantité de données cibles disponibles pour réaliser l'adaptation.

6.2.2.1 Vraisemblance pondérée

L'idée principale de l'adaptation au domaine par vraisemblance pondérée (Weighted Likelihood Domain Adaptation [Garcia-Romero and McCree, 2014]) est de décomposer l'expression du maximum de vraisemblance pour l'apprentissage de la PLDA en la pondération de deux termes relatifs à chaque domaine. Cette méthode a l'avantage de fonctionner avec une faible quantité de données cibles. On note $(\boldsymbol{\phi_{ij}})$ l'ensemble des n_i *i-vectors* du locuteur i. La distribution conjointe des *i-vectors* est donc :

$$p((\boldsymbol{\phi_{ij}})|\boldsymbol{\Gamma}, \boldsymbol{\Lambda}) = \mathcal{N}((\boldsymbol{\phi_{ij}}); 0, \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T + \tilde{\boldsymbol{\Lambda}}) \tag{6.2}$$

L'idée de l'adaptation est de décomposer l'expression maximum de vraisemblance pour introduire un coefficient de pondération α :

$$L(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \alpha L_{in}(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) + (1-\alpha)L_{out}(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) \tag{6.3}$$

Où

$$L_k(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \frac{1}{N_k}\sum_{s=1}^{S_k}\sum_{j=1}^{n_{ik}} log(p((\boldsymbol{\phi_{ij}})|\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda})) \tag{6.4}$$

N_k est le nombre d'*i-vectors* du domaine k and S_k est le nombre de locuteurs. L'avantage de la méthode est de pouvoir choisir le coefficient de pondération, qui quantifie l'influence des données cibles par rapport aux données sources lors de l'apprentissage. L'estimation des paramètres de la PLDA adaptée est similaire à la méthode classique, à ceci près que l'apprentissage se fait sur deux corpus en simultané, à l'aide du paramètre de pondération. A l'étape Espérance, on calcule la moyenne *a posteriori* $E[\boldsymbol{h_{ik}}]$ et la corrélation $E[\boldsymbol{h_{ik}}\boldsymbol{h_{ik}}^T]$ des variables locuteur cachées pour chaque corpus.

A l'étape Maximisation, les paramètres sont mis à jour selon :

$$\boldsymbol{\Phi_{new}} = \left(\sum_{k=cible}^{source} \dot{\alpha_k}\sum_{i=1}^{S_k}\sum_{j=1}^{n_{ik}} \boldsymbol{\phi_{ijk}}E[\boldsymbol{h_{ik}}]^T\right)\left(\sum_{k=cible}^{source} \dot{\alpha_k}\sum_{i=1}^{S_k} n_{ik}E[\boldsymbol{h_{ik}}\boldsymbol{h_{ik}}^T]\right)^{-1} \tag{6.5}$$

$$\boldsymbol{\Lambda_{new}} = \sum_{k=cible}^{source} \dot{\alpha_k} \sum_{i=1}^{S_k} \sum_{j=1}^{n_{ik}} \left[\boldsymbol{\phi_{ijk}} \boldsymbol{\phi_{ijk}^T} - \boldsymbol{\Phi_{new}} E[\boldsymbol{h_i k}] \boldsymbol{\phi_{ijk}^T} \right] \tag{6.6}$$

avec $\dot{\alpha}_{cible} = \alpha$ et $\dot{\alpha}_{source} = 1 - \alpha$.

Dans la littérature [Garcia-Romero and McCree, 2014], les résultats montrent que la PLDA adaptée est plus efficace que la PLDA source, et que le taux d'égale erreur diminue à mesure que le nombre de locuteurs du domaine cible augmente.

6.2.2.2 Interpolation *a posteriori*

Quand le corpus cible contient suffisamment de données (nombre de sessions supérieur à la dimension des *i-vectors*), une approximation de la méthode précédente consiste à entraîner séparément les paramètres de la PLDA source et de la PLDA cible, avec l'algorithme EM, puis d'interpoler les matrices PLDA source et cible. L'avantage de cette méthode est que l'on n'a pas besoin de conserver les *i-vectors* du domaine source, seulement les matrices PLDA. La mise à jour des matrices se fait de la manière suivante :

$$\boldsymbol{\Phi \Phi^T_{final}} = \alpha_1 \boldsymbol{\Phi \Phi^T_{in}} + (1 - \alpha_1) \boldsymbol{\Phi \Phi^T_{out}} \tag{6.7}$$

$$\boldsymbol{\Lambda_{final}} = \alpha_2 \boldsymbol{\Lambda_{in}} + (1 - \alpha_2) \boldsymbol{\Lambda_{out}} \tag{6.8}$$

On peut déterminer α_1 et α_2 par recherche exhaustive, les deux paramètres pouvant ne pas être positionnés à la même valeur. Dans [Garcia-Romero and McCree, 2014], les résultats montrent que l'interpolation apporte un gain par rapport à l'utilisation de la PLDA source seule, et que la méthode donne des performances comparables à l'approche par vraisemblance pondérée.

6.2.3 Adaptation non supervisée

Dans certains cas, les *i-vectors* d'adaptation ne sont pas annotés en locuteurs [Khoury et al., 2014; Shum et al., 2014b; Villalba and Lleida, 2014], l'adaptation de la PLDA doit se faire de façon non supervisée, en utilisant une méthode de regroupement pour les étiqueter automatiquement. Par exemple, dans [Shum et al., 2014b], les paramètres de la PLDA source sont utilisés pour calculer des similarités entre les *i-vectors* cibles. La matrice des similarités permet ensuite de regrouper ces *i-vectors* en différentes classes-locuteurs, qui peuvent alors servir à estimer une PLDA adaptée par interpolation avec la PLDA source. Les résultats montrent que l'interpolation est la plus efficace lorsque le nombre de locuteurs est bas, car si celui-ci est suffisamment élevé, il est préférable de n'apprendre que la PLDA cible seule.

Dans [Khoury et al., 2014], l'approche est différente, elle consiste à progressivement regrouper les *i-vectors*, de façon hiérarchique, en mettant à jour le modèle de calcul de similarités au fil des regroupements. Une première étape consiste à regrouper faiblement les vecteurs en classes de petite taille, à l'aide de la similarité cosine. Les classes sont considérées pures. Ensuite, un modèle PLDA est appris sur ces petites classes, permettant de mettre à jour les similarités entre celles-ci et de décider le regroupement suivant à effectuer. Un nouveau modèle PLDA est estimé de nouveau, de façon périodique, après un nombre de regroupements donné. Cette approche utilise uniquement les données cibles, et nécessite donc d'en avoir suffisamment à disposition pour estimer les modèles.

Les auteurs de [Villalba and Lleida, 2014], quant à eux, ont proposé d'utiliser une méthode de Bayes variationnel pour adapter la PLDA. Le principe, itératif, repose sur l'utilisation d'une variable latente pour caractériser les données d'adaptation (des *i-vectors*). En l'occurence, la variable latente représente le locuteur correspondant au *i-vector*. L'approche nécessite de spécifier (ou estimer) le nombre de locuteurs présents dans les données d'adaptation.

6.2.4 Quid de la SRL ?

Jusque récemment, l'adaptation au domaine était principalement étudiée pour la tâche de vérification de locuteurs, pour laquelle les enregistrements audios (téléphoniques, la plupart du temps) ne contiennent en général que la voix d'un locuteur, et où les conditions de variabilité de domaine sont clairement identifiées (prise de son téléphonique vs. microphonique). Les méthodes d'adaptation existantes visent à estimer une meilleure représentation de la variabilité d'une collection, qu'il s'agisse de la variabilité totale ou de la variabilité intra-locuteur. En général, l'adaptation se fait à partir d'un modèle ou de données sources, en utilisant les données de développement (du domaine cible). La Segmentation et le Regroupement en Locuteurs est une tâche plus difficile, où les enregistrements doivent d'abord être segmentés en locuteurs avant de regrouper les segments par locuteur.

6.3 Stratégie d'adaptation proposée pour la SRL

Le but de notre travail est d'étudier comment la connaissance imparfaite de nos collections cibles peut servir à améliorer un système de SRL appris sur des données source, en utilisant des méthodes d'adaptation au domaine. Comme nous l'avons vu dans la section 6.2, la modélisation de la variabilité intra- et inter-locuteur est une étape clé dans le processus d'adaptation. Les collections cibles sont des émissions télévisées dont la taille augmente à mesure que de nouveaux épisodes sont diffusés. Certains locuteurs apparaissent dans différents épisodes : nous proposons de les

utiliser pour adapter le système de SRL de façon non supervisée. Puisque nous souhaitons pouvoir réaliser l'adaptation sur n'importe quel type de collection, la méthode doit pouvoir passer à l'échelle.

La méthode proposée, illustrée par la figure 6.1, consiste en l'adaptation de notre système de SRL à l'état de l'art, appris sur des données sources, avec les données de la collection traitée (collection cible). Sur la figure, la stratégie d'apprentissage supervisé (*baseline*), décrite à la section 4.1, est représentée avec les flèches bleues pleines, tandis que la stratégie d'adaptation non supervisée, décrite dans la section qui suit, est représentée avec les flèches oranges en pointillés.

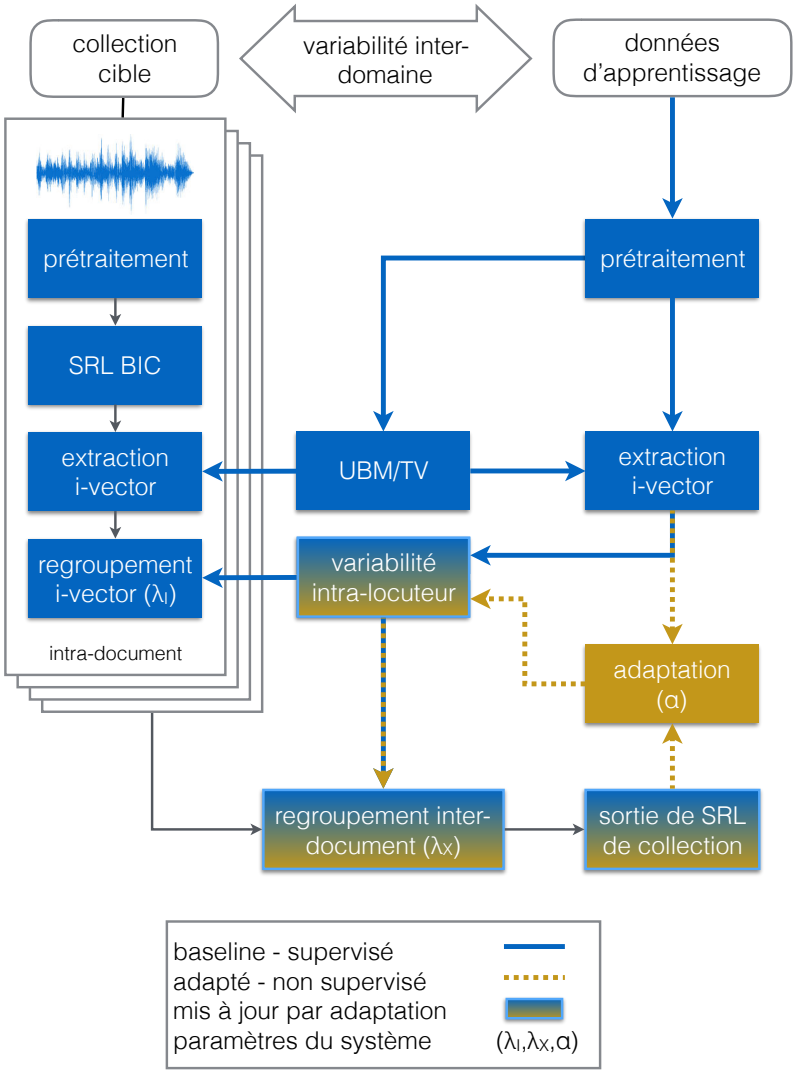

FIGURE 6.1 – Vue d'ensemble du système de SRL *baseline* (lignes bleues pleines) et adapté (lignes bleues en pointillés).

6.3.1 Adaptation non supervisée

La stratégie d'adaptation proposée est itérative : elle alterne adaptation des modèles et regroupement inter-document. En effet, certaines classes-locuteurs géné-

rées par le regroupement inter-document contiennent des segments de parole issus de différents documents, et peuvent contribuer à l'estimation de la variabilité intra-locuteur par une méthode d'adaptation au domaine. Si les modèles de variabilité sont mis à jour, les similarités inter-document peuvent être re-calculées de façon plus précise et devraient améliorer la qualité du regroupement inter-document. L'approche est donc itérative car la mise à jour des classes-locuteurs (plus précises car issues de modèles adaptés à la collection) permet une nouvelle adaptation des modèles, et ainsi de suite.

Même si le système de SRL *baseline* commet des erreurs, on suppose que les classes-locuteurs produites de façon automatique contiennent suffisamment d'information pour affiner l'estimation de la variabilité intra-locuteur/inter-document. L'aspect itératif du processus devrait progressivement améliorer la précision du système de SRL. Les méthodes d'adaptation utilisées pour les calculs de similarité cosine/WCCN, PLDA et cosine/TR sont détaillés dans les trois sections suivantes.

La méthode pourrait très bien servir à mettre à jour les regroupements intra-document, mais, pour des raisons applicatives, nous avons décidé de nous concentrer sur la mise à jour des regroupements inter-document. Une fois le regroupement intra-document terminé, la représentation d'un document consiste seulement en quelques *i-vectors* représentant les différents classes-locuteurs détectées (un seul vecteur par classe), et le nombre de locuteurs présents dans un document ne peut dépasser quelques dizaines. Dès lors, le regroupement inter-document est applicativement facile à mettre à jour dans la mesure où il suffit de recalculer des distances entre *i-vectors* et rejouer le regroupement inter-document. Si ces deux opérations peuvent avoir un coût quadratique en nombre de *i-vectors* à regrouper, nous travaillons sur des collections suffisamment petites pour que ce ne soit pas un problème.

Les paramètres du système de SRL avec adaptation sont donc λ_I, le seuil de regroupement *i-vector* intra-document, λ_X, le seuil de regroupement *i-vector* inter-document, et α, le paramètre d'adaptation, qui pondère l'influence des données source et cible. En pratique, on effectue d'abord une première passe de regroupement *baseline* qui dépend uniquement de (λ_I, λ_X), puis les itérations d'adaptation, qui dépendent uniquement de (λ_X, α). Il n'y a donc pas vraiment d'influence directe de λ_I sur le processus d'adaptation. Remarquons seulement que les regroupements intra-document n'étant pas interdits lors du regroupement inter-document, il est possible que leur nombre varie selon l'itération d'adaptation considérée.

6.3.1.1 Adaptation de la covariance intra-classe

Pour l'adaptation de la WCCN, nous proposons de calculer une nouvelle matrice $\boldsymbol{W}_{\boldsymbol{adaptée}}$, comme l'interpolation entre la matrice WCCN source et la matrice WCCN cible, calculée à partir des classes-locuteurs produites par le système de SRL.

$$\boldsymbol{W_{adaptée}} = \alpha \boldsymbol{W_{cible}} + (1 - \alpha) \boldsymbol{W_{source}} \tag{6.9}$$

Lors d'une nouvelle itération, on adapte toujours la matrice source avec la matrice cible calculée sur les dernières classes-locuteurs. Notons qu'avec cette approche, si un même locuteur se trouve dans le corpus d'apprentissage et dans la collection cible, on ne peut pas utiliser l'information de sa variabilité inter-domaine.

6.3.1.2 Adaptation de la PLDA

Concernant la PLDA, pour des raisons de passage à l'échelle, il n'est pas possible d'utiliser l'interpolation *a posteriori*, car estimer une PLDA cible à partir des classes-locuteurs seules est difficilement envisageable si le nombre de classes est trop faible. En théorie, on pourrait réduire la dimension de la PLDA, mais cela impliquerait de devoir calibrer la dimension selon la taille du problème, et donc d'avoir à calibrer un seuil de regroupement par valeur de dimension. Nous proposons donc d'utiliser la méthode d'adaptation par vraisemblance pondérée, présentée dans la section 6.2.2.1.

6.3.1.3 Adaptation du réseau TR

Concernant l'adaptation du réseau de neurones pour le calcul de scores TR, nous proposons de poursuivre l'apprentissage du réseau de compensation *baseline* avec les classes-locuteurs générées par la SRL *baseline*. De la même façon que pour la WCCN et la PLDA, nous introduisons un paramètre d'adaptation α qui permet de pondérer l'influence des classes sources et cibles dans le calcul de la *loss*.

$$\mathcal{L}(\mathcal{T}) = \alpha \sum_{i}^{N_{cible}} max(0, \Delta_i + \beta) + (1 - \alpha) \sum_{i}^{N_{source}} max(0, \Delta_i + \beta) \tag{6.10}$$

Des paramètres supplémentaires sont à étudier pour réaliser l'adaptation : le nombre d'époques à réaliser et le choix de l'époque initiale. Vaut-il mieux adapter un réseau pré-entraîné ou entraîner un nouveau réseau de zéro ? Nous faisons le choix d'adapter en réalisant 250 époques d'adaptation d'un réseau *baseline* pré-entraîné, figé à 1500 époques. Les raisons de ce choix sont détaillées dans l'annexe C.

6.4 Expériences

6.4.1 Système *baseline*

Le rappel des résultats du système de SRL *baseline* est présenté dans la table 6.1. Plusieurs configurations sont comparées, avec différentes méthodes de calcul de similarités (cosine avec WCCN, PLDA et TR) et de regroupement (HAC ou CC).

Le système *baseline* est intégralement appris sur les données sources. La dimension de l'UBM est de 256, celle de la matrice TV est de 200 et celle de la PLDA est de 100 : ce sont les dimensions retenues au chapitre 3. Pour chaque méthode de calcul de similarités, les DER intra- et inter-document sont affichés pour différentes configurations de regroupement (λ_I, λ_X) : une configuration optimale dédiée à chaque corpus et une optimale commune aux deux collections. Ces résultats sont issus de ceux présentés aux sections 4.2.2.2 et 5.4.

Quand on observe le DER inter-document dans la table 6.1, on voit que les taux d'erreur varient de 27.5% à 16.5% pour LCP et de 24.2% à 13.4% pour BFM. Le meilleur système utilise le réseau TR pour le calcul des scores. Notons qu'il n'existe pas de configuration *dédiée* à LCP ou BFM avec l'approche TR/CC, car la configuration *commune* optimise conjointement les performances sur les deux collections.

similarité	WCCN				PLDA				TR	
regroupement	CC		HAC		CC		HAC		CC	
collection	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM
$baseline_{dediée_{LCP}}$	22.6	15.8	19.7	24.2	20.1	15.4	18.1	21.5	-	-
$baseline_{dediée_{BFM}}$	27.5	15.0	23.0	17.6	24.9	**13.4**	19.1	15.7	-	-
$baseline_{commune}$	22.9	15.4	20.6	19.0	21.2	13.6	19.1	15.7	**16.5**	**13.4**

TABLE 6.1: Récapitulatif des DER inter-document *baseline* sur les collections cibles complètes, avec les différentes mesures de similarités et regroupements.

6.4.2 Adaptation *oracle*

Avant de tester la stratégie d'adaptation proposée sur les collections cibles et afin de quantifier les gains possibles, nous décidons de réaliser une expérience d'adaptation avec les *i-vectors oracle*, c'est-à-dire les *i-vectors* extraits à partir de la segmentation de référence des collections cibles, pour les trois méthodes de calcul de similarité proposées.

Les résultats sont présentés dans les figures 6.2 pour le regroupement HAC et 6.3 pour le regroupement CC. Pour les trois types de score, les modèles *baseline*, appris sur les données source, sont adaptés avec les données cibles *oracle*, avec un coefficient d'adaptation α variant de 0.1 à 1. Ces modèles adaptés sont utilisés pour le regroupement inter-document des *i-vectors* issus du regroupement intra-document avec les modèles *source*. L'expérience $\alpha = 1$ n'est pas présentée pour la PLDA et la WCCN, car les données cibles seules ne permettent pas d'adapter. Le cas $\alpha = 0$ n'est pas affiché, puisqu'il correspond à l'expérience *baseline*. Enfin, pour l'adaptation du réseau TR, nous n'avons pas étudié le cas du regroupement HAC, qui ne donnait pas de résultats satisfaisants à l'expérience *baseline*. Seul le DER inter-document est présenté.

Avec le regroupement HAC, on observe sur la figure 6.2 des gains de 10% à 40% en relatif, selon les cas. Avec la similarité cosine/WCCN, les meilleurs gains possibles sont obtenus pour des valeurs de α élevées (pour BFM, gain de 17.6% à

13.4% avec $\alpha = 0.8$, pour LCP, de 19.7% à 16.1% avec $\alpha = 0.9$). Concernant la PLDA, la plage d'optimalité est moins évidente, avec un gain pour BFM de 15.7% à 12.3% avec $\alpha = 0.3$ et pour LCP de 18.1% à 11.0% avec $\alpha = 0.7$, dans le meilleur des cas. Quelle que soit la configuration, l'adaptation permet de diminuer le DER, et l'adaptation de PLDA semble être plus efficace que la WCCN.

Dans le cas de l'adaptation du réseau de TR avec regroupement CC, la valeur affichée correspond à la valeur du DER obtenue avec un réseau adapté après 250 époques. Pour des raisons de temps de calcul, l'expérience n'a pas été aussi précise que celles du chapitre 5, nous avons réalisé l'adaptation sur un seul réseau, et nous n'avons pas cherché à quantifier la variabilité du DER.

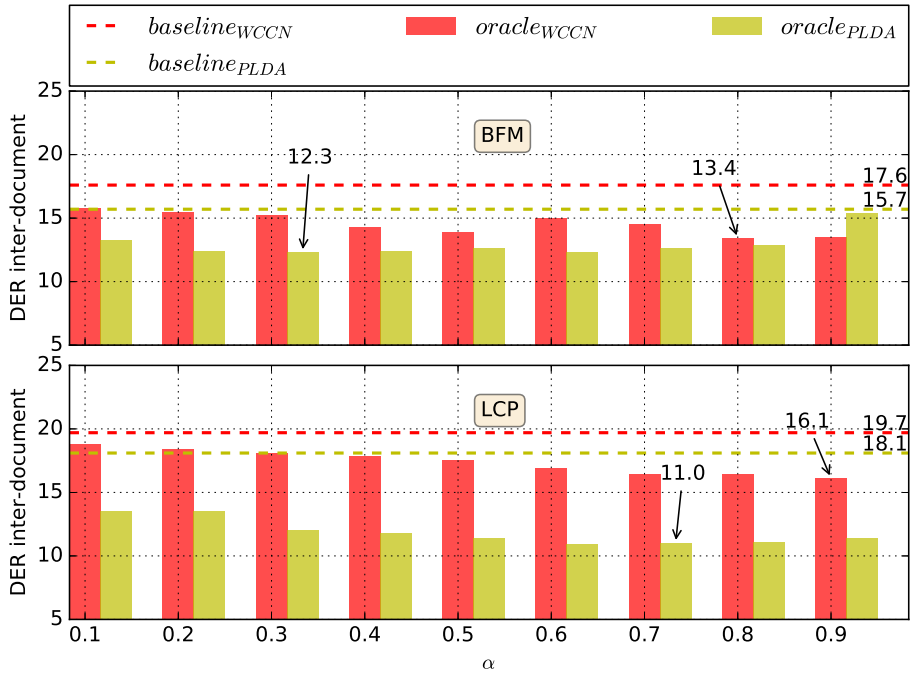

FIGURE 6.2 – Effet de l'adaptation au domaine *oracle* sur le DER inter-document des deux collections cibles. Comparaison des systèmes WCCN et PLDA avec un regroupement HAC, en fonction d'α, le coefficient d'adaptation.

Du côté du regroupement CC, sur la figure 6.3, les gains possibles varient de 8% à 51% en relatif. Que ce soit avec la similarité cosine/WCCN ou cosine/TR, les meilleurs gains sont atteints pour des valeurs d'α élevées ($\alpha = 0.8$ ou $\alpha = 0.9$, selon les cas), tandis qu'avec la PLDA, l'optimum se situe à 0.5. Remarquons que dans l'absolu, les meilleures performances sont obtenues avec les scores TR pour BFM et avec les scores PLDA pour LCP. Comme avec le regroupement HAC, le gain possible en adaptant la WCCN est plus limité et dépendant des performances de la *baseline* : pour LCP, la *baseline* étant à 22.6%, on arrive dans le meilleur des cas à 20.8%.

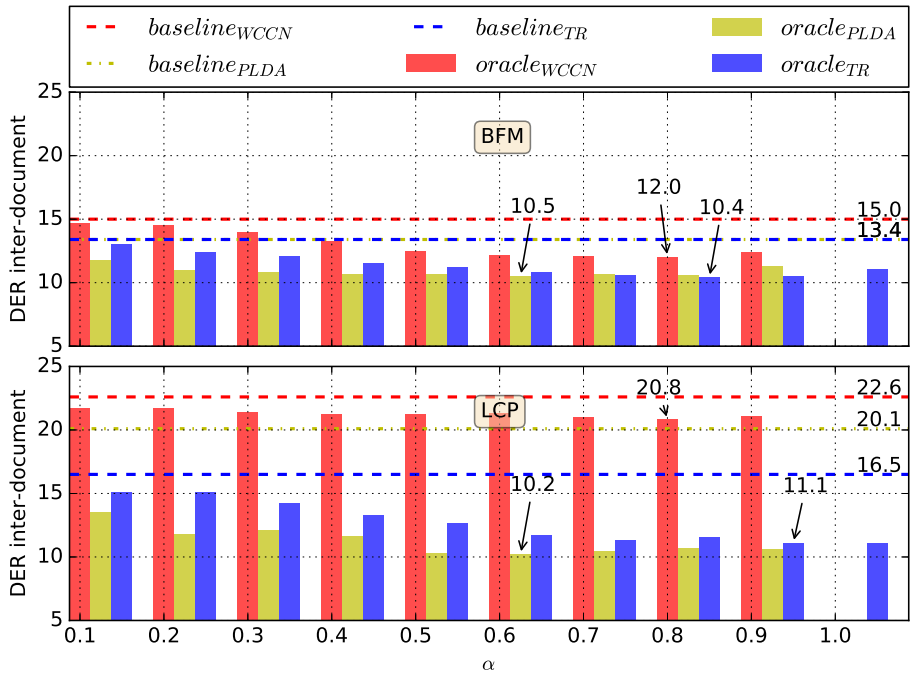

FIGURE 6.3 – Effet de l'adaptation au domaine *oracle* sur le DER inter-document des deux collections cibles. Comparaison des systèmes WCCN, PLDA et TR avec un regroupement CC, en fonction d'α, le coefficient d'adaptation.

Lorsqu'on compare les deux méthodes de regroupement, on constate que les meilleurs gains possibles le sont avec le regroupement CC. Pour les trois méthodes de calcul de similarités, les résultats montrent un gain possible de 8 à 50% grâce à l'adaptation. Notons que même si les valeurs optimales de α ne sont pas les mêmes selon les expériences, on observe une amélioration du DER quelle que soit la valeur du coefficient de pondération. L'adaptation de la WCCN donne les moins bons résultats du point de vue de la marge de progression, tandis que la PLDA affiche les gains possibles les plus élevés.

6.4.3 Adaptation itérative

L'expérience précédente nous ayant permis de valider le concept de l'adaptation, nous arrêtons maintenant d'utiliser les *i-vectors* extraits à partir de la segmentation de référence, et les remplaçons par les *i-vectors* cibles, extraits à partir de la SRL BIC intra-document. Après un premier regroupement inter-document, ces *i-vectors* sont regroupés en classes-locuteurs qui sont utilisées pour adapter les paramètres de calcul de similarités. Chaque expérience d'adaptation est caractérisée par un triplet $(\lambda_I, \lambda_X, \alpha)$, λ_I étant le seuil de regroupement intra-document, λ_X le seuil de regroupement inter-document et α le coefficient d'adaptation. Nous proposons également d'itérer le processus d'adaptation : après une $k^{ème}$ adaptation, les nouvelles classes-locuteurs sont utilisées pour adapter une $(k+1)^{ème}$ fois.

Dans les expériences suivantes, nous réalisons 4 itérations successives d'adaptation, et présentons les résultats obtenus pour les mêmes méthodes que les expériences *oracle* : regroupement HAC avec similarité cosine/WCCN ou PLDA, regroupement CC avec similarité cosine/WCCN, PLDA ou cosine/TR. Les plages de valeurs explorées sont les suivantes : de -80 à 20, avec un pas de 10 pour λ_I et λ_X, de 0.1 à 1, avec un pas de 0.1 pour α. Les seuils λ_I et λ_X s'appliquent sur des valeurs analogues à des distances, selon la même conversion distance/similarité que dans les expériences du chapitre 3.

6.4.3.1 Similarité PLDA

Regroupement HAC Concernant la PLDA avec le regroupement HAC, les résultats sont présentés en figures 6.4 et 6.5, pour 4 itérations successives d'adaptation. Ces figures représentent des paquets d'histogrammes, un paquet correspondant à une configuration d'adaptation $(\lambda_I, \lambda_X, \alpha)$. La première barre d'un paquet correspond au DER *baseline* et les 4 suivantes indiquent les DER après chaque itération d'adaptation, de la première à la quatrième ($iter_1$ à $iter_4$). La dernière barre, de couleur bleu clair ($iter_{4_{best}}$), indique le meilleur DER atteignable après la quatrième itération (si on allait lire le DER à un seuil $\lambda_{X'}$ différent de celui de l'expérience). L'intérêt de cette dernière valeur est de voir si la méthode proposée (garder un même seuil de regroupement à chaque itération) permet d'atteindre le meilleur DER possible après adaptation. Sont également indiquées sur les figures les performances *oracle* (trait en pointillés noirs) et les performances de la meilleure *baseline* (en pointillés bleus sur la figure 6.5). Sur chacune des figures qui suivront, la partie supérieure concerne la collection BFM et la partie inférieure concerne la collection LCP. Sur la partie inférieure de la figure 6.4, on peut donc lire l'information suivante : pour $\alpha = 0.5$ (cinquième paquet d'histogrammes), le DER inter-document *baseline* est de 18.1% et diminue jusqu'à atteindre 14.4% à l'itération $iter_4$. La valeur $iter_{4_{best}}$ est aussi à 14.4%, c'est-à-dire qu'on a atteint le meilleur DER possible après adaptation. Par ailleurs, le DER inter-document *oracle* dans cette configuration est de 11.4%.

La figure 6.4 présente les résultats comme une fonction d'α, à λ_I et λ_X fixés, tandis que la figure 6.5 les présente comme une fonction de λ_X, à α et λ_I fixés. Les figures permettent d'observer le voisinage de la configuration optimale de regroupement pour LCP ($\lambda_I = -10, \lambda_X = 10, \alpha = 0.5$) et BFM ($\lambda_I = 10, \lambda_X = 10, \alpha = 0.5$). Les résultats montrent que pour plusieurs triplets $(\lambda_I, \lambda_X, \alpha)$, l'adaptation itérative améliore progressivement le DER inter-document.

Pour LCP, le meilleur DER est atteint pour 4 itérations d'adaptation (14.4%), à partir d'une baseline à 18.1%, l'oracle étant à 11.3%. Pour BFM, le meilleur DER est de 13.6%, avec une baseline à 15.7% et un oracle à 12.6%. Les figures montrent que le meilleur gain en DER est obtenu à la première itération, suivi de gains plus faibles aux itérations suivantes. C'est particulièrement vrai pour LCP, où 2 ou 3

itérations sont parfois nécessaires pour converger. On peut également noter que la valeur optimale de α est proche de 0.5 pour les deux collections, et que le gain est plus faible à mesure que l'on s'approche de 0 ou 1. On remarque enfin que pour $\alpha = 0.5$, le DER après adaptation est égal au meilleur DER (barre bleue claire), pour les deux collections, ce qui valide l'approche qui consiste à conserver un même seuil de regroupement à chaque itération.

Quand on s'intéresse à la figure 6.5, on remarque que même à un seuil de regroupement inter-document sous-optimal (ie. $\lambda_X \neq \lambda_{X_{baseline}}$), la méthode proposée peut améliorer le DER initial et même battre le meilleur résultat *baseline*. C'est vrai lorsqu'on ne s'éloigne pas trop de la meilleure configuration. Sinon, on peut observer une légère dégradation du DER baseline (c'est visible sur la figure 6.5 pour BFM à la configuration $\lambda_X = -20$). Dans de rares cas, la convergence peut être chaotique (sur la figure 6.4, pour $\alpha = 0.2$ sur BFM et $\alpha = 0.3$ sur LCP).

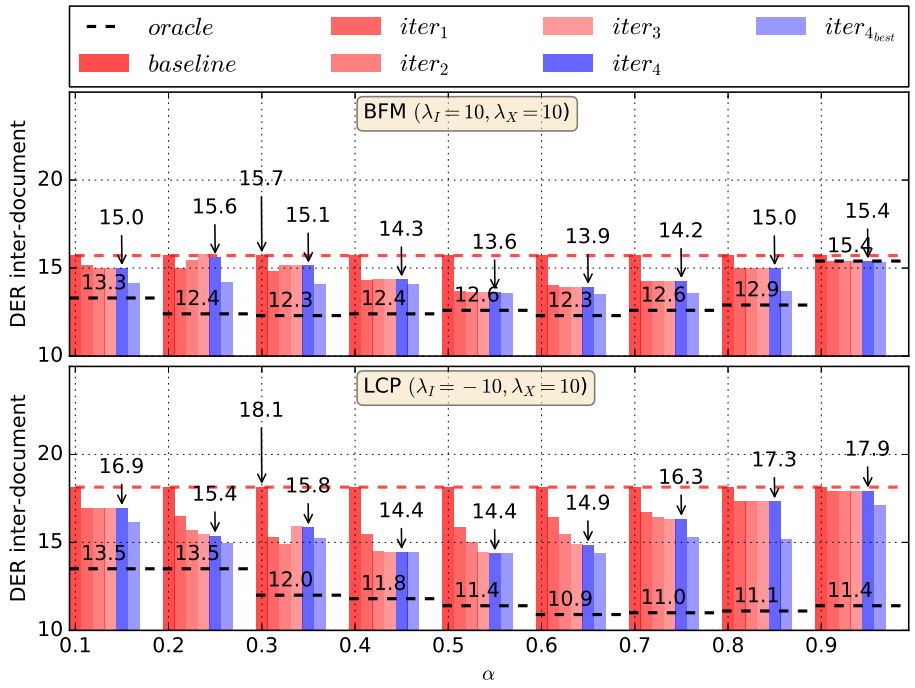

FIGURE 6.4 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction d'α, avec la similarité PLDA et le regroupement HAC.

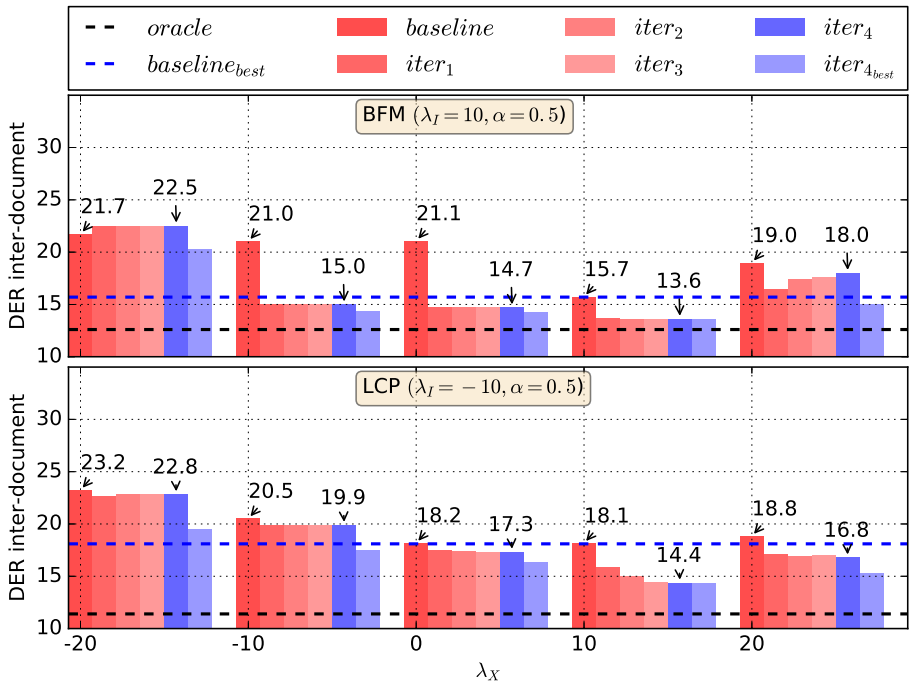

FIGURE 6.5 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction de λ_X, avec la similarité PLDA et le regroupement HAC.

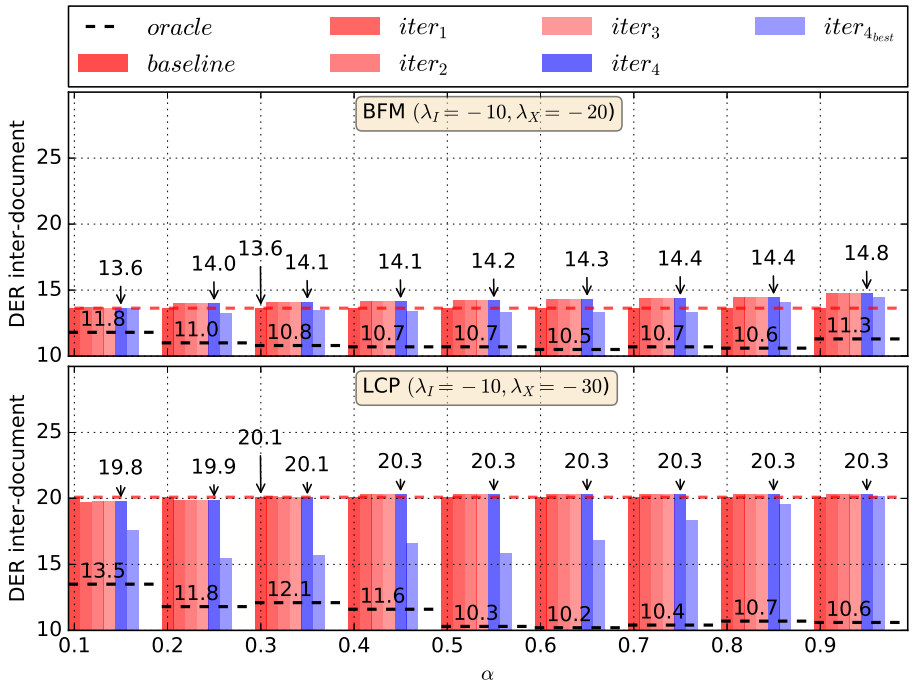

FIGURE 6.6 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction d'α, avec la similarité PLDA et le regroupement CC.

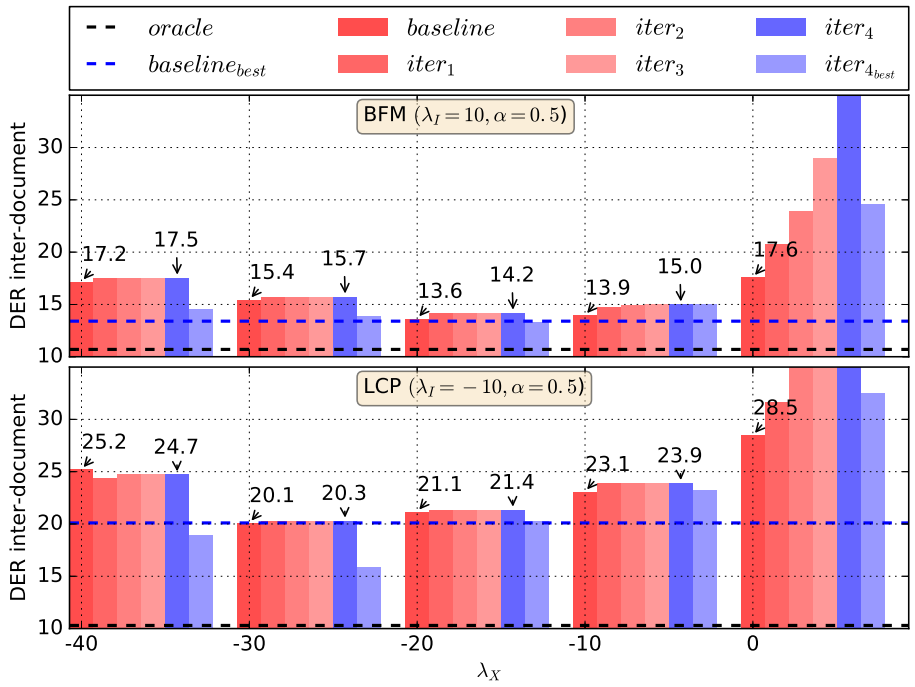

FIGURE 6.7 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction de λ_X, avec la similarité PLDA et le regroupement CC.

Regroupement CC Quand on compare avec le regroupement CC (figures 6.6 et 6.7, on n'observe pas ce phénomène de convergence itérative. D'abord, sur le corpus BFM, notons que le système *baseline* donne déjà de très bons résultats avec un DER à 13.6% (égal au meilleur DER obtenu après adaptation en utilisant le regroupement HAC). Ainsi, l'adaptation dégrade très légèrement le DER à la première itération puis stagne. Concernant LCP, l'expérience donne des résultats similaires : hormis une très légère amélioration pour $\alpha = 0.1$ ou $\alpha = 0.2$ sur la figure 6.6, où l'on passe de 20.1% à 19.8% d'erreur, on observe une légère dégradation puis stagnation aux itérations suivantes. Remarquons que le meilleur DER atteignable à la dernière itération ($iter_{4_{best}}$ à 15.3% pour $\alpha = 0.2$) est bien meilleur que celui obtenu par la méthode proposée ($iter_4$ à 19.9%). Il semblerait donc que l'adaptation puisse apporter un gain, mais nécessite d'utiliser un seuil de regroupement différent avant et après adaptation, ce qui demande une calibration supplémentaire.

Lorsqu'on s'intéresse à la robustesse au seuil (figure 6.7), le phénomène de légère dégradation puis stagnation est observé à la plupart des seuils. Remarquons également que lorsqu'on se place à un seuil qui dépasse l'optimum *baseline* ($\lambda_X = 0$), c'est-à-dire lorsqu'on a "trop regroupé", l'adaptation donne des résultats de plus en plus mauvais à chaque itération. Ce phénomène d'emballement est probablement dû à la composition des classes-locuteurs utilisées pour adapter : elles contiennent plusieurs locuteurs, ce qui fausse la modélisation lors de l'adaptation, et amène le système à faire plus d'erreurs à l'itération suivante. Ce phénomène est d'autant plus

marqué avec un regroupement CC (hiérarchique à saut minimal) : un regroupement associatif est beaucoup plus sensible aux erreurs qu'un regroupement HAC (hiérarchique à saut maximal), plus conservatif.

6.4.3.2 Similarité cosine/WCCN

Regroupement HAC Passons maintenant à la similarité cosine/WCCN avec le regroupement HAC. Les résultats sont présentés dans les figures 6.8 et 6.9. L'allure des histogrammes est similaire à l'approche HAC/PLDA, avec une amélioration graduelle du DER pour chaque valeur d'α et une certaine robustesse au seuil. Lors des expériences d'adaptation *oracle* (voir section 6.4.2), nous avions noté que l'adaptation de la WCCN donnait de meilleurs résultats pour des valeurs d'α élevées, c'est ce qu'on observe aussi lorsqu'on adapte avec les données réelles, sur la figure 6.8. Les meilleurs gains sont observés pour $\alpha = 0.8$ sur les deux collections, avec un DER après adaptation de 15.1% pour BFM et de 18.1% pour LCP, au niveau du meilleur DER atteignable après adaptation. Remarquons pour LCP une convergence légèrement instable à la troisième et quatrième itération, pour quelques valeurs de α. Pour $\alpha = 0.9$, on note même une dégradation du DER par rapport à la *baseline*.

Du côté de la figure 6.11, où l'on observe les performances pour $\alpha = 0.8$ en fonction du seuil de regroupement inter-document, on constate que le processus est efficace quel que soit le seuil, à condition de se placer avant l'optimum. Comme pour le système HAC/PLDA, l'adaptation peut améliorer le meilleur système *baseline*, même si on part d'un résultat obtenu à un seuil sous-optimal. Remarquons que pour $\lambda_X = -20$, sur le corpus LCP, l'adaptation dégrade légèrement le DER initial. Ceci va dans le même sens que les observations précédentes sur le système CC/PLDA, où l'adaptation à un seuil au-delà de l'optimal peut dégrader les performances.

Regroupement CC Concernant le regroupement CC, toujours avec la similarité cosine/WCCN, dont les performances sont présentées dans les figures 6.10 et 6.11, on constate dans la plupart des cas un léger gain en DER après adaptation. Dans la configuration ($\lambda_I = -50, \lambda_X = -50, \alpha = 0.6$), le DER passe de 15.0% à 14.7% pour le corpus BFM et de 22.9% à 22.2% pour le corpus LCP. Les performances sont au niveau du meilleur DER atteignable.

Encore une fois, comme on peut le voir sur la figure 6.10 qui présente les résultats en fonction du seuil λ_X, pour $\alpha = 0.6$, l'approche est robuste au seuil à condition de le positionner à une valeur inférieure à l'optimum, sans quoi le DER peut se dégrader (par exemple, pour le corpus BFM avec $\lambda_X = -40$).

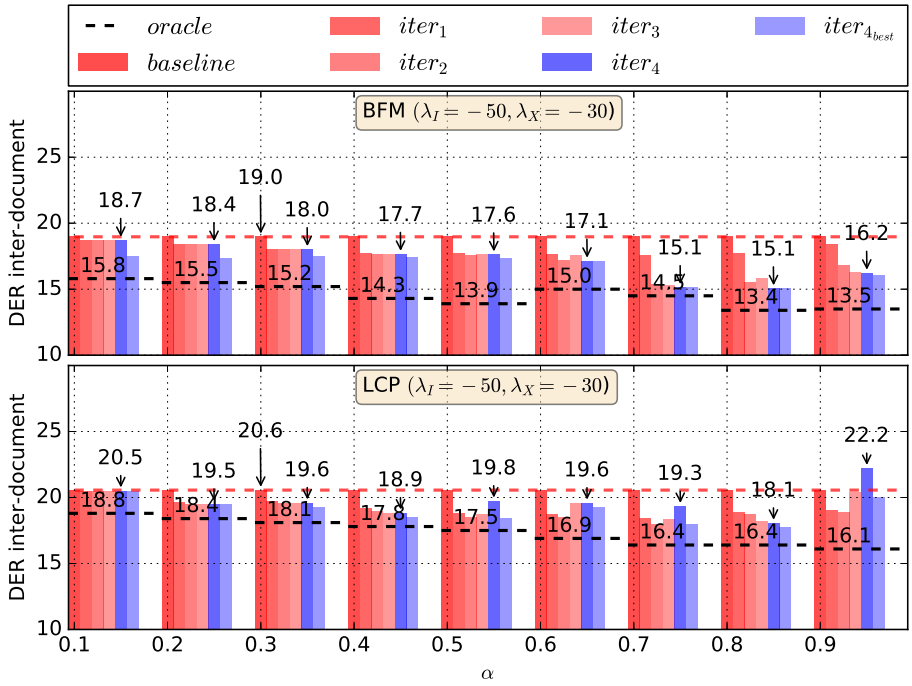

FIGURE 6.8 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction d'α, avec la similarité cosine/WCCN et le regroupement HAC.

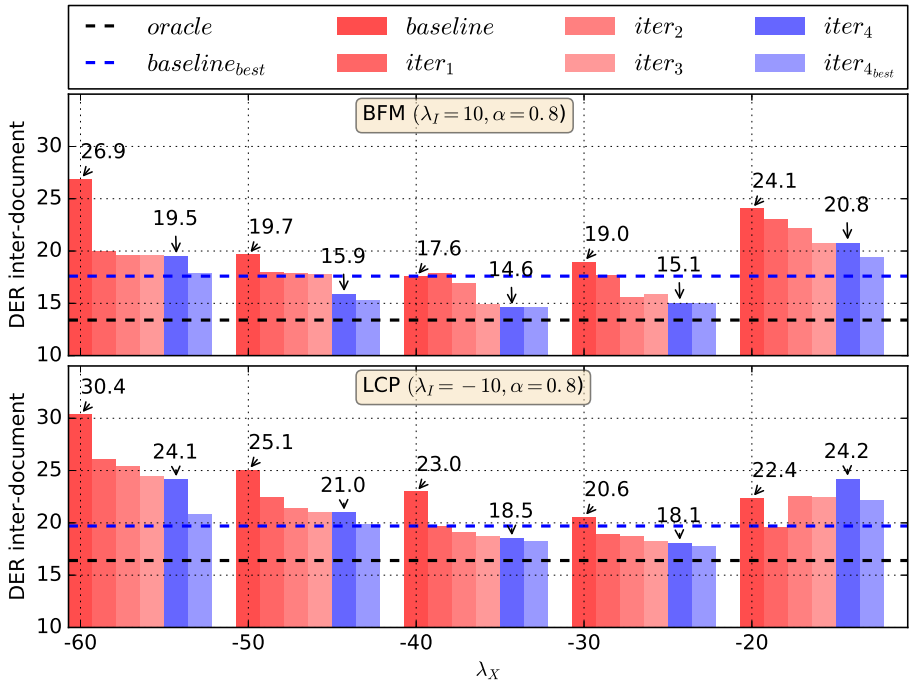

FIGURE 6.9 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction de λ_X, avec la similarité cosine/WCCN et le regroupement HAC.

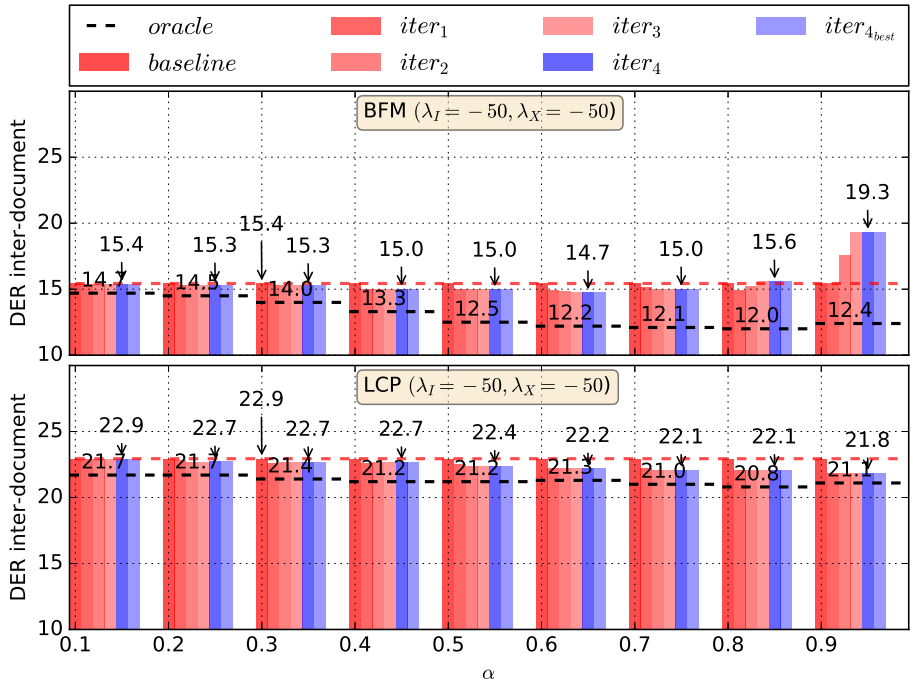

FIGURE 6.10 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction d'α, avec la similarité cosine/WCCN et le regroupement CC.

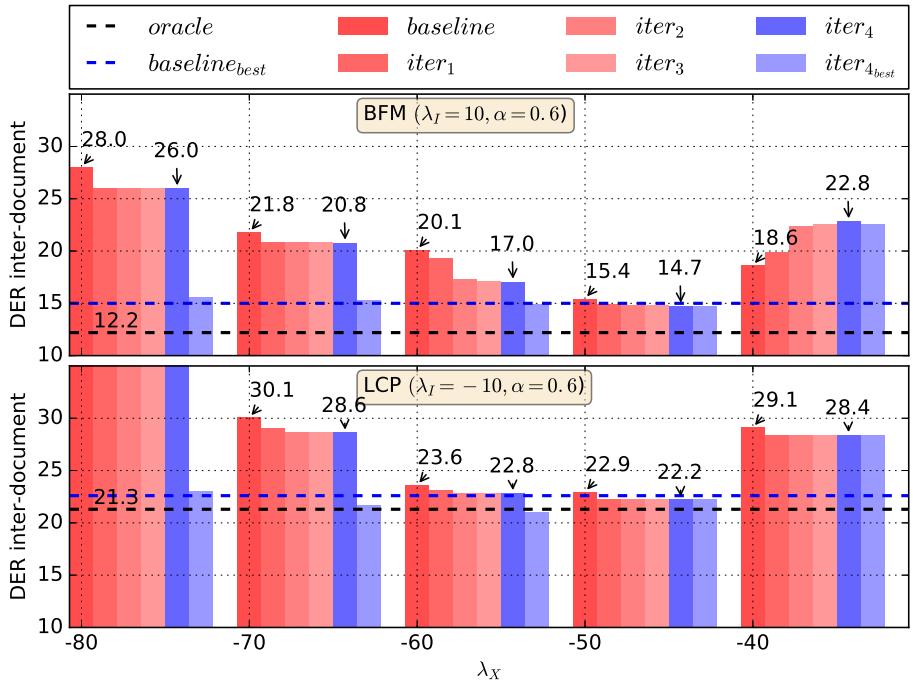

FIGURE 6.11 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction de λ_X, avec la similarité cosine/WCCN et le regroupement CC.

6.4.3.3 Similarité cosine/compensation neuronale TR

Les figures 6.12 et 6.13 présentent les résultats d'adaptation du réseau de neurones utilisé pour compenser la variabilité intra-locuteur/inter-document. Les paramètres d'adaptation sont les suivants : poursuite de l'apprentissage du réseau *baseline*, avec 250 époques d'apprentissage sur des triplets issus des données source et cible. C'est la configuration qui a été déterminée à l'annexe C, où on discute des stratégies d'adaptation du réseau de neurones. La pondération des données se fait au niveau de la fonction de coût. Pour générer les figures, l'expérience d'adaptation a été menée sur un réseau qui donne des performances proches de la moyenne des réseaux *baseline* testés au chapitre 5 : 13.4% de DER sur BFM et 16.5% de DER sur LCP (contre 13.3% et 16.6% respectivement, en moyenne). L'adaptation itérative se fait de la manière suivante : à chaque itération, l'apprentissage repart du réseau *baseline* (issu de 1500 époques d'apprentissage sur les données source) pour 250 époques, il ne repart donc pas du réseau adapté précédent.

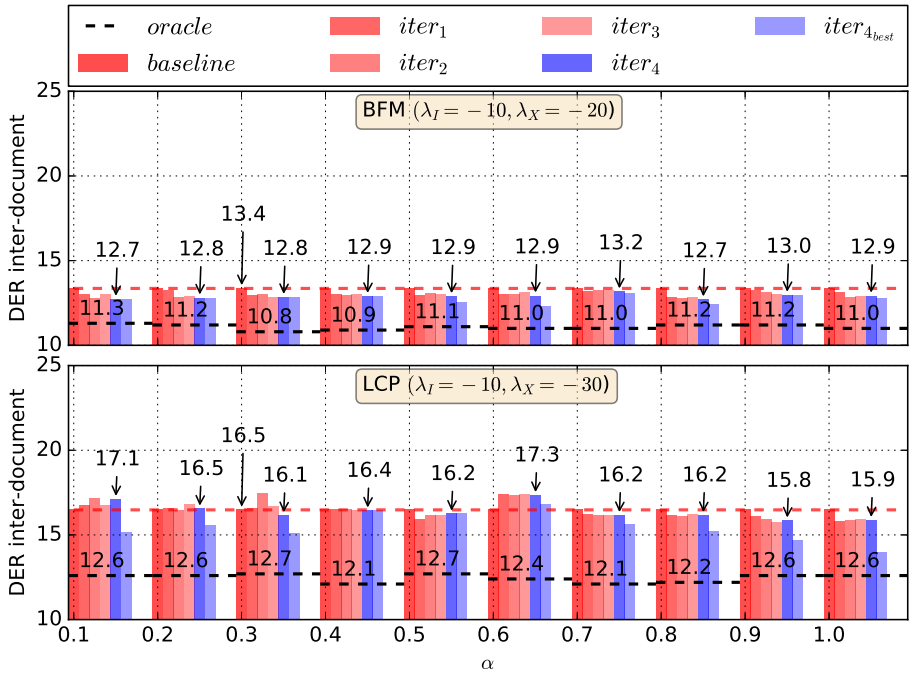

FIGURE 6.12 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction d'α, avec la similarité cosine/TR et le regroupement CC.

Comme on peut le voir sur la figure 6.12, il est possible d'améliorer les performances du système *baseline* en adaptant le réseau de neurones. Pour BFM, l'adaptation apporte un gain quelle que soit la valeur de α, avec dans le meilleur cas un DER à la quatrième itération à 12.7%, pour $\alpha = 0.8$, proche du meilleur DER atteignable. Concernant LCP, les résultats sont plus contrastés. L'adaptation fonctionne plutôt pour des valeurs de α élevées, avec un DER à 15.8% dans le meilleur cas,

pour $\alpha = 0.9$. Lorsqu'α est inférieur à 0.7, l'adaptation peut légèrement dégrader le DER pour l'amener à 17.3% dans le pire des cas ($\alpha = 0.6$). Si l'on s'intéresse aux meilleurs DER atteignables, on constate que pour $\alpha = 1$, il serait possible d'atteindre un DER d'environ 14% à condition d'utiliser un seuil de regroupement différent après la dernière itération d'adaptation (ce phénomène est étudié en annexe C).

Du côté de la figure 6.13, qui présente les résultats en fonction du seuil λ_X, pour $\lambda_I = -10$ et $\alpha = 0.8$, on observe toujours un léger gain possible au voisinnage de l'optimal, à l'exception du cas $\lambda_X = -50$ pour BFM, où le DER passe de 16.8% à 18.1%.

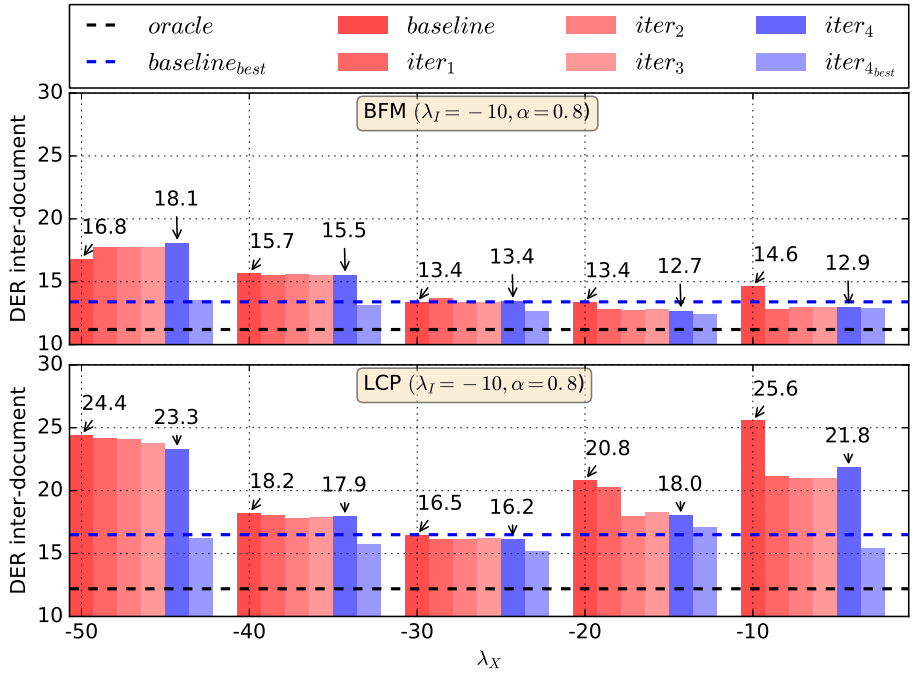

FIGURE 6.13 – DER inter-document sur les deux collections cibles, pour les itérations d'adaptation 0 (*baseline*) à 4, en fonction de λ_X, avec la similarité cosine/TR et le regroupement CC.

6.4.3.4 Récapitulatif des résultats

La table 6.2 récapitule les résultats obtenus pour différentes configurations. Les performances *adaptée* correspondent à la valeur du DER après 4 itérations d'adaptation. La configuration dédiée ($dediée_{collection}$) correspond au jeu de paramètres $(\lambda_I, \lambda_X, \alpha)$ optimal pour chaque collection, à méthode de regroupement et de calcul de similarités donnés, tandis que la configuration *commune* correspond à un jeu de paramètres qui optimisent le DER moyen des deux collections. Notons que les performances des systèmes *adaptée* peuvent être obtenus à partir d'une *baseline* qui n'est pas l'optimale, c'est-à-dire en utilisant un seuil d'adaptation λ_X qui n'est pas celui donnant le meilleur DER inter-document *baseline*.

similarité	WCCN				PLDA				TR	
regroupement	CC		HAC		CC		HAC		CC	
collection	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM
$baseline_{dediée_{LCP}}$	22.6	15.8	19.7	24.2	20.1	15.4	18.1	21.5	-	-
$baseline_{dediée_{BFM}}$	27.5	15.0	23.0	17.6	24.9	**13.4**	19.1	15.7	-	-
$baseline_{commune}$	22.9	15.4	20.6	19.0	21.2	13.6	19.1	15.7	**16.5**	**13.4**
$oracle_{best}$	20.8	12.0	16.1	13.4	10.2	10.5	11.2	12.3	12.1	11.3
$adaptée_{dediée_{LCP}}$	21.7	16.5	17.8	19.8	19.5	18.8	**14.4**	15.1	-	-
$adaptée_{dediée_{BFM}}$	22.2	14.7	18.5	14.6	25.9	13.5	**14.9**	13.4	-	-
$adaptée_{commune}$	22.2	14.7	18.5	14.6	20.8	13.6	14.9	13.4	**15.3**	**12.7**

TABLE 6.2: Récapitulatif des résultats d'adaptation sur les collections cibles complètes, avec les différentes mesures de similarités et regroupements.

Tout d'abord, on constate que pour la majorité des tuples (collection, regroupement, similarité), il existe une configuration d'adaptation itérative qui améliore le système *baseline*, que la configuration soit *dediée* ou *commune*. La seule exception concerne le regroupement CC combiné à la similarité PLDA, où les performances du système $baseline_{dediée_{BFM}}$ sont à 13.4% sur la collection BFM. Par ailleurs, on observe une très légère dégradation lors de l'adaptation, avec un DER de 13.5%. Par ailleurs, pour les similarités PLDA et WCCN associées au regroupement CC, on constate dans la configuration $dediée_{LCP}$ une dégradation des performances pour la collection BFM (de 15.8% à 16.5% et de 15.4% à 18.8%). Ceci ne s'observe pas pour le regroupement HAC, où lorsqu'on se place à la configuration dédiée d'une collection, l'adaptation améliore aussi les performances sur l'autre collection.

Le meilleur DER après adaptation est de 12.7% pour BFM, avec un regroupement CC et la similarité TR et de 14.4% pour LCP, avec un regroupement HAC et la similarité PLDA. Que ce soit pour l'approche cosine/WCCN ou PLDA, le choix de la méthode de regroupement semble avoir peu d'impact pour BFM. En revanche, il apparait clairement que le regroupement HAC donne de meilleurs résultats sur LCP que le regroupement CC. En somme, les meilleurs systèmes utilisant l'adaptation itérative sont donc les systèmes CC/TR et HAC/PLDA.

similarité	WCCN				PLDA				TR	
regroupement	CC		HAC		CC		HAC		CC	
collection	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM	LCP	BFM
$baseline_{commune}$	22.9	15.4	20.6	19.0	21.2	13.6	19.1	15.7	**16.5**	**13.4**
$adaptée_{@baseline}$	22.2	14.7	18.1	15.1	20.4	14.0	17.2	13.6	**15.8**	**13.3**
$adaptée_{commune}$	22.2	14.7	18.5	14.6	20.8	13.6	**14.9**	13.4	15.3	**12.7**

TABLE 6.3: Récapitulatif des résultats d'adaptation sur les collections cibles complètes, avec les différentes mesures de similarités et regroupements.

Dans la table 6.3, nous présentons les résultats de l'adaptation appliquée dans la même configuration (λ_I, λ_X) que la $baseline_{commune}$ (ligne $adaptée_{@baseline}$). Dans ce cas, le seul paramètre supplémentaire à fixer par rapport à la configuration *baseline* est le coefficient d'adaptation α. On voit alors qu'on gagne à adapter, même si l'amélioration n'est pas aussi importante qu'en s'autorisant à utiliser un seuil d'adapta-

tion λ_X différent de celui donnant les meilleures performances *baseline*. Dans ces conditions, le système d'adaptation donnant les meilleures performances dans des conditions de calibration minimales est le système TR/CC, avec un DER à 13.3% pour BFM et 15.8% pour LCP, suivi du système PLDA/HAC, avec des DER à 13.6% et 17.2%, respectivement. Remarquons cependant que le système TR/CC donnait les meilleures performances *baseline*, par conséquent, si l'on s'intéresse à la progression relative du DER par rapport à la *baseline*, c'est le système PLDA/HAC qui l'emporte.

6.4.4 Analyse en locuteurs

Cette section est dédiée à l'analyse des résultats d'adaptation. Comme on connait la composition en locuteurs des collections cibles, on peut étudier l'évolution des classes-locuteurs à travers l'adaptation itérative : est-ce que le gain en DER est dû à une précision accrue sur les locuteurs uniques ou à la capacité à mieux regrouper les prises de parole des locuteurs récurrents ?

6.4.4.1 Collection LCP, système PLDA/HAC

Nous avons sélectionné une expérience présentée à la section 6.4.3 : adaptation itérative sur la collection LCP complète, utilisant la PLDA pour le calcul des similarités et les paramètres ($\lambda_I = -10, \lambda_X = 10, \alpha = 0.5$). Sur la figure 6.14, on représente cette expérience du point de vue de l'attribution de parole d'une itération à une autre : cela permet de visualiser les contributions aux variations du DER inter-document.

La première colonne montre les locuteurs pour lesquels on observe une variation d'attribution de temps de parole d'une itération à une autre, la deuxième indique le nombre de documents dans lesquels le locuteur apparait, la troisième correspond au rôle de la personne (invité ou journaliste), et les quatre colonnes de droite décrivent les variations d'attribution de parole au cours des itérations (par exemple, la $3^{ème}$ colonne ($0 \rightarrow 1$) indique la différence d'attribution de parole entre le regroupement *baseline* et celui de la première itération d'adaptation). Chaque cellule est colorée, selon que l'itération permet de retrouver (en bleu) ou perdre (en rouge) du temps de parole du locuteur (la durée de parole retrouvée ou perdue est écrite dans la cellule). Par ailleurs, la hauteur de chaque ligne est proportionnelle au logarithme du temps de parole total de son locuteur. Une façon de lire la figure est, par exemple : la deuxième ligne concerne Germain Andrieux, un journaliste qui parle dans 11 épisodes de l'émission. Après la première itération d'adaptation, 198 secondes de parole ont été retrouvées (ie. la classe-locuteur attribuée à Germain Andrieux a gagné 198 secondes de parole de Germain Andrieux), tandis que 151 secondes supplémentaires l'ont été à la seconde itération, pour un total de 349 secondes.

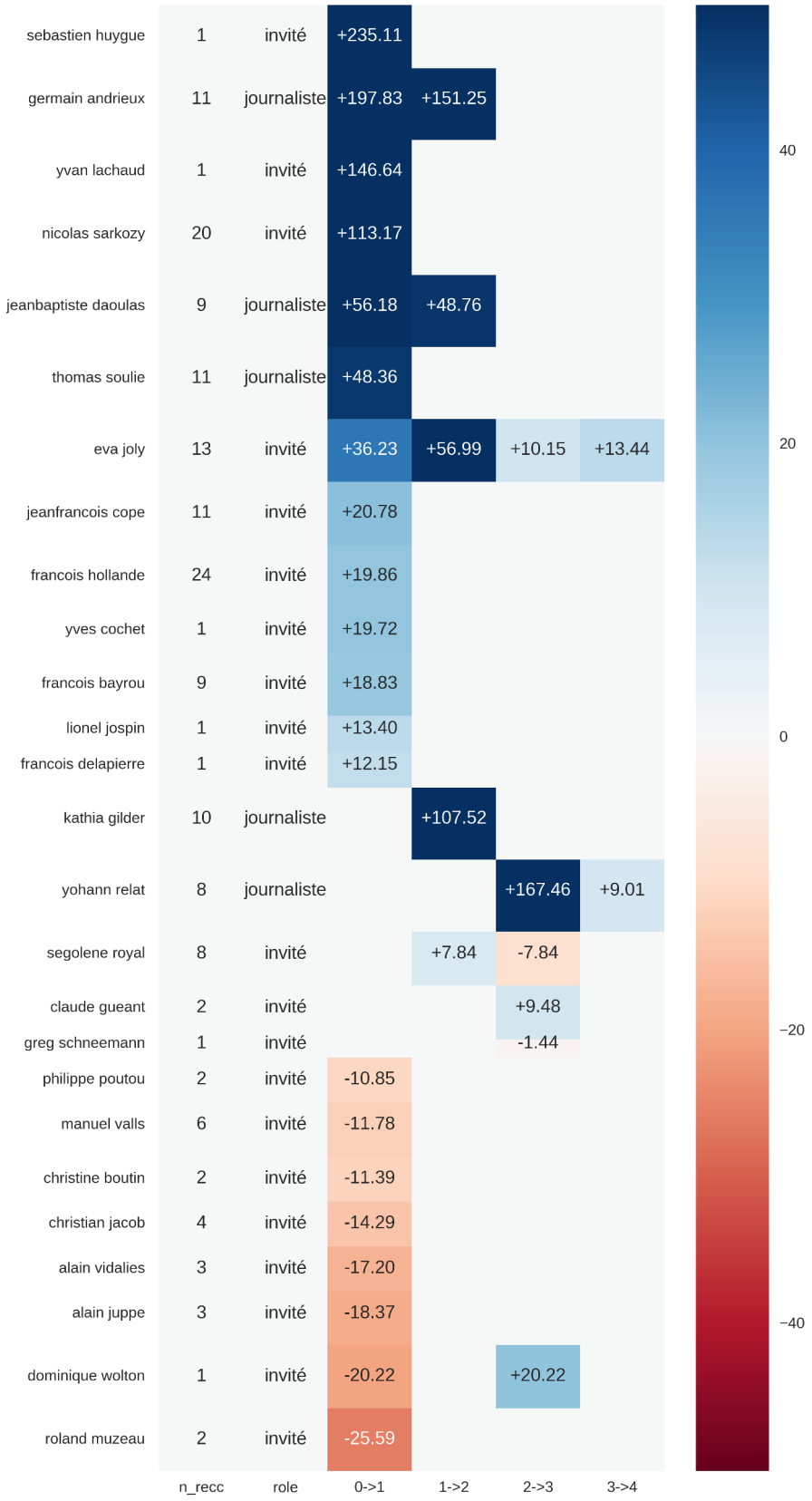

FIGURE 6.14 – Analyse de l'évolution de l'attribution de parole correcte pendant le processus d'adaptation itérative. L'expérience est réalisée sur la collection LCP, avec un regroupement HAC et les scores PLDA, pour les paramètres ($\lambda_I = -10, \lambda_X = 10, \alpha = 0.5$). De l'itération $iter_0$ (*baseline*) à $iter_4$, le DER inter-document varie de 18.1% à 14.4%. Les locuteurs **ponctuels** sont ceux pour qui $n_{recc} = 1$, les **récurrents** sont ceux pour qui $n_{recc} > 1$.

Le premier constat est qu'on peut distinguer trois types de locuteurs : ceux pour qui l'adaptation permet de retrouver de la parole correcte en plusieurs itérations (par exemple, Eva Joly à la $7^{ème}$ ligne), ceux qui gagnent ou perdent de la parole en une seule itération, généralement la première (par exemple Nicolas Sarkozy à la $4^{ème}$ ligne), et ceux qui gagnent puis perdent (ou l'inverse) la même durée de parole au cours de deux itérations différentes (par exemple, Ségolène Royal à la $16^{ème}$ ligne).

Lorsqu'on s'intéresse à la récurrence des locuteurs (le nombre de documents où ils apparaissent), on peut noter que même des locuteurs ponctuels sont affectés par le processus d'adaptation, en général au cours de la première itération. On remarque également que la plupart des locuteurs qui gagnent du temps de parole au cours de plusieurs itérations sont des locuteurs récurrents. Ceci indique que l'adaptation améliore bien la modélisation de la variabilité des locuteurs récurrents, ce qui était la motivation initiale pour itérer l'adaptation.

Enfin, on constate que la contribution principale à la réduction du DER inter-document est due aux locuteurs récurrents. Pour la collection LCP, on sait qu'environ 80% du temps de parole total concerne des locuteurs récurrents. La même expérience a été représentée sous forme d'histogramme détaillant l'évolution des taux d'erreur par type de locuteur au cours des itérations d'adaptation, à la figure 6.15. La figure confirme que les gains sont majoritairement obtenus sur les locuteurs récurrents.

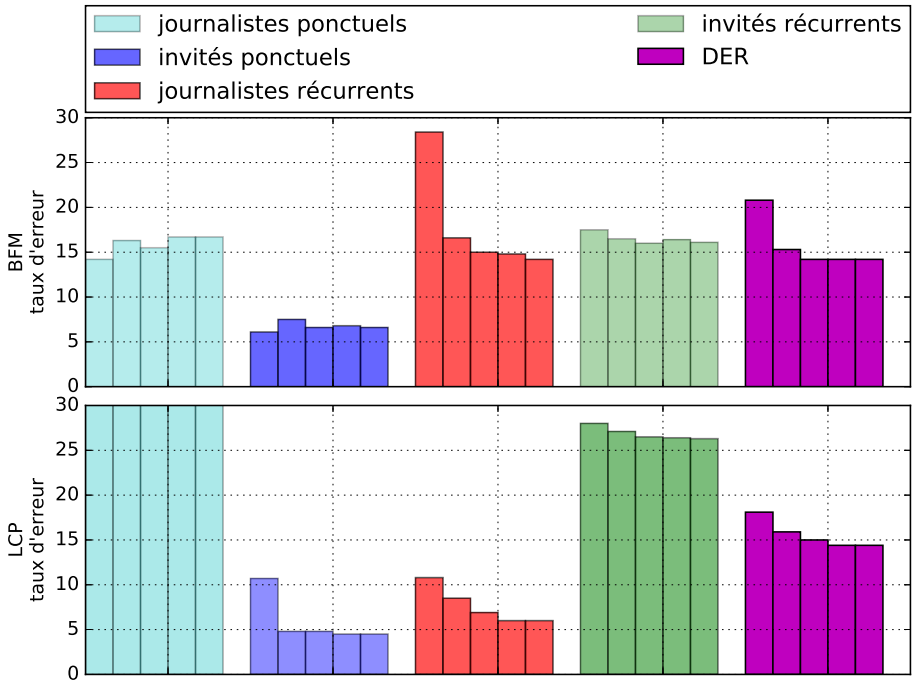

FIGURE 6.15 – Evolution des taux d'erreur classe lors de l'adaptation du système HAC/PLDA. Chaque barre d'histogramme correspond à une itération d'adaptation (de 0 pour la *baseline* à 4 pour la quatrième itération d'adaptation). Pour la collection BFM, $(\lambda_I = -10, \lambda_X = 0, \alpha = 0.5)$, pour LCP, $(\lambda_I = -10, \lambda_X = 10, \alpha = 0.5)$.

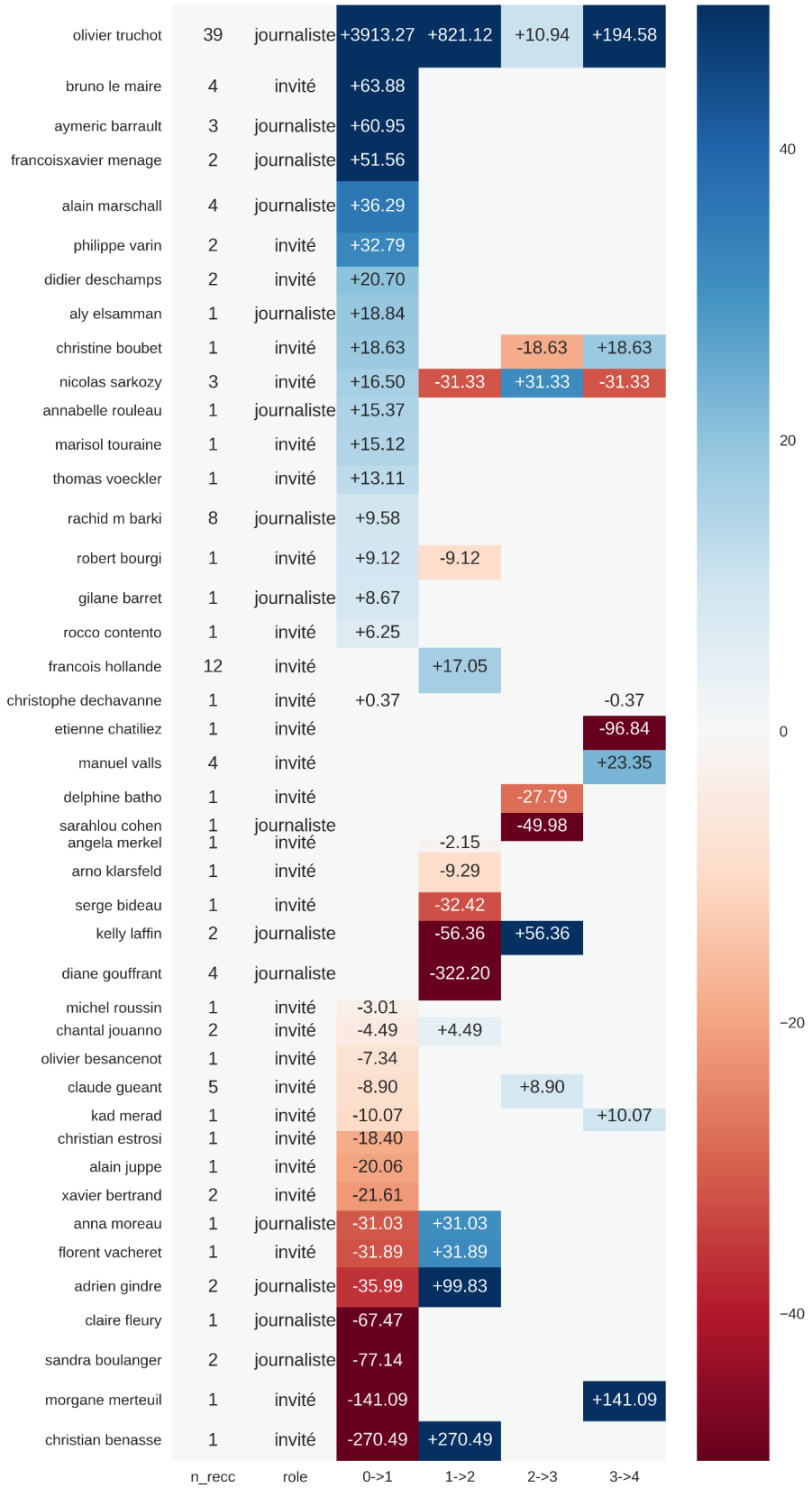

FIGURE 6.16 – Analyse de l'évolution de l'attribution de parole correcte pendant le processus d'adaptation itérative. L'expérience est réalisée sur la collection BFM, avec un regroupement HAC et les scores PLDA, pour les paramètres $(\lambda_I = -10, \lambda_X = 0, \alpha = 0.5)$. De l'itération $iter_0$ (*baseline*) à $iter_4$, le DER inter-document varie de 20.8% à 14.2%.

6.4.4.2 Collection BFM, système PLDA/HAC

En ce qui concerne la collection BFM, nous avons choisi de visualiser une expérience dont la *baseline* n'est pas l'optimal. En effet, compte-tenu des bonnes performances avant adaptation et de la faible évolution relative du DER dans le meilleur cas (de 15.7% à 13.6%), choisir un cas sous-optimal permet d'observer un plus grand nombre de phénomènes évolutifs. L'expérience choisie voit son DER varier de 20.8% à 14.2%. Elle est représentée sur la figure 6.16, avec les mêmes critères de visualisation par locuteur que pour l'expérience décrite à la section précédente. L'évolution des taux d'erreur par type de locuteur est également illustrée par la figure 6.15.

Le constat est similaire pour chaque type de locuteur, mais cette fois il n'y a qu'un locuteur récurrent pour lequel le processus permet de récupérer du temps de parole correct au cours des itérations successives : c'est le locuteur principal de l'émission, Olivier Truchot. Pour la plupart des autres locuteurs, on constate une variation lors d'une seule itération, avec parfois un effet d'alternance entre une même durée de parole retrouvée et de parole perdue à deux adaptations différentes (c'est particulièrement visible pour Nicolas Sarkozy à la $9^{ème}$ ligne). Il s'agit généralement d'un *i-vector* qui est tantôt associé correctement à son locuteur, tantôt rattaché à un mauvais locuteur, voire à aucun locuteur. Ces résultats ne sont pas trop surprenants, compte tenu du fait que même si 55% du temps de parole de l'émission concerne des locuteurs récurrents, le locuteur principal en accapare à lui seul quasiment la moitié. Le nombre de locuteurs affectés par l'adaptation est assez équilibré entre les locuteurs pour qui l'adaptation dégrade les résultats, et ceux pour qui elle les améliore, et, *in fine*, la baisse du DER indique que la durée de parole correctement affectée a progressé au cours de l'adaptation.

6.4.4.3 Même DER, comportement différent ?

Au cours des expériences, nous avons constaté, sur la collection BFM, des DER assez proches pour des systèmes de SRL n'utilisant pas le même type de regroupement ou de similarités. Nous proposons donc d'étudier comparativement les systèmes du point de vue du taux d'erreur nominal et taux d'erreur classe. La figure 6.17 présente les taux nominaux d'erreur par type de locuteur, pour les deux systèmes suivants : CC/TR *baseline* et HAC/PLDA après 4 itérations d'adaptation. Respectivement, le DER inter-document des deux systèmes est de 13.4% et 13.6%.

Si les DER sont très proches, le comportement des deux systèmes varie selon le type de locuteur considéré. Comparativement, le système CC/TR est bien meilleur sur les locuteurs ponctuels, avec 6.2% contre 15.2% d'erreur classe pour les journalistes et 5.0% contre 7.4% pour les invités. Le système CC/TR préserve bien les locuteurs ponctuels, parmi lesquels seulement une quinzaine d'individus affichent un taux d'erreur de 100%.

Comme nous avons pu le constater dans les analyses précédentes, l'apport principal de l'adaptation concerne les locuteurs récurrents, en particulier les journalistes : le système HAC/PLDA adapté l'illustre bien, avec un taux d'erreur classe de 11.4%, contre 12.6% avant adaptation (non représenté sur la figure). Le système CC/TR affiche de son côté un taux d'erreur classe de 14.3% : ses performances sur les différents types de locuteurs sont plus équilibrées, même si c'est au détriment des journalistes récurrents.

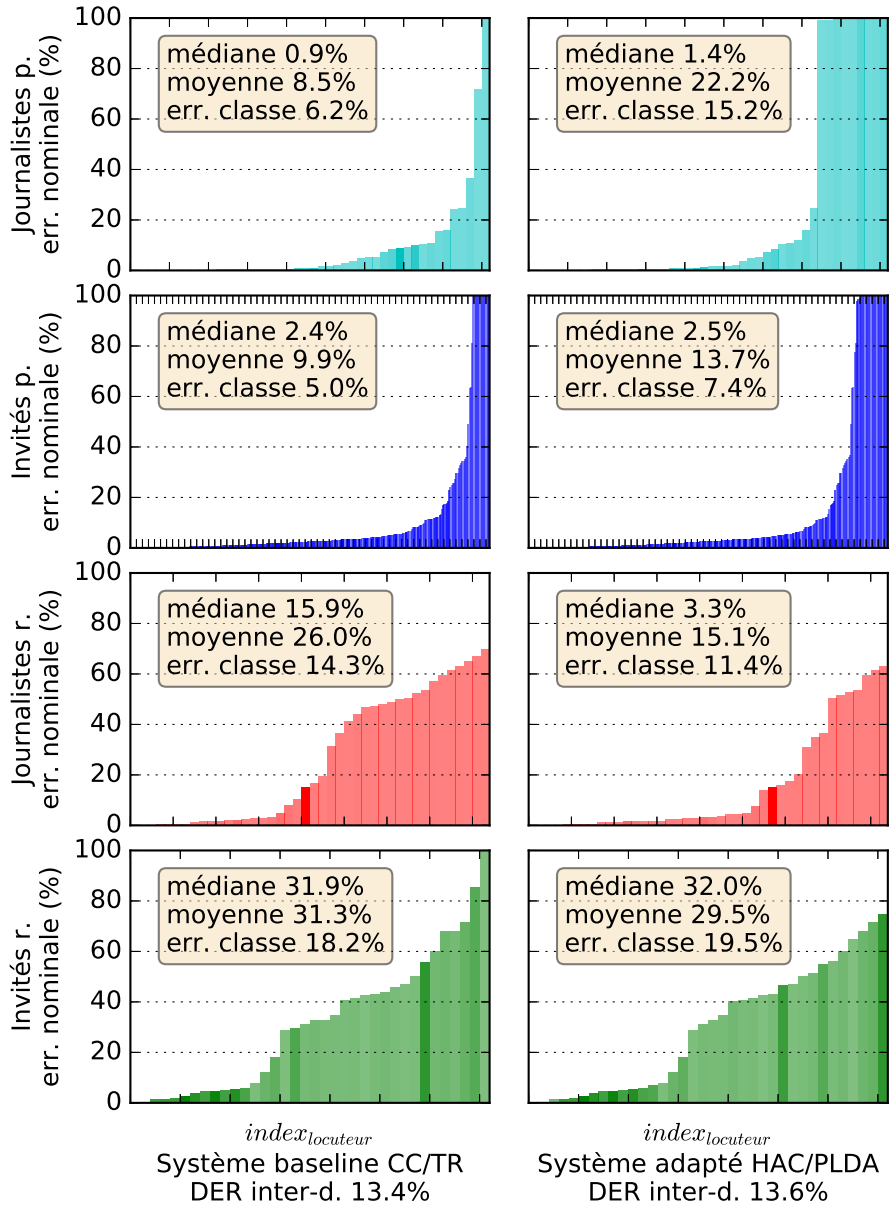

FIGURE 6.17 – Analyse d'erreur par type de locuteur. Comparaison du système HAC/PLDA après 4 itérations d'adaptation ($\lambda_I = 10, \lambda_X = 10, \alpha = 0.5$) et du système *baseline* CC/TR ($\lambda_I = -10, \lambda_X = -30$).

6.5 Conclusions

Dans ce chapitre, nous avons proposé une méthode d'adaptation itérative pour différents modèles de compensation de variabilité intra-locuteur/inter-document, dans le cadre de la SRL de collection. L'approche consiste à utiliser les classes-locuteurs produites par un système de SRL, entraîné sur un corpus d'un domaine source différent des collections cibles, pour adapter les paramètres de calcul de similarité entre *i-vectors*. La philosophie de l'approche est d'améliorer progressivement la qualité du regroupement inter-document au fil des itérations d'adaptation, en exploitant la connaissance partielle de la variabilité intra-locuteur/inter-document des données cibles.

Différentes méthodes d'adaptation ont été proposées pour les modèles de compensation WCCN, PLDA et TR, et testées avec les méthodes de regroupement HAC ou CC sur deux collections d'une quarantaine d'épisodes. Ces différentes méthodes ont en commun le fait d'utiliser une pondération entre données sources et cibles pour adapter les modèles de compensation de variabilité, à l'aide d'un coefficient α. Cette pondération permet de doser la confiance à accorder aux données cibles, dont les classes-locuteurs peuvent contenir des erreurs, et aux données sources, annotées manuellement, mais qui ont l'inconvénient de ne pas être du même domaine que les données des collections qu'on cherche à traiter.

Les résultats montrent que l'approche proposée fonctionne et diminue le DER pour la majorité des configurations testées, à l'exception du système CC/PLDA. L'aspect itératif permet d'améliorer progressivement le DER inter-document dans certaines configurations (particulièrement celles utilisant le regroupement HAC), et en général, la baisse la plus importante est obtenue à la première itération. Le système de SRL étudié dépend d'un seuil λ_X pour le regroupement inter-document, et les résultats indiquent que pour les trois méthodes de compensation de variabilité, l'adaptation est robuste au choix du seuil, c'est-à-dire qu'elle peut apporter un gain même si le système de SRL initial ne donne pas les résultats optimaux. C'est un résultat important quand on sait que le choix du seuil peut varier selon le type de locuteur pour lequel on cherche à optimiser les performances. Les deux approches donnant les meilleurs résultats sont les approches d'adaptation de PLDA avec regroupement HAC et d'adaptation TR avec regroupement CC.

Chapitre 7

Taille des collections et adaptation

Résumé

Ce chapitre complète le précédent en posant la question de l'influence de la taille des collections sur l'optimalité des paramètres d'adaptation. Jusqu'ici, le système de SRL a été évalué sur deux collections d'une quarantaine de documents. A partir de sous-collections de différentes tailles, nous étudions l'optimalité des paramètres d'adaptation des systèmes HAC/WCCN, HAC/PLDA et CC/TR. Les résultats montrent une dépendance de l'optimalité du coefficient d'adaptation α à la taille de la collection traitée, pour les systèmes HAC/WCCN et HAC/PLDA. Ceci nous amène à proposer une formule paramétrique, afin de rendre le coefficient dépendant de la taille de la collection sur laquelle est réalisée ladite adaptation. La taille est exprimée en nombre de locuteurs récurrents. Nous définissons également le DER moyen pondéré, qui permet de rendre compte des performances de SRL sur l'ensemble des sous-collections d'une collection donnée. Les expériences montrent que si adapter sur des sous-collections de trop petite taille dégrade le DER, le choix d'une approche paramétrique peut éviter une telle dégradation.

7.1 Introduction

Dans les chapitres précédents, nous avons proposé une nouvelle méthode de compensation de variabilité intra-locuteur/inter-document, dite TR, pour le calcul des similarités entre segments-locuteurs (au chapitre 5). Nous avons également proposé une stratégie d'adaptation itérative, qui consiste à exploiter les données de la collection pour mettre à jour les modèles de compensation de variabilité et de réduire les taux d'erreur de SRL (au chapitre 6). La stratégie s'est révélée efficace sur deux collections d'une quarantaine d'épisodes, en réduisant le DER inter-document.

Les résultats du chapitre précédent sont cependant incomplets, puisqu'ils ne traitent pas de la question de la taille des collections. Si les paramètres d'adap-

tation optimaux ont été bien identifiés pour les trois méthodes de compensation de variabilité (WCCN, PLDA, TR), on n'a aucune certitude quant à leur efficacité pour traiter des collections d'une dizaine d'épisodes seulement, ou au contraire de plusieurs centaines. Or, les auteurs de [Garcia-Romero et al., 2014] ont montré que le coefficient optimal d'adaptation d'un modèle 2-covariance dépendait du nombre de locuteurs utilisés pour l'adaptation, sur la tâche de reconnaissance du locuteur.

Dans ce chapitre, nous abordons la problématique de l'influence de la taille d'une collection sur l'adaptation : faut-il en faire varier les paramètres, peut-on adapter sur de petites collections ? Au chapitre précédent, nous avons montré que l'adaptation au domaine pouvait améliorer les performances de SRL sur deux collections différentes. Les expériences ont été menées sur des collections de 42 et 45 documents, en utilisant un coefficient d'adaptation α fixé empiriquement. Nous allons maintenant étudier le passage à l'échelle de la méthode. En effet, si un α de 0.5 donne de bonnes performances pour environ 40 épisodes avec un regroupement HAC et la similarité PLDA, nous devons étudier si la taille de la collection a une influence sur le paramètre optimal. Par exemple, une petite collection ne peut contenir qu'un faible nombre de locuteurs récurrents, dont la contribution à l'adaptation des paramètres pourrait être mauvaise. Dans ce cas, il serait peut-être plus sage de donner plus de poids aux données d'apprentissage initiales qu'à un nombre limité de classes-locuteurs cibles. Les expériences d'adaptation du chapitre 6 ont montré des zones d'optimalité du coefficient d'adaptation α variables selon les méthodes de calcul de similarités : autour de 0.5 pour la PLDA, autour de 0.8 pour la WCCN et proche de 1 pour la similarité TR.

7.2 Propositions

7.2.1 Passage à l'échelle

Notre système de SRL présenté à la section 6.3 dépend d'un triplet de paramètres $(\lambda_I, \lambda_X, \alpha)$. λ_I et λ_X sont les seuils de décision des regroupements intra- et inter-document, α le paramètre d'adaptation. Les collections cibles sont généralement stockées dans l'ordre chronologique de diffusion et leur taille augmente donc au fil du temps. En fonction de la taille des collections cibles, il pourrait être utile de leur donner plus ou moins de poids dans le processus d'adaptation. Si on adapte les paramètres avec seulement quelques épisodes, le nombre de classes-locuteurs récurrentes risque d'être trop faible. Il faut alors envisager de limiter la contribution des données cibles pour éviter d'adapter de manière imprécise.

Au lieu de définir α de façon empirique, nous proposons de le rendre dépendant de la taille de la collection cible, avec une formule inspirée de l'adaptation MAP [Reynolds et al., 2000]. Dans le cas du calcul de la WCCN ou de l'estimation de

la PLDA, le nombre de classes-locuteurs et de sessions sont des facteurs clés. Dans [Garcia-Romero and McCree, 2014], les auteurs, qui travaillent sur l'adaptation du modèle 2-covariance, montrent que la valeur optimale d'α dépend du nombre de locuteurs utilisés pour l'adaptation. Ainsi, on souhaite qu'α soit proche de 0 (respectivement 1) quand le nombre de classes-locuteurs disponibles pour l'adaptation est faible (resp. élevé), ce qui nous amène à proposer la formule suivante :

$$\alpha = \frac{{S_{cible}}^p}{{S_{cible}}^p + r^p} \tag{7.1}$$

S_{target} correspond au nombre de locuteurs récurrents (c'est-à-dire les classes-locuteurs qui contiennent des segments provenant d'au moins 3 documents cibles différents). On appelle r la taille *virtuelle* du corpus d'apprentissage source, en nombre de locuteurs récurrents, et p est un coefficient d'ajustement. La notion de taille *virtuelle* en locuteurs récurrents correspond au nombre de locuteurs récurrents de la collection cible pour lequel le poids des données source et cible est équivalent ($\alpha = 0.5$), quel que soit le nombre réel de locuteurs récurrents du corpus d'apprentissage. Si on positionne p à 1 et r au nombre réel de locuteurs récurrents du corpus source, l'adaptation proposée est équivalente à apprendre la PLDA de manière classique sur la concaténation des données source et cible, moyennant une approximation. En effet, apprendre la PLDA adaptée consiste à maximiser la vraisemblance (rappel de la section 6.2.2.1) :

$$L(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \alpha L_{cible}(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) + (1-\alpha) L_{source}(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) \tag{7.2}$$

Avec

$$L_k(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \frac{1}{N_k} \sum_{s=1}^{S_k} \sum_{j=1}^{n_{ik}} log(p(\boldsymbol{\phi_{ij}} | \boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda})) \tag{7.3}$$

En notant $N = N_{cible} + N_{source}$ le nombre total d'*i-vectors*, on a

$$\alpha = \frac{N_{cible}}{N_{cible} + N_{source}} = \frac{N_{cible}}{N} \iff 1 - \alpha = \frac{N_{source}}{N} \tag{7.4}$$

$$\iff L(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \frac{N_{cible}}{N} \frac{1}{N_{cible}} (\sum_{cible} \dots) + \frac{N_{source}}{N} \frac{1}{N_{source}} (\sum_{source} \dots) \tag{7.5}$$

$$\iff L(\boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda}) = \frac{1}{N} \sum_{i=1}^{S} \sum_{j=1}^{n_i} log(p(\boldsymbol{\phi_{ij}} | \boldsymbol{\Phi\Phi^T}, \boldsymbol{\Lambda})) \tag{7.6}$$

C'est la formule de maximisation de la vraisemblance sur un jeu de N *i-vectors* d'apprentissage sur $S = S_{cible} + S_{source}$ locuteurs récurrents, introduite à la section 3.2.6. Avec la formule d'adaptation proposée (équation 7.1), en considérant que

$r = S_{source}$ et que

$$\frac{S_{cible}}{S_{cible} + S_{source}} \approx \frac{N_{cible}}{N_{cible} + N_{source}} \tag{7.7}$$

On se ramène au cas classique d'apprentissage de la PLDA sur la concaténation des données source et cible[1]. En pratique, on a donc intérêt à positionner r en fonction du nombre de locuteurs récurrents de la collection cible pour booster sa contribution par rapport à un apprentissage classique sur la concaténation des données source et cible.

Quelques courbes décrivant la formule proposée sont présentées en figure 7.1. La figure représente l'évolution d'α, en fonction du nombre de locuteurs (ou classes-locuteurs) récurrents, pour $r = 20$ ou $r = 60$ et p variant de 1 à 3.

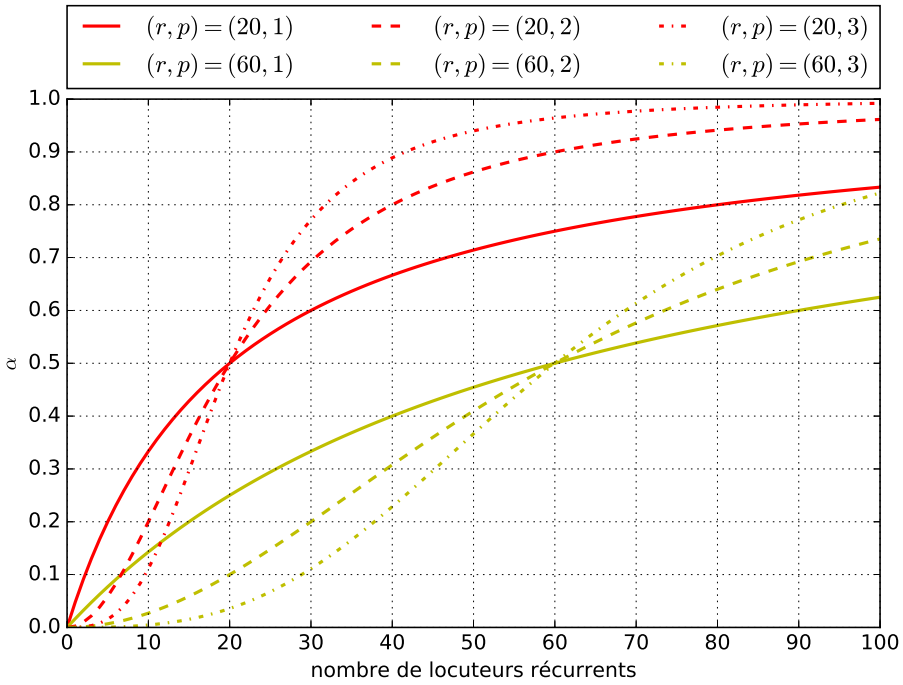

FIGURE 7.1 – Exemples de la formule d'adaptation proposée, pour $r = 20$ ou $r = 60$ et p variant 1 à 3.

7.2.2 DER moyen pondéré : définition

Pour étudier l'influence de la taille des collections sur les performances de l'adaptation, nous répétons les expériences du chapitre précédent, non plus sur les collections complètes mais sur des collections de taille variable. Pour ce faire, pour chacune des deux collections cibles, leurs documents étant triés par ordre chronologique, nous définissons N sous-collections, chaque sous-collection k contenant les

1. C'était la stratégie employée lors de nos travaux préliminaires, qui ont fait l'objet d'une publication [Le Lan et al., 2016a,b]. Cependant, dans ces travaux préliminaires, l'écart inter-domaine entre données d'apprentissage et données cibles était plus faible : les données d'apprentissage contenaient des épisodes des émissions cibles.

k premiers épisodes de la collection, pour $k \in [1, N]$. Pour chaque sous-collection, le regroupement inter-document est évalué dans la configuration *baseline* et après 2 itérations d'adaptation, pour les différentes méthodes de calcul de similarités (HAC/WCCN, HAC/PLDA, CC/TR). Les données utilisées pour l'adaptation sont uniquement celles de la sous-collection. Les expériences sont conduites indépendamment pour chaque sous-collection (c'est-à-dire que les résultats obtenus pour un document dans une sous-collection peuvent changer pour le même document dans une autre sous-collection), et chaque regroupement est effectué sur le paquet d'*i-vectors* obtenus après la segmentation intra-document. Les résultats obtenus pour la $k^{ème}$ sous-collection n'ont aucune influence sur les résultats de la $(k+1)^{ème}$.

Les expériences dépendent toujours des paramètres λ_I, λ_X, les seuils de regroupement, et α, le coefficient d'adaptation. Une recherche exhaustive est réalisée pour $\alpha \in [0, 1[$. Pour évaluer les performances sur l'ensemble des sous-collections, on définit le DER inter-document moyen pondéré (waDER) correspondant au DER moyen der_k de chaque sous-collection k, pondéré par sa durée totale d_k. La formule est explicitée dans l'équation 7.8.

$$waDER = \frac{\sum_{k=1}^{N} d_k der_k}{\sum_{k=1}^{N} d_k} \tag{7.8}$$

7.3 Influence de la taille des collections

7.3.1 Système HAC/PLDA

Les meilleures configurations d'adaptation du système HAC/PLDA sont présentées dans les figures 7.2 et 7.3. Chaque figure est divisée en deux parties. La partie supérieure représente le DER inter-document pour les sous-collections 1 à N, pour les itérations 0 (*baseline*) à 2. L'histogramme de la partie inférieure représente le DER intra-document de chaque épisode.

Les figures 7.2 et 7.3 montrent que les deux itérations améliorent le waDER de 15.2% à 13.3% et de 16.3% à 13.9%, pour $\alpha = 0.5$ et $\alpha = 0.3$, respectivement. Comme vu dans la section 6.4.3, la seconde itération n'apporte pas autant d'amélioration sur BFM que sur LCP. Sur LCP, elle commence à apporter un gain significatif à partir du 26-ième épisode. Auparavant, son effet est nul. Lorsqu'on s'intéresse aux sous-collections de petite taille, on voit que le processus d'adaptation commence à faire effet à partir d'une collection d'environ 8 documents. Pour cette sous-collection, le nombre de classes-locuteurs utilisées pour réaliser l'adaptation est de 4, que ce soit pour BFM ou LCP (cette information n'est pas visible sur les figures). Même avec un nombre de classes-locuteur si faible, l'utilisation d'un α relativement élevé semble ne pas poser problème.

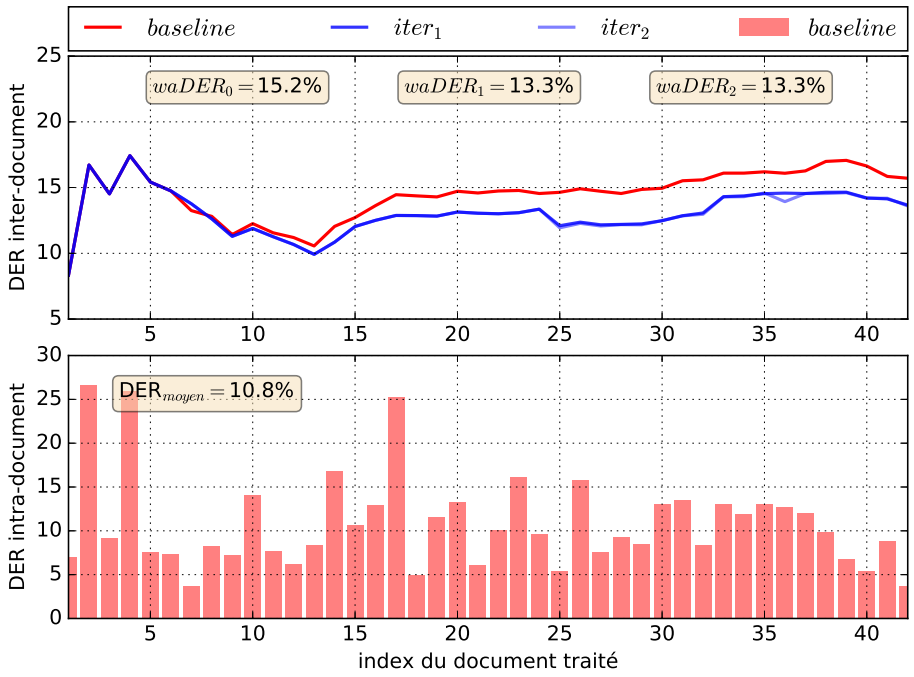

FIGURE 7.2 – DER inter-document des sous-collections de BFM, pour les itérations d'adaptation 0 à 2 du système HAC/PLDA. Les paramètres de l'expérience sont $(\lambda_I = 10, \lambda_X = 10, \alpha = 0.5)$.

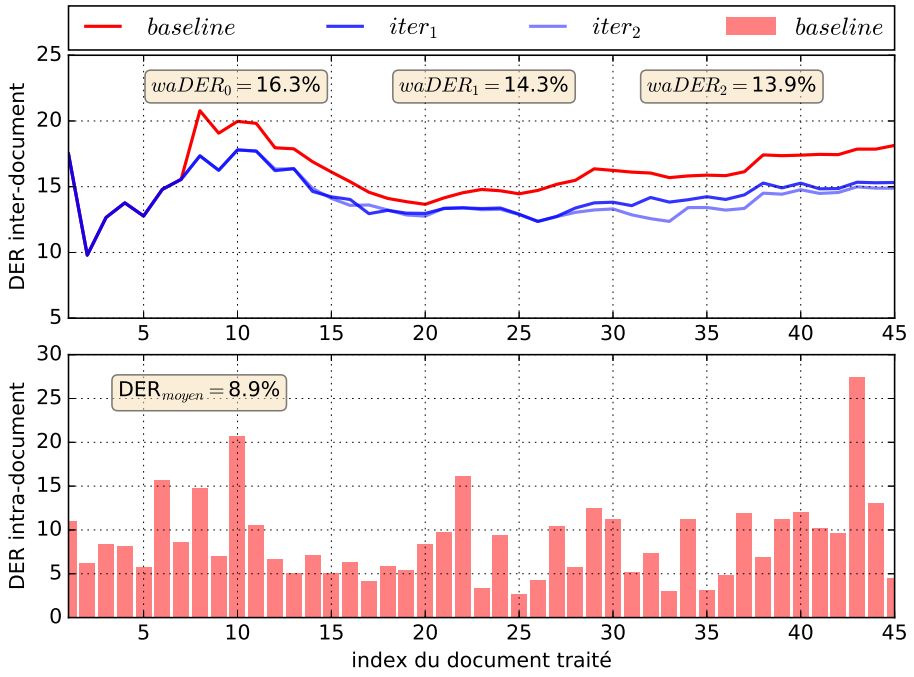

FIGURE 7.3 – DER inter-document des sous-collections de LCP, pour les itérations d'adaptation 0 à 2 du système HAC/PLDA. Les paramètres de l'expérience sont $(\lambda_I = -10, \lambda_X = 10, \alpha = 0.3)$.

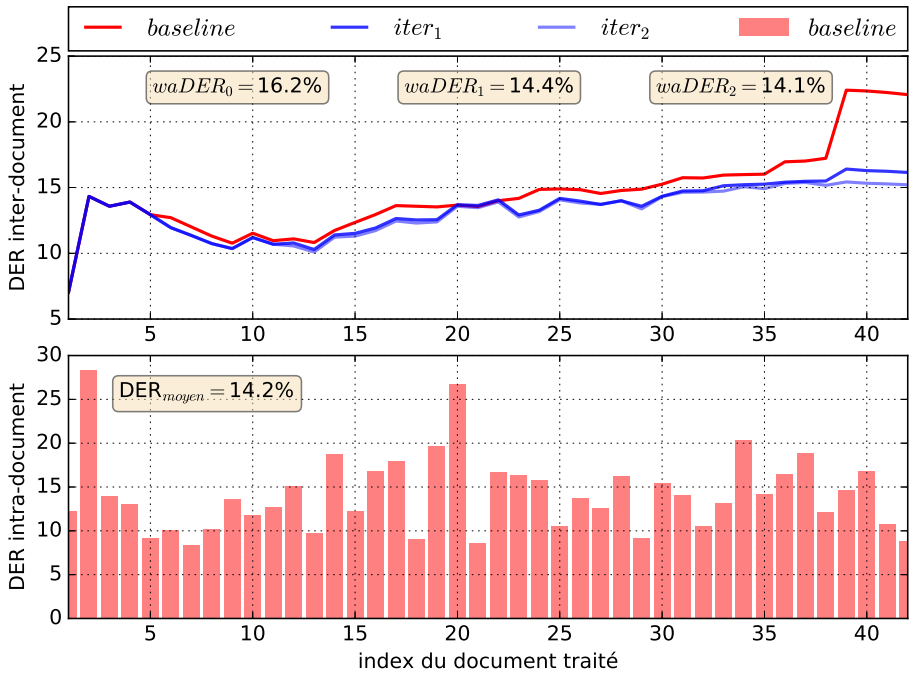

FIGURE 7.4 – DER inter-document des sous-collections de BFM, pour les itérations d'adaptation 0 à 2 du système HAC/WCCN. Les paramètres de l'expérience sont $(\lambda_I = -30, \lambda_X = -40, \alpha = 0.5)$.

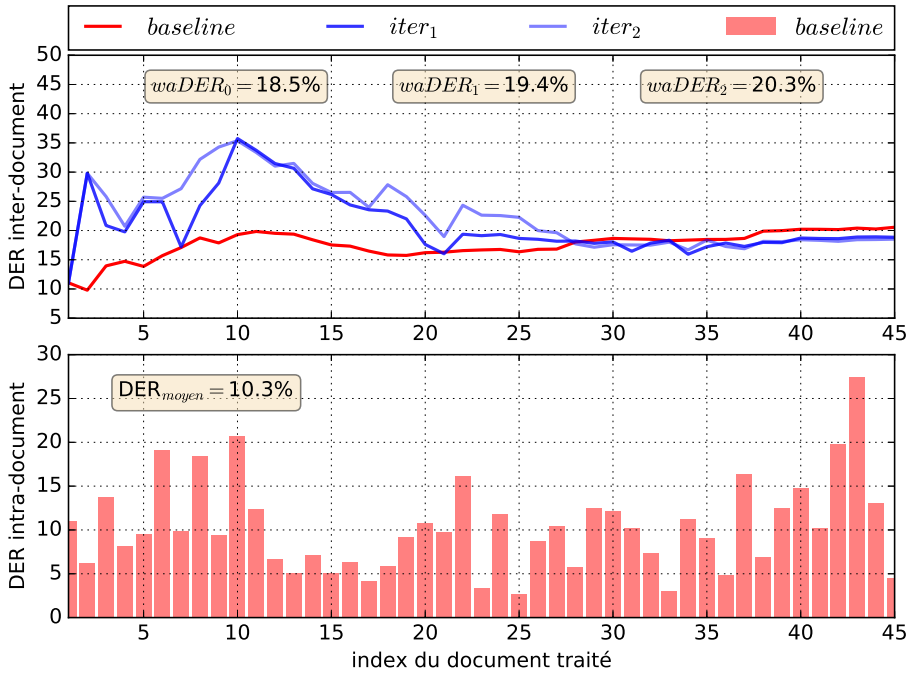

FIGURE 7.5 – DER inter-document des sous-collections de LCP, pour les itérations d'adaptation 0 à 2 du système HAC/WCCN. Les paramètres de l'expérience sont $(\lambda_I = -40, \lambda_X = -30, \alpha = 0.5)$.

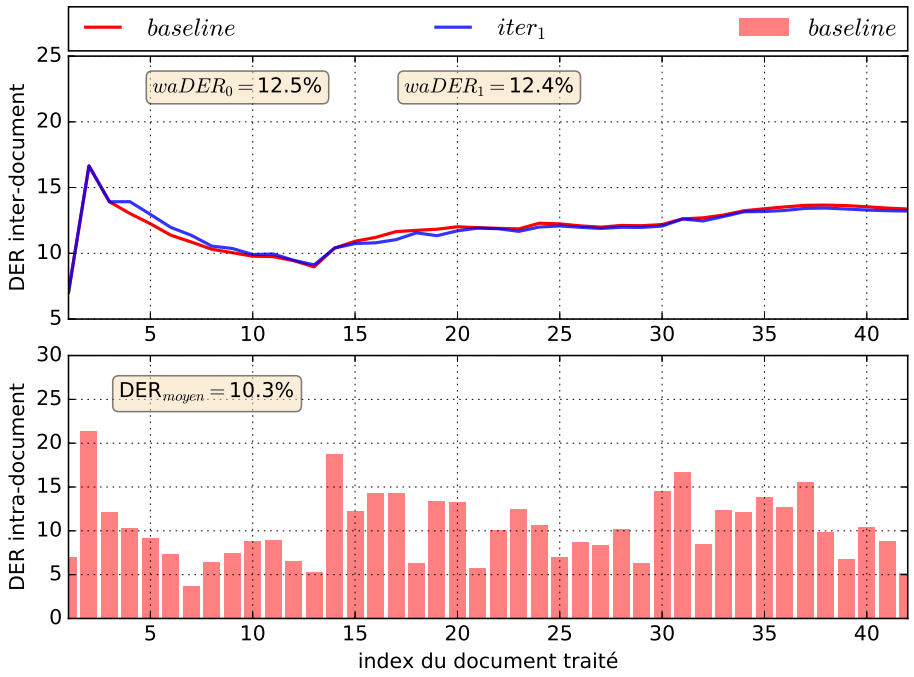

FIGURE 7.6 – DER inter-document des sous-collections de BFM, pour les itérations d'adaptation 0 à 1 du système CC/TR. Les paramètres de l'expérience sont ($\lambda_I = -10, \lambda_X = -30, \alpha = 0.9$).

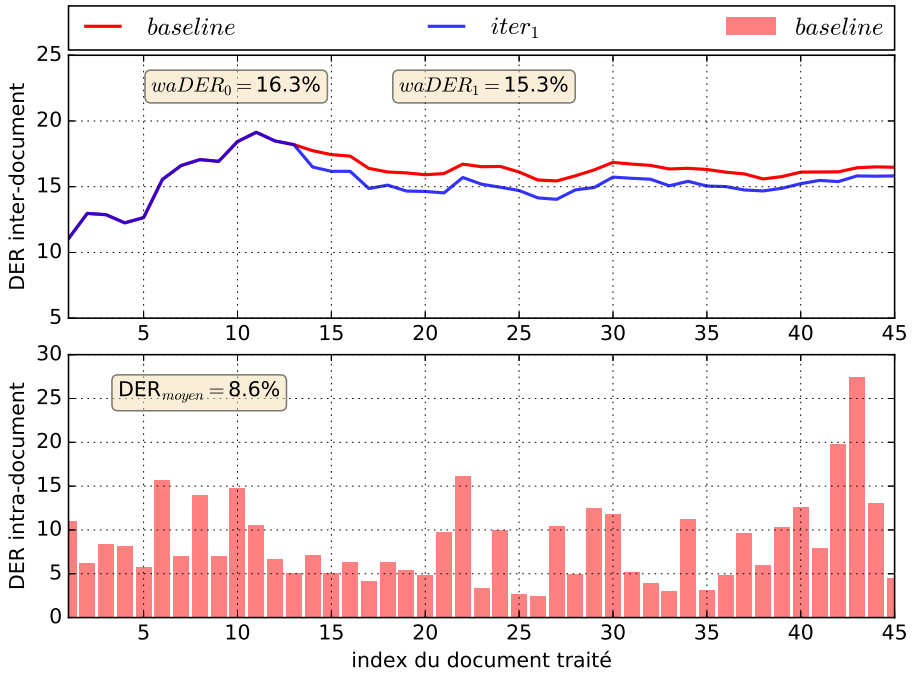

FIGURE 7.7 – DER inter-document des sous-collections de LCP, pour les itérations d'adaptation 0 à 1 du système CC/TR. Les paramètres de l'expérience sont ($\lambda_I = -10, \lambda_X = -30, \alpha = 0.9$).

7.3.2 Système HAC/WCCN

En ce qui concerne les performances d'adaptation du système HAC/WCCN, présentées dans les figures 7.4 et 7.5, l'expérience donne de bons résultats pour BFM, avec un waDER diminuant de 16.2% à 14.0%. En revanche, pour LCP, le comportement du système est inconstant : on remarque que l'adaptation est efficace à partir de la 28-ième sous-collection. En-deçà, l'adaptation peut dégrader le DER de la sous-collection jusqu'à le multiplier par deux, par exemple pour la sous-collection de taille 10. Pour ce système, plus on augmente α, plus la taille de la sous-collection doit être importante pour que l'adaptation réduise le DER *baseline*. Par exemple, lorsqu'$\alpha = 0.9$, l'adaptation ne commence à donner des résultats corrects qu'à partir d'une sous-collection de 42 documents.

7.3.3 Système CC/TR

Pour le système CC/TR, pour des raisons de temps de calcul, nous n'avons réalisé l'expérience que pour une itération d'adaptation (l'adaptation du réseau de neurones dure quelques minutes quand l'adaptation de la PLDA dure quelques secondes). Les résultats sont présentés dans les figures 7.6 et 7.7, pour $\alpha = 0.9$. Cette fois, comme nous l'avions vu lors des expériences d'adaptation du réseau neuronal de compensation de variabilité, le gain apporté par l'adaptation est plus faible. Pour les sous-collections de BFM, le DER varie peu, on observe parfois une légère dégradation, mais le waDER reste quasiment constant, de 12.5% en *baseline* à 12.4% après adaptation. Pour les sous-collections de LCP, l'adaptation n'apporte rien avant la sous-collection de taille 13, mais elle permet, pour les sous-collections de taille supérieure, de gagner environ 1 point de DER. Le waDER varie de 16.3% à 15.3%.

7.4 Optimalité du coefficient d'adaptation

Les expériences de la section précédente montrent que le choix d'un α fixe ne fonctionne pas pour LCP lorsqu'on utilise la similarité cosine/WCCN. En fonction de la taille de la sous-collection, la valeur optimale de α pourrait varier. Pour chaque type de compensation de variabilité, et pour chaque paire de seuil (λ_I, λ_X), on répète l'expérience précédente dix fois, mais à partir de collections dont les documents sont mélangés aléatoirement (et non plus dans l'ordre chronologique). Ceci permet d'avoir plusieurs résultats pour des sous-collections de différentes tailles. Avec α variant de 0 à 0.9, avec un pas de 0.1, il devient possible d'étudier les valeurs optimales de α en fonction de la taille des sous-collections, en moyenne. Les résultats sont présentés dans les figures 7.8 et 7.9, pour les scores PLDA et WCCN, respectivement. Ces figures sont inspirées de la figure 3 de [Garcia-Romero and McCree, 2014]. Pour des raisons de temps de calcul, l'expérience n'a pas été réalisée pour le système

CC/TR (l'adaptation du réseau de neurones dure quelques minutes quand l'adaptation de la PLDA dure quelques secondes). Cependant, nos expériences préliminaires semblent indiquer que pour ce système, la taille de la collection a peu d'influence sur l'optimalité du coefficient d'adaptation, toujours située autour de $\alpha = 0.9$.

Dans chaque figure, le graphique de la partie supérieure concerne BFM, tandis que la partie inférieure concerne les sous-collections de LCP. Sur chaque graphe, différentes zones colorées apparaissent, dont les frontières sont fonction de la taille de la sous-collection et de α. La zone la plus claire correspond à la zone optimale du DER après adaptation, nous la notons $\hat{\alpha}$. Chaque frontière correspond à la valeur de α où le DER inter-document dépasse une certaine limite par rapport au DER optimal. Par exemple, la seconde zone la plus claire correspond à un DER compris entre $\hat{\alpha} + 0.2\%$ et $\hat{\alpha} + 0.5\%$, en valeur absolue. La zone bleue hachurée délimite la zone "interdite", c'est-à-dire la zone où le DER après adaptation dépasse la *baseline*, et où l'adaptation ne fonctionne donc pas. Cette zone n'apparait que dans la partie supérieure de chaque graphe, puisque la limite inférieure correspond à la baseline ($\alpha = 0$ est équivalent à ne pas adapter). Lorsqu'α augmente, l'adaptation commence toujours par améliorer le DER *baseline* jusqu'à un point où il remonte. En conclusion, chaque zone hormis la zone bleue est une zone où l'adaptation améliore ou égale la *baseline*.

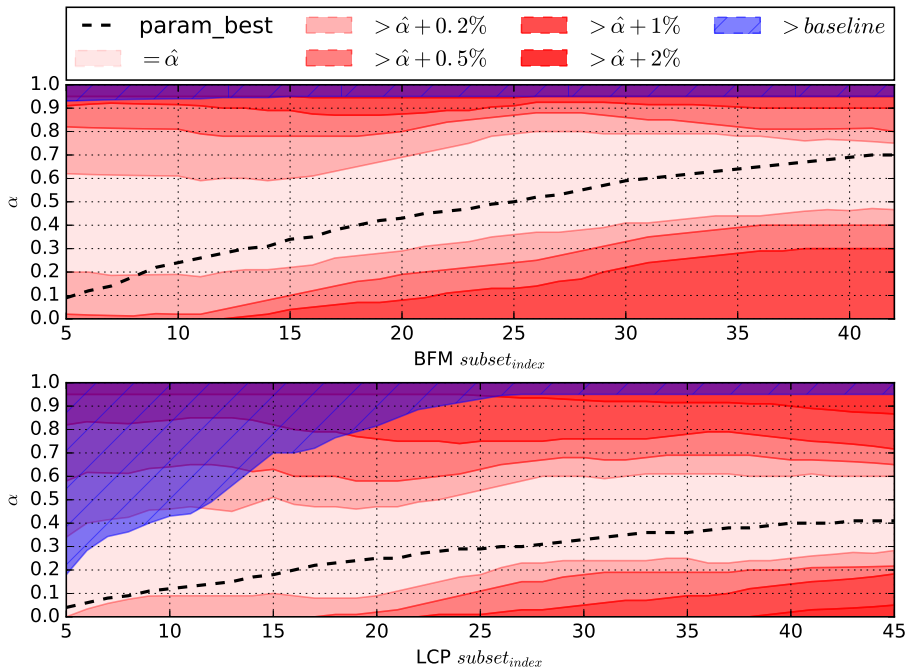

FIGURE 7.8 – Représentation isométrique de l'optimalité du paramètre α, pour l'expérience d'adaptation de PLDA. Les aires de couleur sont fonction de la taille des sous-collections et des valeurs de α. La configuration pour BFM est ($\lambda_I = 10, \lambda_X = 10$), et pour LCP ($\lambda_I = -10, \lambda_X = 10$). Les lignes de niveau ont été lissées sur une fenêtre de taille 5 documents.

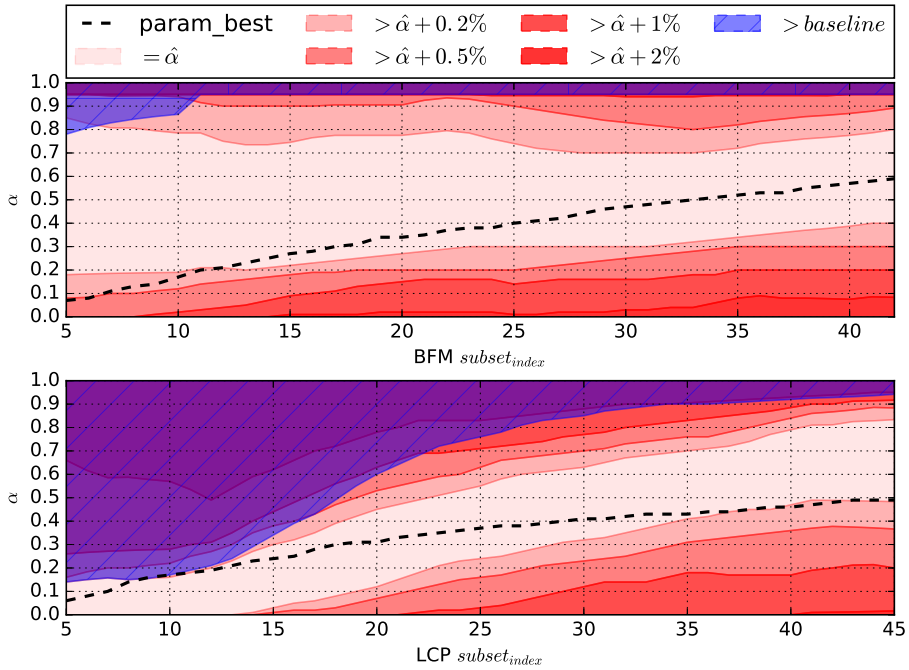

FIGURE 7.9 – Représentation isométrique de l'optimalité du paramètre α, pour l'expérience d'adaptation de la WCCN. Les aires de couleur sont fonction de la taille des sous-collections et des valeurs de α. La configuration pour BFM est ($\lambda_I = -30, \lambda_X = -40$), et pour LCP ($\lambda_I = -40, \lambda_X = -30$). Les lignes de niveau ont été lissées sur une fenêtre de taille 5 documents.

Une première observation commune à chaque graphe est que la zone optimale $\hat{\alpha}$ est centrée sur une valeur de α croissante fonction de la taille de la sous-collection. Cependant, la tendance varie en fonction de la collection ou de la méthode de calcul de similarités. Pour la PLDA, les figures se ressemblent, avec une zone d'optimalité large de 0.2 à 0.4.

En ce qui concerne l'approche WCCN, pour BFM, on voit que la zone d'optimalité est très large au début et tend à se resserrer autour de 0.4 pour la collection complète. Pour LCP, la zone d'optimalité est très fine pour de petites sous-collections, l'optimal étant très proche de 0 (voire égal à 0), et s'étend à 0.3 pour la collection complète. Comme nous l'avons vu dans la section précédente, pour l'expérience LCP-cosine/WCCN avec α petit, l'adaptation dégrade le DER *baseline* si la sous-collection est trop petite : la zone interdite est assez importante dans le graphe inférieur de la figure 7.9.

7.5 Adaptation paramétrique

Dans cette section, on remplace le coefficient d'adaptation fixe α par une version paramétrique, pour tenir compte de la taille de la collection pour laquelle on adapte le système de SRL. La formule paramétrique, présentée à la section 7.2.1, dépend du nombre de classes-locuteurs récurrentes générées lors du regroupement

inter-document précédent. Nous proposons de rejouer les expériences sur les collections aléatoires de la section précédente, mais en utilisant cette fois la formule paramétrique. On effectue une recherche exhaustive pour $r \in \{2, 4, 8, 16, 32, 64, 128\}$ et $p \in \{1, 2, 3\}$. Pour chaque type de compensation de variabilité, on compare les meilleurs waDER, sur 10 collections aléatoires, pour l'approche paramétrique et l'approche fixe. Les résultats sont présentés dans la table 7.1 et les valeurs paramétriques optimales de α sont représentées avec une ligne noire en pointillés sur les figures 7.8 et 7.9.

Lorsqu'on observe les figures 7.8 et 7.9, on voit que les courbes paramétriques suivent les zones d'optimalité, comme espéré. Pour BFM, au vu de la formule paramétrique choisie, les courbes doivent traverser les zones sous-optimales pour les sous-collections les plus petites. C'est pourquoi l'approche paramétrique est légèrement moins efficace que l'approche fixe (+0.1 point de waDER) : en effet, quand on observe les graphes, on peut très bien tracer une ligne horizontale qui resterait dans la zone d'optimalité pour toutes les sous-collections. En ce qui concerne LCP, aucune courbe horizontale ne peut suivre la zone d'optimalité. L'approche paramétrique donne de meilleurs résultats que l'approche fixe, avec un gain de 0.3 point pour le système HAC/WCCN et 0.1 point pour le système HAC/PDLA, comme on peut le lire dans la table 7.1.

système	HAC/WCCN		HAC/PLDA	
collection	LCP	BFM	LCP	BFM
$waDER_{baseline}$	19.6	18.1	17.2	16.3
α	0.5	0.4	0.4	0.5
$waDER_{adapt_{fixe}}$	18.6	**14.2**	15.1	**13.7**
r	32	16	64	16
p	1	1	1	1
$waDER_{adapt_{param}}$	**18.3**	14.3	**15.0**	13.8

TABLE 7.1: Récapitulatif des résultats d'adaptation, en comparant l'adaptation fixe et paramétrique. Les valeurs présentées sont les waDER moyennés sur les 10 collections triées aléatoirement. Les performances *adapt* sont obtenues après deux itérations d'adaptation.

7.6 Conclusion sur la taille des collections

Dans cette partie, nous avons étudié l'influence de la taille des collections sur l'optimalité du paramètre d'adaptation α. Les expériences ont été menées sur les sous-collections de BFM et LCP, sur les systèmes HAC/WCCN, HAC/PLDA et CC/TR, en mesurant le DER moyen pondéré sur l'ensemble des sous-collections. Si pour le système CC/TR, l'optimalité du coefficient d'adaptation ne semble pas dépendre de la taille de la collection, nous avons mis en évidence cette dépendance

pour les deux autres systèmes.

Au vu des zones d'optimalité du coefficient d'adaptation selon la taille des sous-collections considérées, nous avons proposé une méthode paramétrique pour fixer ce coefficient, en le rendant dépendant du nombre de locuteurs récurrents de la sous-collection cible considérée. Les expériences montrent que si adapter sur des sous-collections de trop petite taille dégrade le DER, le choix d'une approche paramétrique peut éviter une telle dégradation.

La taille des collections considérées étant relativement faible (une quarantaine de documents), il faut rester prudent quant à la validité de l'approche paramétrique sur des collections bien plus importantes. Celle-ci reste à démontrer. On peut cependant noter que les résultats de [Garcia-Romero and McCree, 2014], dont les travaux portent sur l'adaptation de domaine en reconnaissance du locuteur, montrent également une dépendance du coefficient d'adaptation à la taille du corpus cible (exprimée en nombre de locuteurs). Si la méthode d'adaptation est différente (modèle 2-covariance, Bayes variationnel) de celle utilisée dans ce manuscrit (PLDA, vraisemblance pondérée), le nombre de locuteurs maximal évalué est de 3790, soit un ordre de grandeur de plus que le nombre de locuteurs des collections cibles sur lesquelles nous réalisons nos expériences.

Chapitre 8

Adaptation incrémentale

Résumé

Ce dernier chapitre aborde la question de la SRL incrémentale de collection, qui consiste à traiter une collection dans l'ordre chronologique de diffusion, sans remettre en cause les résultats du traitement de documents passés. Nous y proposons une modification des méthodes de regroupement et d'adaptation au domaine, qui ont été pensées jusqu'ici dans une architecture de regroupement inter-document global. Les expériences sont notamment consacrées à l'étude des différences de performances entre systèmes à regroupement global et incrémental, pour les systèmes de SRL HAC/WCCN, HAC/PLDA et CC/TR. Elle permettent de mettre en évidence que le bénéfice des contributions principales de la thèse (adaptation au domaine, compensation neuronale de variabilité) reste valable quand on passe d'une architecture globale à incrémentale. Les résultats montrent que le système CC/TR est le plus performant pour la SRL incrémentale de collection, avec une dégradation des performances par rapport à la SRL globale de 1.5% de DER dans le pire des cas (en absolu). On constate également que l'adaptation au domaine permet d'améliorer les performances du système HAC/PLDA au point de talonner celles du réseau de neurones.

8.1 Introduction

Dans les chapitres précédents, nous avons travaillé avec un système de SRL basé sur une architecture de regroupement global. Ce type d'architecture est adéquat pour traiter des collections de taille fixée. Chaque document de la collection est d'abord traité séparément (SRL intra-document), puis quand tous les documents ont été segmentés et regroupés, on applique le regroupement inter-document global. Dans ce cadre, nous avons proposé une nouvelle méthode de compensation de variabilité intra-locuteur/inter-document, dite TR, pour le calcul des similarités entre *i-vectors*.

Nous avons également proposé une stratégie d'adaptation itérative, qui consiste à exploiter les données de la collection pour mettre à jour les modèles de compensation de variabilité et réduire les taux d'erreur de SRL. La stratégie s'est révélée efficace sur deux collections d'une quarantaine d'épisodes, dans le cadre d'un regroupement global.

Par ailleurs, dans certains contextes applicatifs, le système n'ingère pas une collection complète à un instant t, mais doit composer avec la temporalité des diffusions. Par exemple, on peut vouloir indexer la collection des épisodes d'un hebdomadaire au fil de l'eau. Il s'agit alors d'indexer chaque nouvelle diffusion, compte tenu des diffusions précédentes, sans modifier l'indexation des mêmes diffusions précédentes. C'est la problématique du regroupement incrémental, décrit à la section 3.4.3. A chaque nouveau document traité, ses classes-locuteurs sont regroupées avec les classes-locuteurs issues des regroupements passés. Ces classes-locuteurs passées ne peuvent pas fusionner, elle ne peuvent qu'agréger de nouveaux éléments.

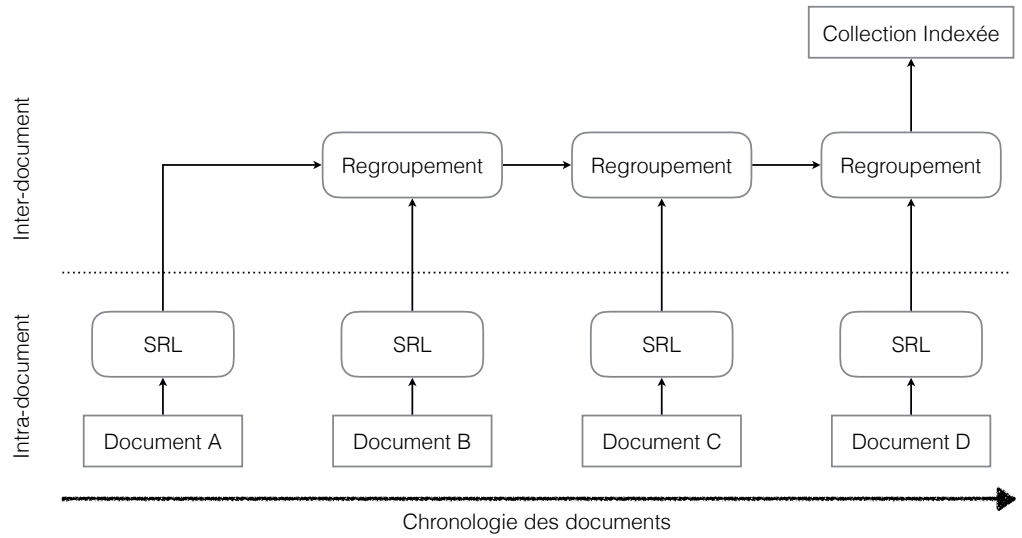

FIGURE 8.1 – Principe du regroupement incrémental pour la SRL de collection.

Cette contrainte applicative amène un certain nombre de questions : comment évoluent les performances par rapport au regroupement global, peut-on adapter tout en traitant les collections de façon incrémentale ? Nous commencerons par discuter des stratégies de regroupement et d'adaptation dans le cadre du traitement incrémental des collections, avant de les évaluer, pour les systèmes de SRL HAC/WCCN, HAC/PLDA et CC/TR. A la fin de ce chapitre, nous serons en mesure de recommander une stratégie d'adaptation pour la SRL incrémentale des collections.

8.2 Stratégies de regroupement incrémental

[Dupuy, 2015] a étudié la question du regroupement incrémental pour la SRL de

collection, en autorisant la fusion de classes-locuteurs passées. La stratégie consiste, lors du traitement d'un nouveau document, à effectuer un regroupement global sur l'ensemble des classes-locuteurs disponibles : les classes-locuteurs issues des regroupements inter-document entre les documents passés et celles issues du regroupement intra-document du document courant. Mécaniquement, rien n'empêche donc à des classes-locuteurs passées de fusionner lors du traitement d'un nouveau document.

Par ailleurs, après avoir ajouté un nouveau document, chaque classe-locuteur est réduite à un seul *i-vector*. Cette approche présente l'avantage de diminuer la complexité de l'ajout de nouveaux documents. Deux variantes sont présentées :

- Après l'ajout d'un document et une fois les regroupements inter-documents effectués, un *i-vector* est recalculé sur l'ensemble du signal de parole de la classe-locuteur. Ce procédé est assez lourd en temps de calcul et nécessite d'accéder aux statistiques des documents passés.
- La seconde variante, appelée "recyclage", consiste à choisir parmi les *i-vectors* constituant une classe-locuteur le *i-vector* le plus central.

Quelle que soit la stratégie employée, les performances obtenues avec le regroupement incrémental sont compétitives avec le système de SRL de collection à regroupement global. La seconde variante, beaucoup plus légère en temps de calcul, ne dégrade pas les performances de plus d'un point de DER inter-document par rapport à la première variante.

Dans nos expériences précédentes, pour le système de SRL à regroupement global, présenté à la section 6.3, nous avons réalisé les expériences avec un regroupement HAC à saut maximum ou un regroupement CC (équivalent à un regroupement hiérarchique à saut minimum). Pour faire le lien avec nos expériences précédentes, notamment sur la question des méthodes de regroupement et d'adaptation au domaine qui nécessitent d'utiliser tous les *i-vectors* constituant une classe-locuteur, nous n'utilisons pas la méthode de réduction des classes-locuteurs proposée par [Dupuy, 2015].

Nous proposons d'utiliser la stratégie de regroupement incrémental de [Van Leeuwen, 2010] : lors du traitement d'un nouveau document, cette approche consiste juste à regrouper chaque classe-locuteur du document courant avec la classe-locuteur passée la plus proche (ou à aucune, selon la valeur du seuil λ_X). Ainsi, des classes-locuteurs issues de regroupements inter-document passés ne peuvent jamais fusionner entre elles.

8.2.1 Regroupement HAC incrémental

Selon la méthode de regroupement HAC à saut maximum, lorsque deux classes-locuteurs fusionnent en une seule ($\boldsymbol{U}$), la mise à jour des distances avec les autres classes $\boldsymbol{V}$ se fait selon l'équation suivante :

$$dist(\boldsymbol{U}, \boldsymbol{V}) = max_{i,j}(dist(\boldsymbol{u_i}, \boldsymbol{v_j})) \tag{8.1}$$

Selon l'algorithme de regroupement incrémental, chaque classe-locuteur issue d'un nouveau document est regroupée avec la classe-locuteur passée la plus proche au sens du regroupement à saut maximum : c'est-à-dire selon la distance entre les deux *i-vectors* les plus distants. Conceptuellement, il s'agit d'un regroupement hiérarchique à saut maximum pour lequel on ajoute des feuilles au cours du temps (voir figure 3.1).

8.2.2 Regroupement CC incrémental

Pour le regroupement en composantes connexes, équivalent à un regroupement HAC à saut minimum, la mise à jour des distances se fait selon l'équation :

$$dist(\boldsymbol{U}, \boldsymbol{V}) = min_{i,j}(dist(\boldsymbol{u_i}, \boldsymbol{v_j})) \tag{8.2}$$

Selon l'algorithme de regroupement incrémental, chaque classe-locuteur issue d'un nouveau document est regroupée avec la classe-locuteur passée la plus proche au sens du regroupement à saut minimum : c'est-à-dire selon la distance entre les deux *i-vectors* les plus proches.

L'approche incrémentale utilisée est moins naturelle pour ce regroupement. A la différence du regroupement à saut maximum qui se veut conservatif, le saut minimum permet que des classes-locuteurs déjà formées fusionnent : par exemple quand une classe-locuteur issue d'un nouveau document se positionne à mi-chemin de deux classes-locuteurs préexistantes.

8.3 Stratégie d'adaptation incrémentale

Lors des expériences sur le regroupement global, nous avons montré qu'adapter les modèles de variabilité intra-locuteur/inter-document permettait de réduire le DER. Dans une optique de regroupement incrémental, pour effectuer les regroupements entre les $N-1$ premiers documents et le suivant, nous proposons d'utiliser des scores de similarité calculés grâce à des modèles adaptés sur les classes-locuteurs issues des $N-1$ premiers documents. Les modèles sont mis à jour après avoir relié chaque document aux classes-locuteurs pré-existantes.

8.4 Evaluation des stratégies proposées

Pour les expériences, nous proposons de comparer les systèmes HAC/WCCN, HAC/PLDA et CC/TR, avec les stratégies de regroupement/adaptation incrémen-

taux proposés. Chaque système dépend d'une configuration $(\lambda_I, \lambda_X, \alpha)$. Les systèmes sont évalués sur les collections BFM et LCP, triées dans l'ordre chronologique de diffusion, dans une configuration sans adaptation (*baseline*) et dans une configuration avec adaptation (*adapt*). Le DER est évalué après chaque regroupement. Les configurations sont les suivantes :

- Pour le système HAC/WCCN, $(\lambda_I = -50, \lambda_X = -40)$ en *baseline*, avec $\alpha = 0.2$ pour l'adaptation.
- Pour le système HAC/PLDA, $(\lambda_I = -10, \lambda_X = 10)$ en *baseline*, avec $\alpha = 0.4$ pour l'adaptation.
- Pour le système CC/TR, $(\lambda_I = -10, \lambda_X = -30)$ en *baseline*, avec $\alpha = 0.9$ pour l'adaptation.

8.4.1 Incrémental vs. Global

La première expérience que nous proposons consiste à comparer les performances entre les architectures de regroupement global et incrémental pour les systèmes *baseline* (sans adaptation). Dans les figures 8.2 et 8.3, nous présentons les résultats des deux architectures sur les trois méthodes de regroupement/compensation de variabilité. Sur les deux figures et pour les 3 systèmes testés, le constat est le même : le regroupement incrémental donne de plus mauvaises performances que le regroupement global. C'est particulièrement flagrant avec la similarité PLDA ou cosine avec WCCN, où le DER sur la collection complète peut augmenter de plus de 10 points (HAC/WCCN sur BFM).

Quels que soient le système et l'architecture de regroupement, l'allure générale des courbes montre une augmentation du DER à mesure que la collection est grande, avec d'importantes variations pour les premiers documents. Ces variations s'expliquent par le fait qu'un nouveau document a plus d'impact sur le DER inter-document lorsque la collection est petite. C'est pourquoi les courbes se lissent à mesure qu'on approche du dernier épisode. On note sur la collection LCP un phénomène de baisse autour du $20^{ème}$ document, après un pic de DER autour du $10^{ème}$, pour les trois systèmes. Sur les deux collections, le système le plus performant est le système CC/TR, que le regroupement soit global ou incrémental. D'ailleurs, avec ce système, on a l'impression que le DER de la collection LCP a tendance à stagner plus qu'à augmenter.

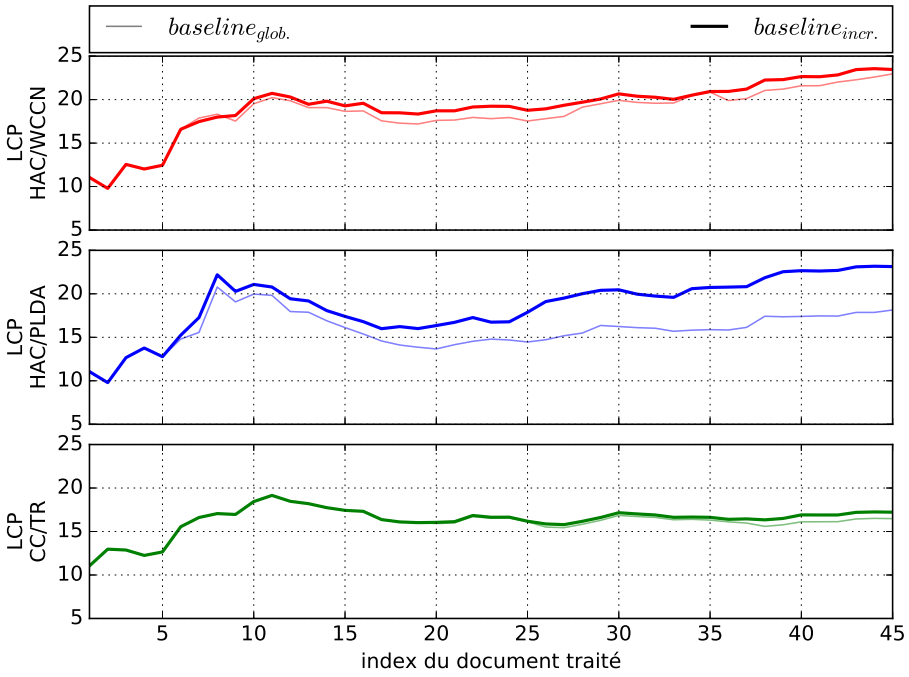

FIGURE 8.2 – Evolution du DER inter-document sur la collection LCP, en fonction de l'index du document traité par le système *baseline* global ou incrémental.

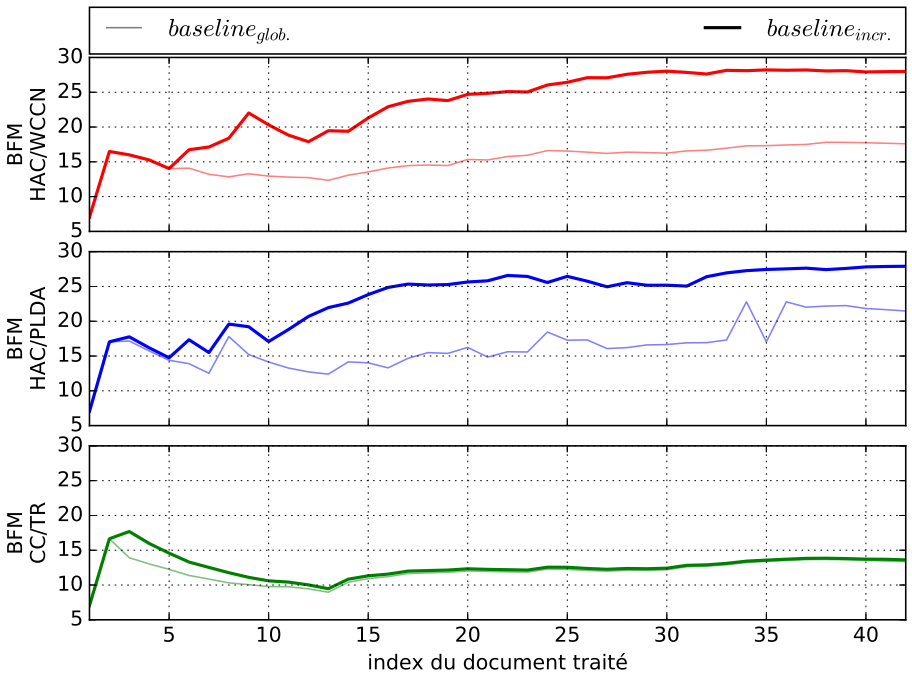

FIGURE 8.3 – Evolution du DER inter-document sur la collection BFM, en fonction de l'index du document traité par le système *baseline* global ou incrémental.

Concernant la collection BFM, pour laquelle l'évolution comparative du DER entre architecture à regroupement global et incrémental est représentée à la figure 8.3, on observe pour les systèmes HAC/WCCN et HAC/PLDA des écarts conséquents. On remarque notamment que l'écart se creuse assez rapidement entre le

$5^{ème}$ et le $15^{ème}$ document de la collection.

Par curiosité, nous avons également étudié ce qui se passe lorsqu'on relâche la contrainte de non fusion de classes passées en regroupement incrémental. Il s'agissait de la stratégie incrémentale adoptée par [Dupuy, 2015] : laisser possible la fusion des classes-locuteurs passées. Dans ce cas, le regroupement incrémental consiste en une succession de regroupements globaux, à l'ajout de chaque nouveau document. Nous avons renseigné les résultats des expériences dans la table 8.1. Les valeurs indiquées correspondent au DER sur chaque collection complète, c'est-à-dire le point le plus à droite des courbes présentées précédemment. L'expérience contrastive, avec fusion autorisée, est présentée aux lignes $BFM_{incr.+f}$ et $LCP_{incr.+f}$.

Si nous n'avons pas représenté cette expérience sur les courbes précédentes, c'est parce que les courbes se confondent avec l'existant. Sur les systèmes à regroupement HAC, autoriser les fusions de classes-locuteurs passées n'a aucune influence sur le DER. Les courbes sont donc confondues avec les courbes $baseline_{incr.}$ des figures 8.2 et 8.3. En effet, le regroupement HAC, conservatif, utilise en configuration *baseline* les mêmes modèles de calcul de similarités pour chaque document. Il est donc impossible que la distance entre deux classes-locuteurs diminue à mesure que la collection s'enrichit de nouveaux documents. La propriété de saut maximum fait que la distance entre deux classes-locuteurs ne peut qu'augmenter : si elles n'ont pas été regroupées par le passé, elle ne le seront jamais. Sur le système à regroupement CC, a contrario, les performances incrémentales avec fusion autorisée sont identiques à celles du regroupement global ($baseline_{glob.}$). En effet, les paramètres de regroupement (λ_I, λ_X) étant identiques dans les deux cas, et les modèles de calcul de similarités étant fixes, le regroupement incrémental est strictement équivalent au regroupement global. Le regroupement s'effectue par chaînage, sur les mêmes *i-vectors* dans les deux architectures, seul l'ordre change.

système config.	HAC/WCCN	HAC/PLDA *baseline*	CC/TR
$BFM_{glob.}$	17.6	21.5	**13.4**
$BFM_{incr.}$	28.0	27.9	**13.6**
$BFM_{incr.+f}$			**13.4**
$LCP_{glob.}$	23.0	18.1	**16.5**
$LCP_{incr.}$	23.5	23.1	**17.2**
$LCP_{incr.+f}$			**16.5**

TABLE 8.1: Comparaison des performances des architectures globale et incrémentale de regroupement inter-document sur chaque collection, sans adaptation.

Lien avec les travaux antérieurs Un système de SRL incrémental a également été évalué sur des collections comparables par [Dupuy, 2015]. On peut noter qu'avec un système à regroupement incrémental CC+ILP/PLDA, avec fusions tardives au-

torisées, l'auteur obtient un DER inter-document de 14.3% sur BFM et 20.9% sur LCP. Encore une fois, l'écart avec les performances présentées dans la table 8.1 (respectivement 27.9% et 23.1% avec le modèle PLDA) s'explique probablement par la présence d'épisodes de *BFM Story* dans les données d'apprentissage. La différence de domaine acoustique entre données d'apprentissage et données cibles doit avoir encore plus d'impact sur les performances dans la configuration de regroupement incrémental, puisque les regroupements sont contraints et qu'un mauvais regroupement ne peut pas être corrigé. On peut tout de même remarquer que le système CC/TR montre de bien meilleurs résultats (respectivement 13.4% et 16.5%), en-deçà de ceux obtenus par [Dupuy, 2015].

8.4.2 Adaptation incrémentale

A présent, nous nous concentrons sur l'architecture incrémentale, et comparons systèmes *baseline* et *adapt*. Le regroupement de classes passées n'est pas autorisé. Les expériences sont illustrées par la figure 8.4, où nous avons représenté les performances comparatives des 6 systèmes (*baseline* ou *adapt*, avec HAC/WCCN, HAC/PLDA ou CC/TR), pour chaque collection.

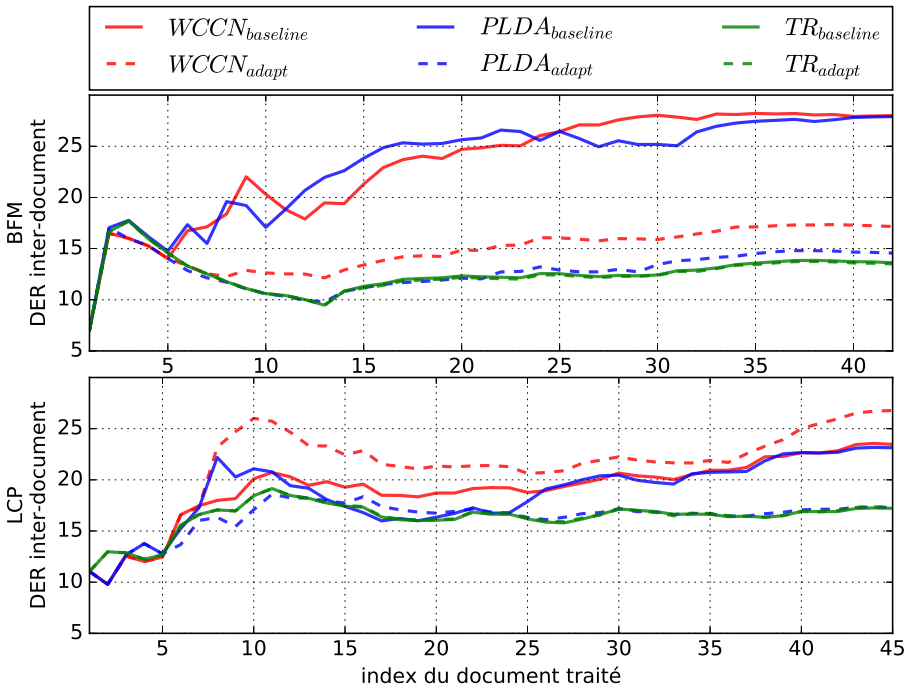

FIGURE 8.4 – Evolution du DER inter-document sur les deux collections cibles, en fonction de l'index du document traité par le système incrémental (*baseline* ou *adapt*, avec la similarité cosine/WCCN, PLDA ou TR).

La comparaison de l'approche *baseline* avec l'approche adaptée (*adapt*) révèle des comportements différents selon les systèmes. Premièrement, on note que l'adaptation du réseau TR donne des performances quasi identiques à la *baseline*, qui étaient déjà

les meilleures. Ceci laisse penser que la modélisation neuronale pour la compensation de variabilité est capable de suffisamment généraliser sur les données d'apprentissage. Adapter le réseau sur les données cibles apporte peu.

Concernant la PLDA, on remarque que l'adaptation apporte une amélioration significative sur les performances, qui tendent à se rapprocher des performances du système CC/TR. Pour la collection BFM, le DER au dernier document est à 14.6%, 1 point plus élevé que pour l'approche TR. Sur LCP, le DER atteint les 17.3%. En revanche, l'adaptation de la WCCN donne des résultats plus mitigés. Si sur BFM elle améliore significativement les résultats pour arriver à un DER final de 17.2% (contre 28.0% en *baseline*), sur LCP elle dégrade la *baseline* à partir du $7^{ème}$ document. Il est intéressant de remarquer que même si des erreurs de regroupement sont commises tôt, le DER peut tout de même diminuer à mesure qu'on regroupe. Une erreur de regroupement isolée a en effet moins d'impact sur le calcul du DER à mesure qu'on ajoute de nouveaux documents.

En résumé, l'approche proposée, qui consiste à regrouper de façon incrémentale sans modifier le passé, en utilisant des modèles de compensation de variabilité adaptés sur les documents déjà traités, semble efficace pour le système HAC/PLDA. Pour le système HAC/WCCN, l'effet de l'adaptation sur la collection LCP est négatif, même si à d'autres seuils de regroupement, sous-optimaux, nous avons constaté qu'il pouvait être efficace. Enfin, le système CC/TR *baseline* semble avoir atteint un palier puisque l'adaptation n'apporte aucun gain, même si on est encore loin des meilleures performances inter-document atteignables : 6.2% sur LCP et 7.8% sur BFM (voir section 4.2.1.2), contre 16.5% et 13.4% avec le système CC/TR.

8.4.3 Relâche de la contrainte de regroupement

Dans la section précédente, nous interdisions la fusion de classes-locuteurs passées. Ici, nous relâchons cette contrainte : on ne peut pas remettre en cause leur composition, mais on peut les fusionner. Pour des raisons de lisibilité, nous illustrons les résultats sur chaque collection par une figure différente (figures 8.5 pour BFM et 8.6 pour LCP). Chaque figure comprend trois graphes, un par système (regroupement/similarité), où nous reportons également les courbes du système à regroupement global $baseline_{glob.}$ et adapté ($adapt_{glob}$). L'échelle des ordonnées varie selon les systèmes pour maximiser la lisibilité.

A la section 8.4.1, nous avons discuté de l'influence de la contrainte de fusion sur les systèmes $baseline_{incr.}$. Pour les systèmes $baseline_{incr.}$ à regroupement HAC (WCCN et PLDA), autoriser les fusions de classes-locuteurs passées n'a aucune influence sur le DER. En revanche, lorsque le système est adapté ($adapt_{incr.+f}$), la distance entre deux classes $dist(\boldsymbol{U}, \boldsymbol{V}) = max_{i,j}(dist(\boldsymbol{u_i}, \boldsymbol{v_j}))$ peut varier au cours du traitement de la collection. Des regroupements entre classes-locuteurs passées ne sont donc pas exclus.

8.4.3.1 Collection BFM

Sur les expériences de la figure 8.5, on observe peu de différences entre les différentes versions des systèmes adaptés ($adapt_{glob.}$, $adapt_{incr.}$ et $adapt_{incr.+f}$). La seule exception concerne le système HAC/WCCN, où, lorsqu'on autorise la fusion, les performances se dégradent à partir de l'épisode 27. Sans fusion, le système incrémental adapté est légèrement meilleur que le système $baseline_{glob.}$, et au niveau du système $adapt_{glob.}$.

Avec la modélisation PLDA, on constate peu de différences entre les trois systèmes adaptés, mais on peut voir qu'ils se situent autour de 15% de DER sur la collection complète, soit 12 points sous le système $baseline_{glob.}$.

Enfin, avec la compensation neuronale de variabilité, les 5 configurations se tiennent dans un intervalle de 0.5%. On constate tout de même que le regroupement global $adapt_{glob.}$ donne les meilleures performances, suivi de près par le regroupement incrémental avec fusion $adapt_{incr.+f}$. Ces deux approches battent légèrement le système $baseline_{glob.}$.

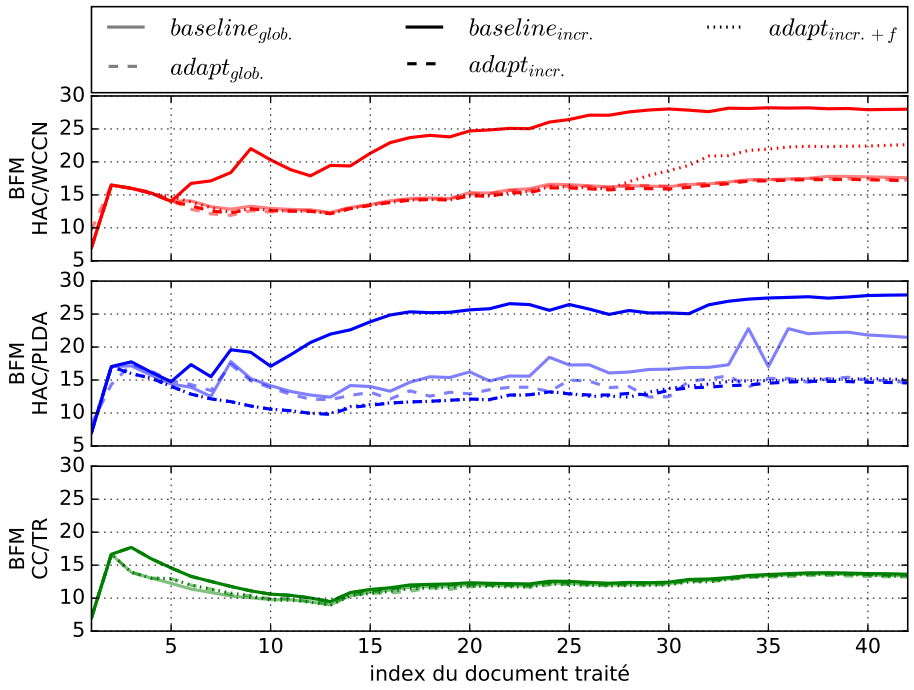

FIGURE 8.5 – Evolution comparative du DER inter-document sur la collection BFM, en fonction de l'index du document traité par le système incrémental (avec ou sans adaptation, avec ou sans fusion de classes-locuteurs passées).

8.4.3.2 Collection LCP

Sur le graphe supérieur de la figure 8.6, on note que relâcher la contrainte d'interdiction de fusion a un effet négatif sur le système HAC/WCCN. Pour ce système, l'approche de regroupement incrémental avec adaptation ($adapt_{incr.}$) dégradait déjà

la $baseline_{incr.}$ en interdisant les regroupements entre classes passées. En les autorisant, le DER final est augmenté de 5 points supplémentaires. Contrairement aux observations sur les systèmes PLDA et TR sur les deux collection, l'adaptation incrémentale n'est pas compétitive avec l'adaptation globale.

Avec la PLDA, autoriser les fusions a un effet bénéfique. Ainsi, le gain de DER final, après avoir traité toute la collection, est de 2 points (15.1% contre 17.3% sans fusion). C'est le système incrémental donnant le DER final le plus faible parmi ceux présentés, et il est compétitif avec le système à regroupement et adaptation globaux. Concernant le système CC/TR, on constate que l'adaptation avec fusion apporte un gain de performances par rapport à la $baseline_{incr.}$, ce qui n'était pas le cas auparavant. Ceci indique que pour ce système, l'adaptation a un effet bénéfique sur le phénomène de fusions de classes-locuteurs passées, car elle permet de faire baisser le DER. Comme pour BFM, le système $adapt_{incr.+f}$ se place entre les systèmes à regroupement global $baseline_{glob.}$ et $adapt_{glob.}$.

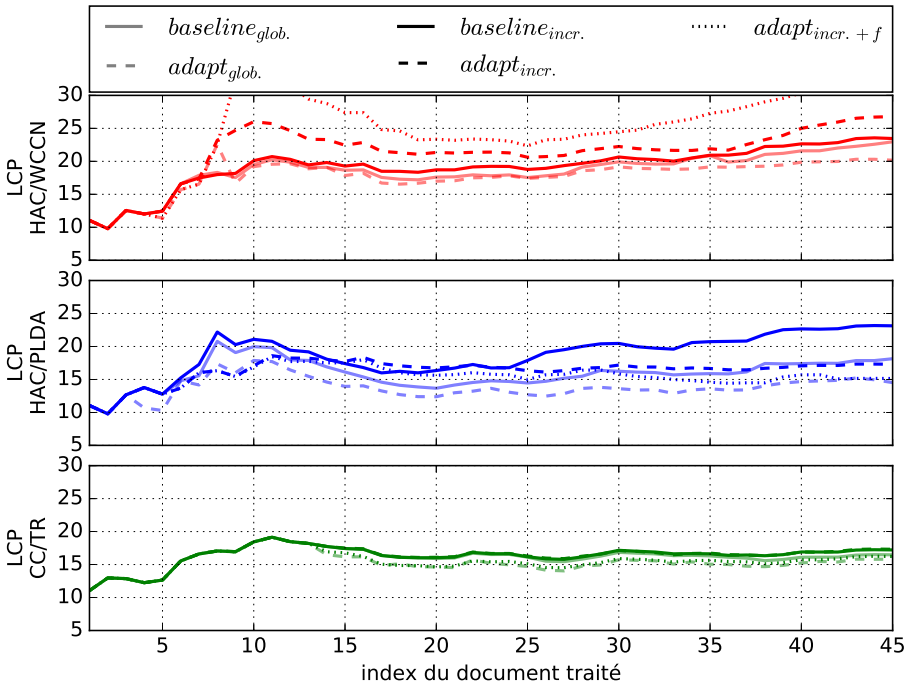

FIGURE 8.6 – Evolution comparative du DER inter-document sur la collection LCP, en fonction de l'index du document traité par le système incrémental (avec ou sans adaptation, avec ou sans fusion de classes-locuteurs passées).

Dans la table 8.2, nous présentons les résultats récapitulatifs des différents systèmes présentés dans les figures. Les DER finaux sont présentés pour les différents systèmes incrémentaux, que l'on autorise le regroupement de classes passées ($BFM_{incr.+f}$ et $LCP_{incr.+f}$) ou non ($BFM_{incr.}$ et $LCP_{incr.}$). Le DER du système global appliqué à la collection complète est également présenté, dans les mêmes configurations $(\lambda_I, \lambda_X, \alpha)$ que les systèmes incrémentaux. Notons également que pour chaque système, la configuration utilisée est commune aux deux collections.

La table illustre bien le fait qu'en autorisant les fusions de classes passées, les performances des systèmes incrémentaux adaptés sont compétitives avec les systèmes à architecture de regroupement global. La seule exception concerne le système HAC/-WCCN appliqué à la collection LCP, où l'adaptation dégrade les performances sur les systèmes incrémentaux. Pourtant, sur le système global, on parvenait à réduire le DER de 3 points par rapport à la *baseline*. Pour les autres systèmes, on constate dans le pire des cas un écart de 0.4 points, pour le système CC/TR appliqué à la collection LCP (15.8% en global contre 16.2% en incrémental). De manière générale, le système le plus performant après adaptation est le système CC/TR, suivi de près par HAC/PLDA. Il est utile de noter que l'adaptation apporte beaucoup plus à la similarité PLDA qu'à la similarité TR, qui est en *baseline* déjà la plus performante.

système	HAC/WCCN		HAC/PLDA		CC/TR	
config.	*baseline*	*adapt*	*baseline*	*adapt*	*baseline*	*adapt*
$BFM_{glob.}$	17.6	17.3	21.5	14.8	13.4	**13.3**
$BFM_{incr.}$	28.0	17.2	27.9	14.6	13.6	**13.5**
$BFM_{incr.+f}$	28.0	17.2	27.9	14.6	13.4	**13.3**
$LCP_{glob.}$	23.0	20.2	18.1	**14.9**	16.5	15.8
$LCP_{incr.}$	23.5	26.8	23.1	17.3	**17.2**	17.3
$LCP_{incr.+f}$	23.5	31.9	23.1	**15.1**	16.5	16.2

TABLE 8.2: Comparatif des DER finaux des systèmes incrémentaux avec les DER des systèmes globaux sur chaque collection complète, dans la même configuration.

En conclusion, la contrainte qui consiste à interdire la fusion de classes-locuteurs issues du regroupement entre documents passés a des effets variables selon les systèmes considérés. Il est indéniable que pour le système CC/TR, imposer une telle contrainte dégrade les performances, car c'est en contradiction avec la philosophie de la méthode de regroupement choisie. Par conception, on peut très bien imaginer que le segment d'un nouveau document joue le rôle de pont entre deux classes préexistantes. Ces deux classes, trop éloignées pour avoir fusionné par le passé, peuvent fusionner si un singleton équidistant des deux se présente. Ces résultats montrent en tout cas que si, pour des raisons applicatives, on veut figer les résultats des traitements passés, on peut perdre jusqu'à 1.1 point de DER par rapport au système incrémental plus permissif.

8.4.4 Influence de l'initialisation

Dans les trois sections précédentes, nous avons effectué le regroupement incrémental à compter du premier document de la collection. Cependant, avoir la connaissance d'un contexte plus large à l'initialisation pourrait permettre d'améliorer les performances, surtout quand on sait qu'il peut y avoir un écart de performances significatif entre un système à regroupement global et incrémental. Nous proposons

donc de répéter les expériences précédentes en effectuant d'abord un regroupement global sur les $(10 \times k)$ premiers documents, avant de traiter de façon incrémentale tous les épisodes suivants. Nous comparons les performances avec ou sans adaptation, sur les deux collections distinctes. Pour la configuration *baseline* (respectivement *adapt*), les résultats sont présentés à la figure 8.7 (resp. 8.8). Avec adaptation, le fonctionnement est le suivant :

- Initialisation par un regroupement global sur une sous-collection de taille donnée.
- Adaptation des modèles sur cette même sous-collection.
- Mise à jour du regroupement (toujours global).
- Pour chaque nouveau document, regroupement incrémental avec les modèles adaptés, puis mise à jour des modèles.

Nous nous plaçons dans le cas incrémental strict, qui n'autorise pas les fusions.

8.4.4.1 Configuration *baseline*

La figure 8.7 comprend deux graphes : la partie supérieure présente les expériences *baseline* de la collection BFM et l'inférieure celles de la collection LCP. Sur chaque graphe, la courbe pleine la plus foncée représente l'expérience incrémentale dès le premier document, sans fusion de classes passées ($b._{incr.@0}$). La courbe pleine plus claire représente le DER obtenu lorsqu'on effectue un regroupement global sur chaque sous-collection constituée des n premiers documents ($b._{glob.}$). Elle illustre le "point de départ" d'une expérience incrémentale initialisée par un regroupement global. Toutes les autres courbes en pointillés réguliers représentent les diverses expériences incrémentales, à partir d'une collection initiale constituée de 10, 20, 30 et 40 documents ($b._{incr.@10}$, $b._{incr.@20}$, $b._{incr.@30}$, $b._{incr.@40}$). Enfin, la courbe en pointillés irréguliers ($a._{glob}$) représente le DER obtenu après adaptation, initialisée par un regroupement global sur chaque sous-collection constituée des n premiers documents.

Premier constat, pour la collection LCP (graphe inférieur de la figure 8.7), on remarque que l'ordre des DER finaux de tous les systèmes *baseline* incrémentaux est fonction du nombre de documents servant à l'initialisation. Ainsi, le DER final de l'expérience incrémentale complète ($b._{incr.@0}$) atteint 23.1%, alors que le regroupement global avec les mêmes paramètres sur la collection complète ($b._{glob.}$) donne un DER de 18.1%. Ensuite, lorsqu'on s'intéresse aux expériences incrémentales intermédiaires, on observe que plus elles sont initialisées sur de grandes sous-collections, plus le DER final est bas. Les DER finaux s'échelonnent alors entre les deux valeurs précitées (18.1% et 23.1%). Ce n'est pas le cas sur BFM, où la courbe $b._{incr.@30}$ atteint un DER final de 18.9% alors qu'avec un regroupement global sur la collection complète ($b._{glob.}$), le DER inter-document est de 21.5%. Sur BFM, la règle "plus l'initialisation est tardive, meilleur est le DER final" n'est pas toujours strictement vérifiée, car on observe que le système $b._{incr.@30}$ donne de meilleurs résultats que le

système $b._{incr.@40}$. Enfin, on peut constater qu'aucun système *baseline*, qu'il soit incrémental ou global, n'égale les performances du système global adapté ($a._{glob}$). Si les données à regrouper sont les mêmes pour chaque expérience, comme le regroupement incrémental consiste à contraindre l'ordre et la combinatoire des regroupements, il est généralement sous-optimal par rapport au regroupement global (même si cela n'empêche pas des exceptions comme la courbe $b._{incr.@30}$: parfois, le regroupement incrémental strict peut "protéger" d'une mauvaise fusion de classes-locuteurs). En configuration *baseline*, les distances entre classes-locuteurs sont inchangées à l'ajout d'un nouveau document. Il semble donc qu'en moyenne on gagne à initialiser avec un regroupement global sur la plus grande collection possible.

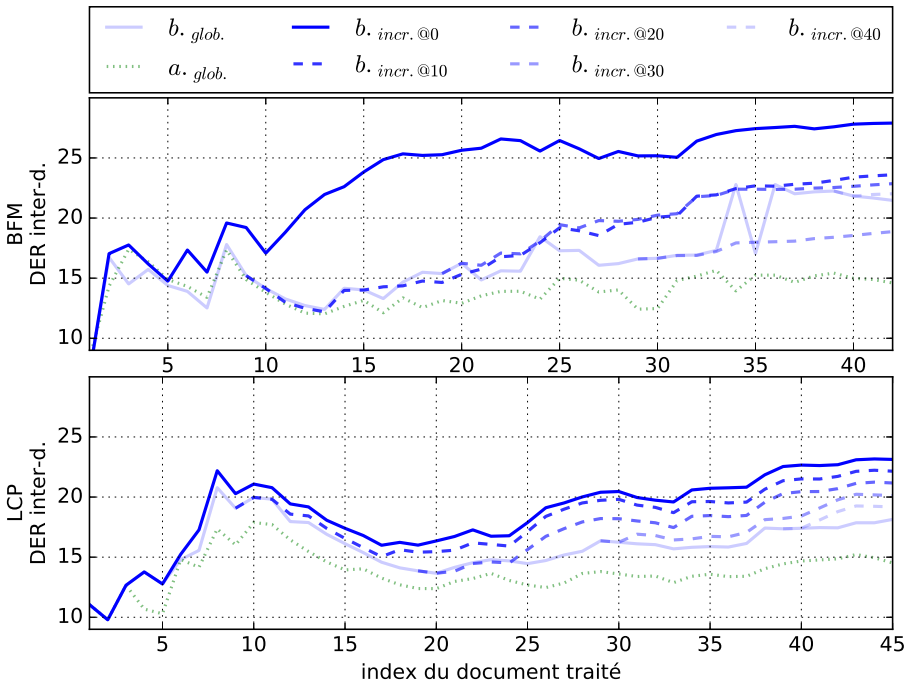

FIGURE 8.7 – Evolution comparative du DER inter-document sur les deux collections cibles, en fonction de l'index du document traité par le système incrémental *baseline* (selon le nombre de documents utilisés pour l'initialisation par regroupement global).

8.4.4.2 Avec adaptation incrémentale

Sur la figure 8.8, on représente le même type d'expériences, mais cette fois avec adaptation incrémentale. Nous avons reporté des graphes précédents les courbes des systèmes à regroupement global ($b._{glob.}$ et $a._{glob.}$), et avons ajouté les courbes des systèmes à regroupement incrémental, avec adaptation, initialisés sur des sous-collections de 10, 20, 30 ou 40 documents ($a._{incr.@10}$, $a._{incr.@20}$, $a._{incr.@30}$, $a._{incr.@40}$ en pointillés verts réguliers). Est également affichée la courbe du système à regroupement incrémental dès le premier épisode ($a._{incr.@0}$), courbe pleine de couleur verte.

En ajoutant l'adaptation, on constate que toutes les expériences incrémentales adaptées donnent un DER final meilleur que le DER global *baseline* ($b._{glob.}$) sur la

collection complète, que ce soit sur BFM ou LCP. Sur BFM, la meilleure stratégie est d'effectuer le regroupement incrémental adapté de zéro ($a._{incr.@0}$), le DER (14.6%) est alors compétitif avec le système global adapté ($a._{glob.}$). On n'observe pas de tendance particulière quant à l'ordre des systèmes selon l'initialisation. Sur LCP, le meilleur résultat est obtenu en initialisant sur la sous-collection des 30 premiers documents. Le DER final atteint alors 15.2%, tandis que le système global adapté sur la collection complète atteint 14.5%.

Si dans la configuration *baseline*, on constate que plus on initialise tard, plus on améliore les performances, ce n'est pas vérifié pour la configuration avec adaptation. Lorsqu'on adapte, les distances entre classes-locuteurs varient à l'ajout de chaque document, ce qui peut modifier l'ordre des regroupements à effectuer par rapport au regroupement global *baseline*. On ne peut donc pas conclure sur l'optimalité d'une initialisation donnée. On remarque seulement que les systèmes incrémentaux avec adaptation ($a._{incr.@...}$) surpassent le système *baseline* à regroupement global ($b._{glob.}$), et peuvent être compétitifs avec une approche à regroupement global avec l'adaptation ($a._{glob.}$).

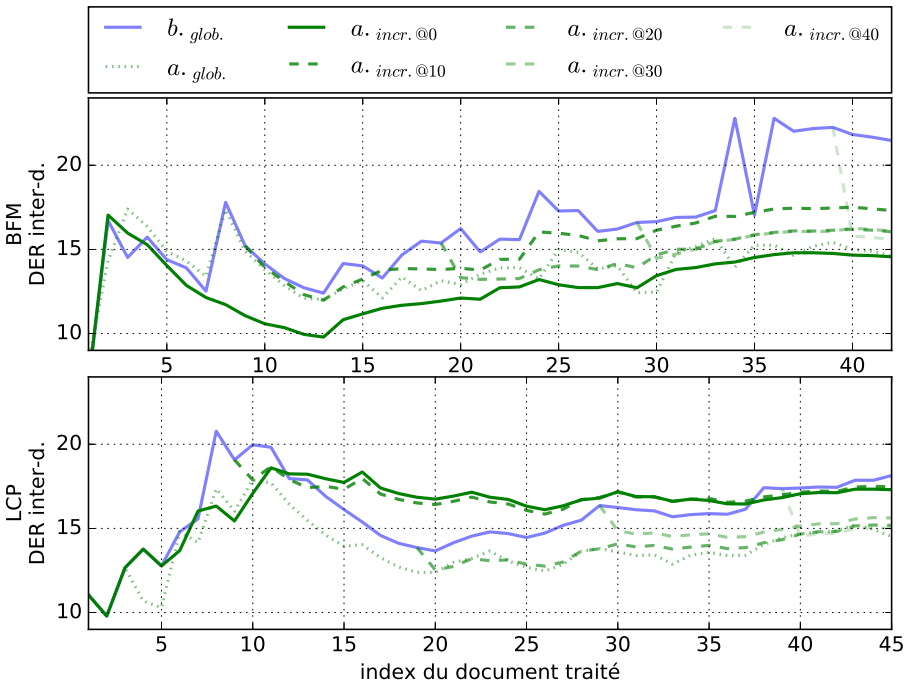

FIGURE 8.8 – Evolution comparative du DER inter-document sur les deux collections cibles, en fonction de l'index du document traité par le système incrémental *adapt* (selon le nombre de documents utilisés pour l'initialisation par regroupement global).

8.4.5 Analyse d'erreur

Nous proposons maintenant de faire un focus sur l'analyse des erreurs du système incrémental. Comme nous avons pu l'observer, l'ajout de certains documents fait beaucoup plus augmenter le DER inter-document que d'autres. Initialement,

nous avons donc cherché à étudier la corrélation entre DER intra-document (avant regroupement inter-document) et DER inter-document, en pensant que l'effet d'un DER intra-document élevé s'observerait sur la courbe du DER inter-document. Cependant, nous rapidement réalisé que ce n'était pas aussi simple.

Au lieu d'analyser le DER intra-document, nous proposons plutôt d'étudier les erreurs de temps de parole. L'idée est la suivante : le regroupement intra-document produit une certaine quantité d'erreurs qui ne seront pas corrigibles : certains segments de parole, ou classes-locuteurs intra-document, peuvent contenir la voix de plus d'un locuteur. Or, à la fin du regroupement intra-document, on décide de réduire chaque classe-locuteur à la moyenne de ses *i-vectors*. Sa composition ne peut donc jamais être remise en cause. Dans de rares cas, il peut seulement y avoir un regroupement intra-document supplémentaire lors du regroupement inter-document, lorsque le seuil de regroupement inter-document λ_X est supérieur au seuil de regroupement intra-document λ_I.

Lorsqu'a lieu le regroupement inter-document, d'autres erreurs peuvent apparaître : une classe-locuteur représentant un locuteur ponctuel peut être fusionnée avec une classe préexistante, une classe représentant un locuteur récurrent peut ne pas être fusionnée alors qu'elle devrait l'être... Lorsqu'on traite un nouveau document de la collection, c'est donc la somme de ces erreurs (intra- et inter-document) qui contribuent au DER inter-document. Ainsi, on peut de visualiser, après l'ajout de chaque document, la durée totale d'erreur des deux types. C'est ce que nous avons fait dans les figures 8.9 pour LCP et 8.10 pour BFM, avec la PLDA pour le calcul des similarités.

Dans chaque figure, on présente le DER inter-document des trois systèmes incrémentaux $baseline_{incr.}$, $adapt_{incr.}$ et $adapt_{incr.+f}$ sur le graphe supérieur. A chaque expérience incrémentale est associé un histogramme cumulé détaillant l'évolution des durées d'erreur de type intra- (en bleu) ou inter-document (en jaune), au traitement de chaque nouveau document. Chaque barre d'histogramme représente la différence de la durée totale d'erreur entre la collection de taille N et celle de taille $N-1$. Ainsi, le DER inter-document après avoir traité un document k correspond à la somme des erreurs de tous les documents précédents.

Cette représentation permet de mettre en évidence des phénomènes particuliers. Par exemple, sur l'histogramme supérieur de la figure 8.9 (*baseline*), on observe au $25^{ème}$ document une barre jaune importante alors que la barre mauve est très petite. Cela signifie que sur ce document, le regroupement intra-document a fait très peu d'erreurs, mais qu'un ou plusieurs mauvais regroupements inter-document ont eu lieu (fusion de deux locuteurs différents, non fusion de deux locuteurs identiques...). Comme pour ce document, la somme d'erreur des deux types est importante, on voit sur le graphe que la courbe de DER inter-document $baseline_{incr.}$ augmente fortement. L'inverse est aussi vrai : l'apport d'une faible quantité d'erreur fait diminuer

le DER inter-document (par exemple autour du $15^{ème}$ de LCP).

Autre phénomène remarquable : certaines barres jaunes sont orientées vers le bas sur le dernier histogramme ($adapt + f$) (par exemple celle du $42^{ème}$ document de LCP), ce qui signifie qu'à l'ajout du document, la durée total d'erreur inter-document a diminué ! En réalité, cela signifie qu'au moins deux classes-locuteurs passées représentant un même locuteur ont fusionné.

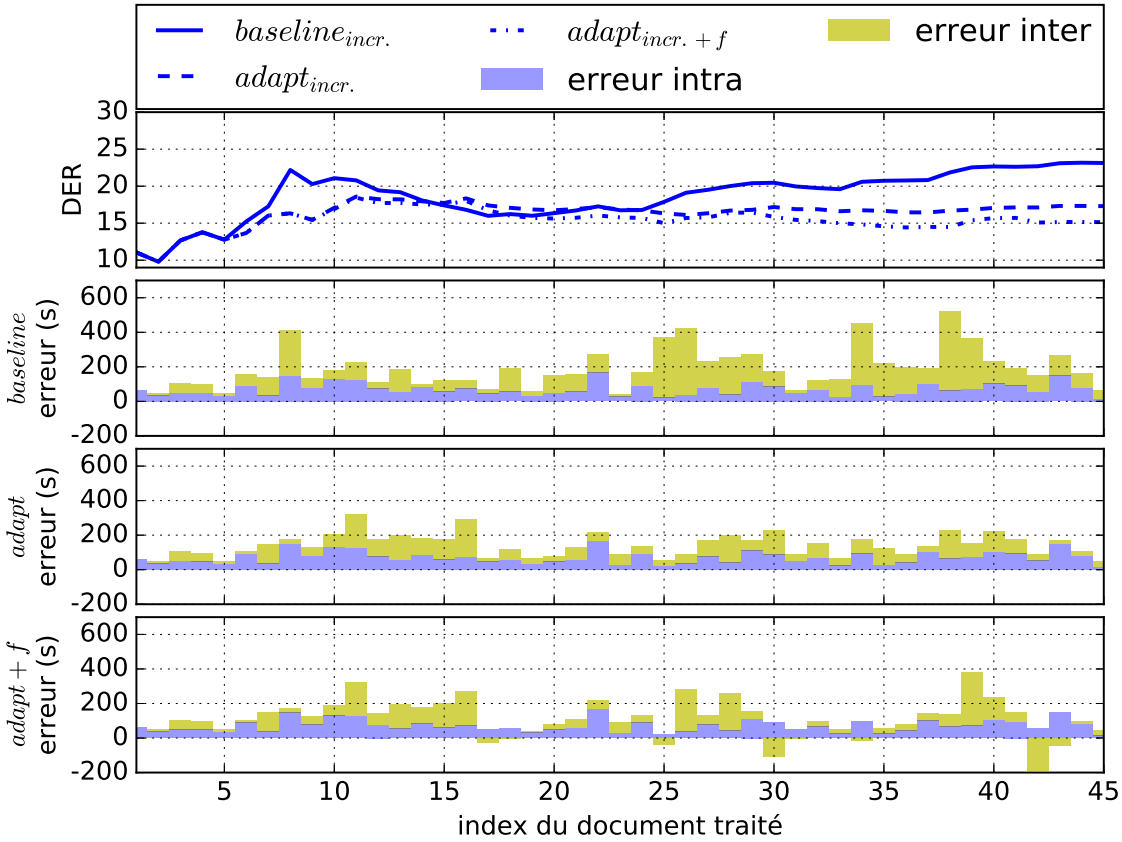

FIGURE 8.9 – Analyse comparative des durées d'erreurs apportées par chaque document au cours du regroupement incrémental, pour le système HAC/PLDA appliqué à la collection LCP (*baseline* ou avec adaptation, avec ou sans fusions tardives autorisées).

Pour la collection LCP (figure 8.9), les trois systèmes ont des performances bien distinctes. La *baseline* est à 23.1%, l'*adapt* à 17.3% et l'$adapt + f$ à 15.1% de DER inter-document sur la collection complète. Or l'erreur intra-document est identique quel que soit le système, puisque le regroupement intra-document est fait par le système *baseline* et n'est pas remis en question par l'adaptation. Corollaire, plus le DER final d'un système est bas, plus la quantité d'erreur inter-document apportée par l'ensemble des documents est faible. Ainsi, on peut constater qu'avec les systèmes adaptés, l'erreur inter-document est grandement réduite par rapport à la *baseline*. Si le regroupement inter-document était parfait, seules les erreurs intra-document subsisteraient. En adaptant avec fusion, on remarque également que

certains documents font diminuer l'erreur inter-document.

Pour la collection BFM (figure 8.10), l'adaptation a un effet impressionnant, puisque sur les 20 premiers documents, quasiment aucune erreur inter-document n'est faite, ce qui explique le DER inter document inférieur à 10% à l'ajout du $12^{ème}$ document. De manière générale, plus un document est ajouté tard, plus la combinatoire des regroupements possibles est importante, ce qui augmente le risque d'erreur de regroupement inter-document. Si les collections traitées sont trop petites pour certifier ce phénomène, on peut quand même penser que l'erreur inter-document apportée va, en moyenne, augmenter à mesure que l'on ajoute des documents. Notons également pour le système *baseline* l'importante quantité d'erreur apportée par certains documents. Par exemple, le $33^{ème}$ apporte un cumul d'erreur intra- et inter-document proche de 1500 secondes, c'est-à-dire 25 minutes d'erreur pour un document qui dure une heure. Enfin, pour le système *adapt* (deuxième histogramme), on observe parfois une diminution de l'erreur inter-document totale à l'ajout de certains documents. Comme les fusions de classes passées sont interdites, il s'agit de regroupements intra-document au sein du document en cours de regroupement inter-document.

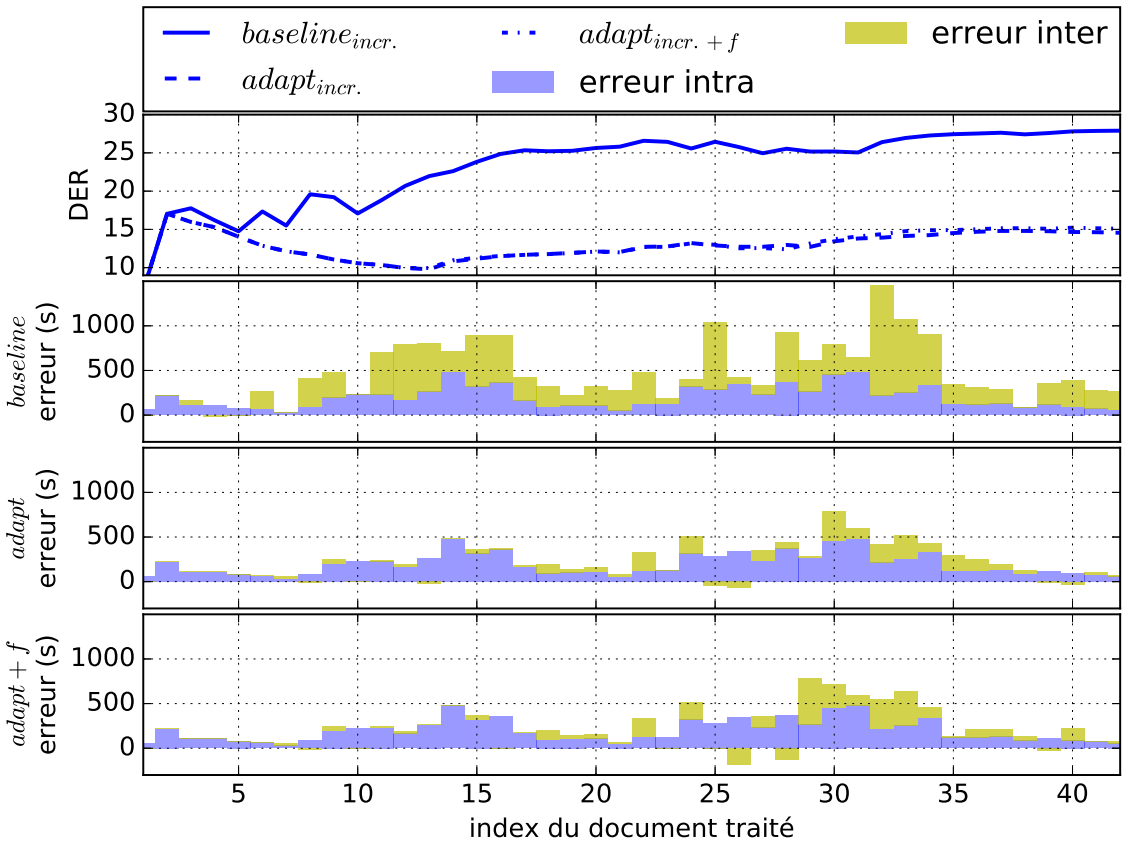

FIGURE 8.10 – Analyse comparative des durées d'erreurs apportées par chaque document au cours du regroupement incrémental, pour le système HAC/PLDA appliqué à la collection BFM (*baseline* ou avec adaptation, avec ou sans fusions tardives autorisées).

8.4.6 Concaténation des collections

Dans cette partie, nous proposons de traiter de façon incrémentale la concaténation des collections BFM et LCP. Dans cette nouvelle collection, les documents sont triés dans l'ordre chronologique de diffusion. Comme les deux émissions ont été diffusées sur la même période, traiter la concaténation dans l'ordre chronologique revient à regrouper alternativement des épisodes de chacune. Ce chevauchement temporel des diffusions implique également que 50 locuteurs sont communs aux deux émissions. Ce sont très majoritairement des invités. Concaténer les deux collections permet d'avoir plus de données à disposition pour réaliser l'adaptation. De plus, la présence de locuteurs identiques dans les deux collections pourrait permettre d'améliorer la modélisation de la variabilité inter-document. Cela peut également permettre d'étudier la robustesse de l'approche : travailler sur des données plus nombreuses et variées peut augmenter le nombre d'erreurs de regroupement.

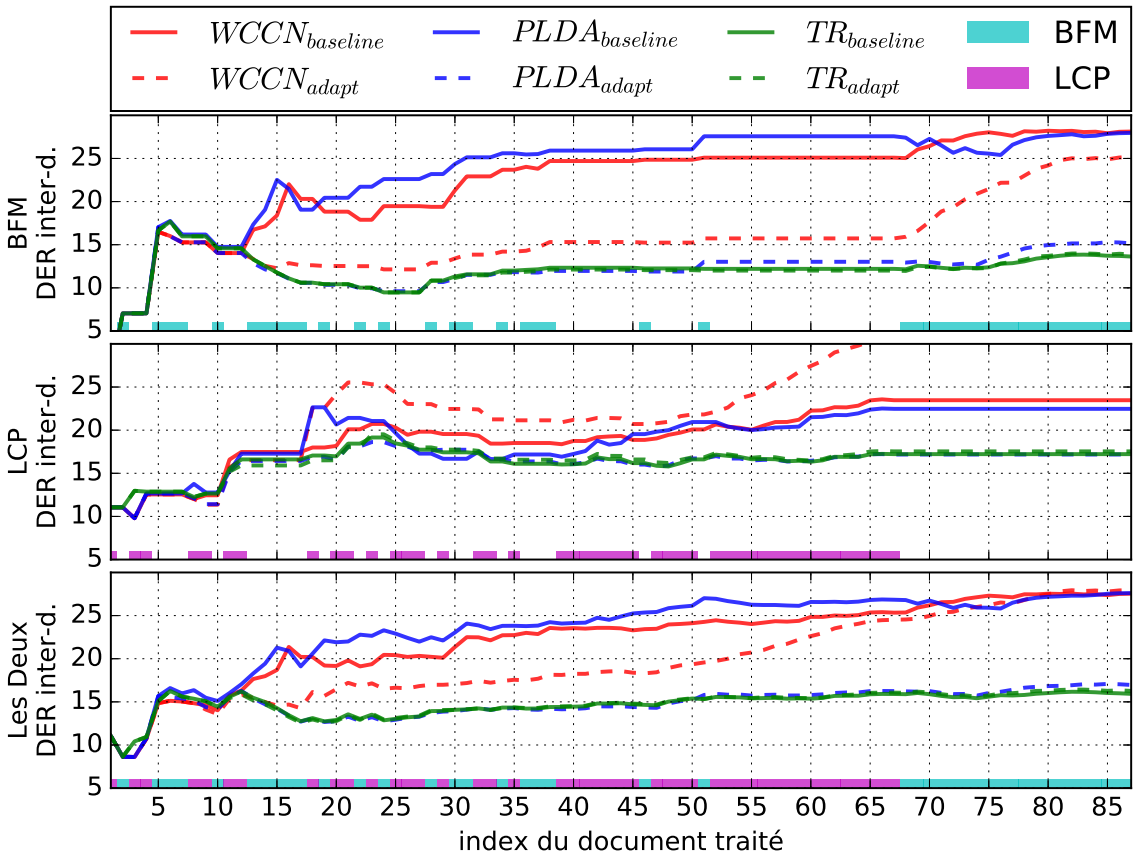

FIGURE 8.11 – Evolution comparative du DER inter-document sur la concaténation des deux collections cibles, pour les systèmes incrémentaux *baseline* et adaptés (*adapt*). Les fusions tardives sont interdites.

8.4.6.1 Incremental strict

La figure 8.11 illustre les performances des différents systèmes testés, avec la contrainte de non modification du passé. La figure est divisée en trois graphes. Le

graphe supérieur indique le DER de la collection BFM, celui du milieu renseigne sur le DER de LCP et le dernier présente le DER inter-document calculé sur la concaténation des deux collections. Au niveau de l'axe des abscisses, nous avons indiqué sous forme d'histogramme coloré la collection d'appartenance de chaque document ajouté. Ceci nous permet d'observer si l'ajout d'un document issu d'une collection a un effet sur le DER de l'autre, ainsi que l'entrelacement des diffusions. Si la première moitié de la collection agrégée montre une alternance forte entre les documents de BFM et LCP, on voit que plusieurs documents de la même collection sont ajoutés consécutivement pour la seconde moitié.

L'allure générale des courbes est semblable aux mêmes expériences réalisées sur les collections distinctes de la section 8.4.2. On peut visuellement constater que lorsque le système traite un document d'une collection donnée, le DER spécifique à l'autre collection ne varie pas. Pour faciliter l'analyse, nous reportons les DER finaux de chaque système dans la table 8.3. Les performances BFM_1 et LCP_1 sont celles obtenues lorsqu'on traite chaque collection séparément, tandis que BFM_2 et LCP_2 correspondent aux expériences sur la concaténation des deux collections, mais en évaluant le DER spécifique à chaque collection. La dernière ligne indique le DER final calculé sur la concaténation des deux collections.

système	HAC/WCCN		HAC/PLDA		CC/TR	
config.	*baseline*	*adapt*	*baseline*	*adapt*	*baseline*	*adapt*
BFM_1	28.0	17.2	27.9	14.6	13.6	**13.5**
BFM_2	28.1	25.2	28.0	15.2	**13.6**	13.9
LCP_1	23.5	26.8	23.1	17.3	**17.2**	17.3
LCP_2	23.5	30.5	22.5	**17.2**	**17.2**	17.5
LesDeux	27.5	27.9	27.6	17.0	**15.9**	16.3

TABLE 8.3: Comparatif des DER finaux des systèmes incrémentaux, selon qu'on traite les collections BFM et LCP distinctement ou ensemble. Les fusions tardives sont interdites.

Dans la table 8.3, on constate qu'avec les systèmes *baseline*, les DER finaux obtenus sur les deux collections varient peu, que les collections soient traitées séparément ou non. En revanche, lorsqu'on considère les systèmes *adapt*, on peut observer une dégradation minime allant jusqu'à 0.6 point pour les approches HAC/PLDA et CC/TR, mais importante avec le système HAC/WCCN. Par rapport au système adapté sur les deux collections séparément, le DER final de BFM augmente de 8 points et celui de LCP de 3.7 points.

Parmi les trois systèmes, sur la concaténation des deux collections, le système CC/TR est le plus performant (*baseline*), avec un DER à 15.9%, suivi de près par le système HAC/PLDA adapté à 17.0%. On constate que l'adaptation dégrade légèrement les performances du système CC/TR, qui passe à 16.3%. Enfin, remarquons que le DER *LesDeux* n'est pas une combinaison linéaire des DER BFM_2 et LCP_2

(ce point sera discuté à la section 8.4.6.3, voir configuration *a posteriori*).

8.4.6.2 Incrémental avec fusion

Comme nous avons vu à la section 8.4.3 qu'en autorisant les regroupements de classes-locuteurs passées, le système CC/TR pouvait donner de meilleures performances, nous présentons les performances des mêmes systèmes que précédemment mais en autorisant la fusion tardive de classes préexistantes à la figure 8.12 et dans la table 8.4.

Avec la contrainte, le système appliqué à la concaténation des deux collections avait pour propriété de ne pas modifier le DER d'une collection lorsqu'un épisode de l'autre collection était traité. Cette propriété ne s'applique pas ici, et on peut très bien voir sur la figure que le DER peut évoluer pour les deux collections, quel que soit l'épisode ajouté (par exemple l'ajout du $60^{ème}$ document, qui appartient à LCP, fait augmenter le DER de BFM pour le système TR_{adapt} (courbe bleue en pointillés du graphe supérieur de la figure 8.12).

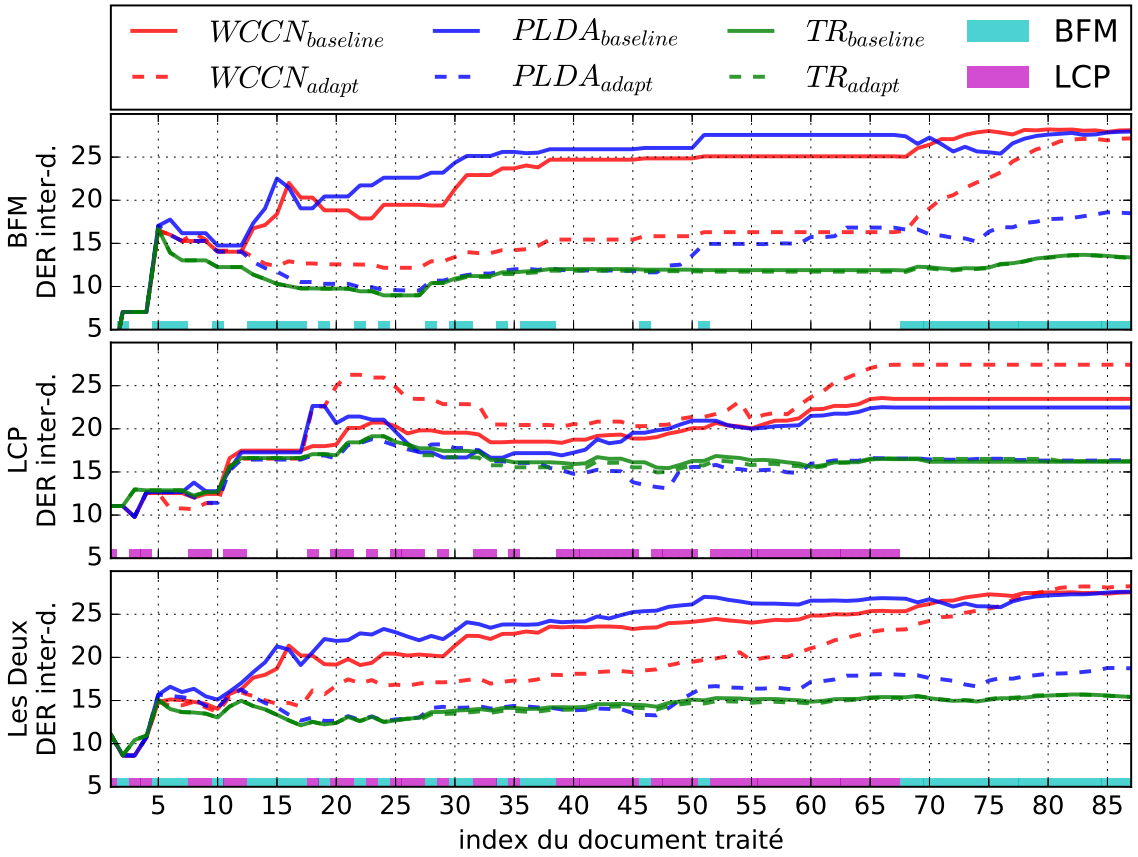

FIGURE 8.12 – Evolution comparative du DER inter-document sur la concaténation des deux collections cibles, pour les systèmes incrémentaux *baseline* et adaptés (*adapt*). Les fusions tardives sont permises.

Si on étudie les DER reportés à la table 8.4, on note que la relâche de la contrainte permet de faire baisser le DER du système CC/TR à 15.4%, contre 15.9% aupa-

ravant. C'est toujours le meilleur système, qu'il soit en configuration *baseline* ou *adapt*. Lorsqu'on traitait les collections séparément, on constatait que l'imposition de la contrainte de non fusion faisait perdre quelques points de DER par rapport à un système incrémental plus permissif. Sur la concaténation des deux collections, avec les approches WCCN et PLDA, le système sans contrainte est moins performant que le système avec contrainte. Avec la WCCN, le DER *adapt* sur la concaténation des deux collections passe de 27.9% à 28.2%, et avec la PLDA de 17.0% à 18.7%. Ceci laisse donc penser que des regroupements illicites ont lieu entre les deux collections, au détriment des performances.

système	HAC/WCCN		HAC/PLDA		CC/TR	
config.	*baseline*	*adapt*	*baseline*	*adapt*	*baseline*	*adapt*
BFM_1	28.0	17.2	27.9	14.6	13.4	**13.3**
BFM_2	28.1	25.2	28.0	16.4	13.4	**13.3**
LCP_1	23.5	31.9	23.1	15.1	16.5	**16.2**
LCP_2	23.5	27.4	22.5	18.5	**16.2**	16.3
LesDeux	27.5	28.2	27.6	18.7	**15.4**	**15.4**

TABLE 8.4: Comparatif des DER finaux des systèmes incrémentaux, selon qu'on traite les collections BFM et LCP distinctement ou ensemble. Les fusions tardives sont permises.

8.4.6.3 Analyse en locuteurs

Lorsqu'on traite les collections cibles ensemble, on peut s'interroger sur l'influence des regroupements entre collections (ou inter-collection). D'une part, certains locuteurs apparaissent à la fois dans LCP et BFM : on en compte tout de même 50, tous des invités (forcément récurrents). D'autre part, le fait de mélanger les collections augmente le risque de voir de mauvais regroupements s'opérer. Afin de voir si traiter les collections ensemble permet de regrouper les locuteurs entre collections, nous comparons trois expériences, pour un système donné (par exemple, le système incrémental CC/TR avec fusion tardive autorisée) :

- **Configuration "interdits"** : dans la première expérience, on traite les deux collections séparément puis on calcule le DER sur la concaténation des résultats de SRL, en s'assurant qu'une classe-locuteur issue d'une collection ne peut avoir le même label que n'importe quelle classe-locuteur issue de l'autre collection. Autrement dit, cela revient à évaluer la qualité d'un système qui ne réaliserait aucun regroupement inter-collection si les collections étaient traitées de façon entrelacée (regroupements inter-collection interdits).
- **Configuration "autorisés"** : dans la deuxième, on traite les deux collections ensemble, de façon entrelacée, puis on calcule le DER sur le résultat de SRL, sans altérer les résultats. Cela correspond au cas classique où l'on autorise des regroupements inter-collection. Ce second DER devrait donc être différent

du premier si de tels regroupements ont lieu (regroupements inter-collection autorisés).

- **Configuration "*a posteriori*"** : dans la troisième expérience, on traite les deux collections séparément, puis on calcule le DER sur chaque collection séparément (ce qui a pour effet de fournir un appariement entre classes-locuteurs et locuteurs de référence, sur chaque collection). Ensuite, on fusionne les appariements (par exemple, la classe-locuteur issue de BFM attribuée à Nicolas Sarkozy est renommée pour avoir le même label que la classe-locuteur issue de LCP attribuée aussi à Nicolas Sarkozy). Les classes-locuteurs non appariées ne sont pas modifiées. Cette expérience simule le cas où les regroupements inter-collection adequats seraient effectués *a posteriori* (regroupements inter-collection *a posteriori*). Dans ce cas, le DER calculé sur la concaténation des deux collections sera égal à la combinaison linéaire des DER calculés sur chaque collection séparée, fonction de leur durée totale respective.

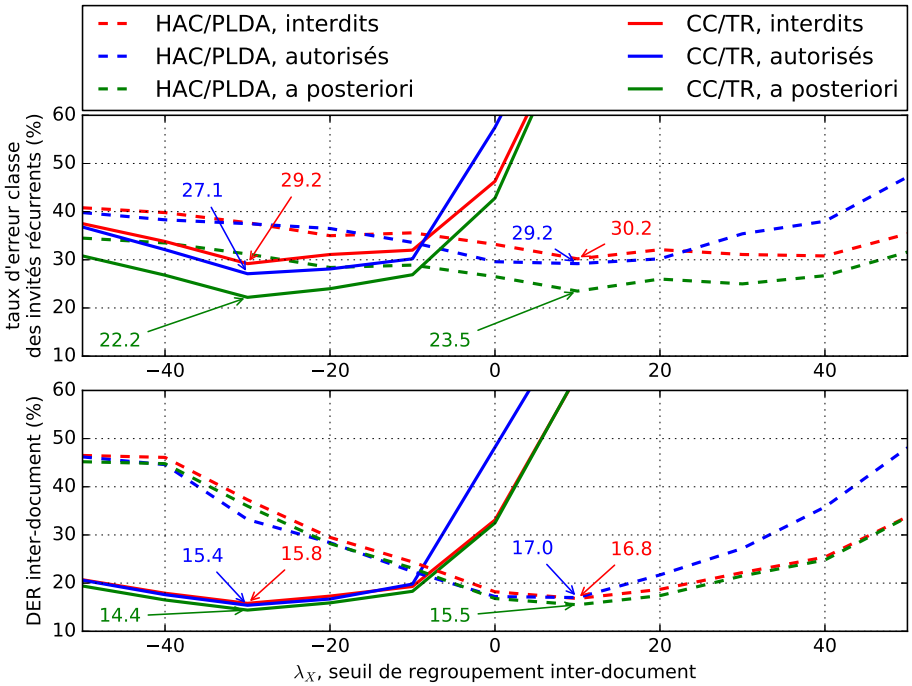

FIGURE 8.13 – Evolution comparative du taux d'erreur classe des invités récurrents et du DER selon la stratégie de regroupement inter-collection (regroupements interdits, autorisés ou *a posteriori*), pour les systèmes incrémentaux HAC/PLDA (avec adaptation, sans fusions tardives) et CC/TR (sans adaptation, avec fusions tardives).

Les résultats de ces trois expériences comparatives appliquées aux systèmes HAC/PLDA (avec adaptation, sans fusions tardives) et CC/TR (sans adaptation, avec fusions tardives) sont présentés à la figure 8.13. On y représente l'évolution du taux d'erreur classe des invités récurrents et du DER inter-document en fonction

du seuil de regroupement λ_X. Pour chaque système, les valeurs sont annotées pour le seuil de regroupement λ_X optimal ($\lambda_X = -30$ pour le système CC/TR, $\lambda_X = 10$ pour le système HAC/PLDA). Logiquement, la courbe verte (regroupements *a posteriori*) est située sous la courbe rouge correspondante, à chaque seuil (regroupements interdits).

La courbe bleue correspond au fonctionnement réel du système lorsqu'on traite les collections emsemble, et on peut constater qu'au seuil optimal, le taux d'erreur classe des invités récurrents est très légèrement meilleur qu'en traitant les collections séparément (27.1% contre 29.2% pour le système CC/TR, par exemple), mais encore loin de la performance qu'on obtiendrait si après avoir traité les collections séparément, on effectuait les regroupements inter-collection pertinents (22.2% pour le système CC/TR). Ceci indique qu'il y a en réalité très peu de regroupements inter-collection réalisés.

Quand on lit le DER, on constate que les différences sont faibles : c'est surtout en concentrant l'analyse sur les invités récurrents qu'on peut observer le phénomène. On peut noter qu'au-delà d'un certain seuil (-10 pour CC/TR, 0 pour HAC/PLDA), le DER "bleu" devient supérieur au DER "rouge", ce qui signifie que l'autorisation des regroupements inter-collection devient contre-productive : il y a plus de mauvais regroupements que de bons regroupements.

En résumé, au seuil optimal, le fait de traiter les collections ensemble permet de gagner quelques points sur les invités récurrents inter-collection, même si cela ne se traduit pas au niveau du DER. Cependant, il faut bien noter qu'une mauvaise calibration du système peut avoir des effets néfastes, car à trop regrouper, on augmente le nombre de regroupements inter-collection illicites, ce qui pénalise fortement le DER.

8.5 Bilan

Dans ce chapitre, nous avons abordé la question de la SRL incrémentale de collection, à l'aide des outils développés dans les chapitres précédents. Pour répondre aux contraintes applicatives de la SRL incrémentale, nous avons modifié les méthodes de regroupement HAC et CC étudiées dans le cadre du regroupement global, et avons proposé une stratégie d'adaptation, dont nous avions vu qu'elle pouvait grandement améliorer les performances en regroupement global.

Comparé au regroupement global, le regroupement incrémental strict est sous-optimal, ce qui explique le DER inter-document plus élevé en configuration incrémentale stricte, sans adaptation, pour les trois systèmes évalués (HAC/WCCN, HAC/PLDA et CC/TR). Cependant, en relâchant la contrainte de non fusion de classes passées, nous avons mis en évidence le fait que le regroupement CC incrémental était équivalent à sa version globale, de par son caractère associatif. Que la

fusion soit autorisée ou non, le système le plus performant sans adaptation est le système CC/TR, suivi de l'approche HAC/PLDA.

L'ajout de l'adaptation apporte peu au système CC/TR, qui était déjà très performant, et permet au système HAC/PLDA de rattraper les performances du réseau de neurones, même si cela nécessite la calibration d'un paramètre supplémentaire. C'est un résultat intéressant sachant que le regroupement HAC est plus robuste que le regroupement CC, comme nous l'avons vu à la section 4.2.2.3.

Concernant l'initialisation du regroupement incrémental, décider d'effectuer un regroupement global sur une partie de la collection se révèle pertinent sans adaptation, car il permet d'améliorer les résultats par rapport à un regroupement incrémental de zéro. En revanche, avec adaptation, il est difficile de tirer des conclusions, regrouper de façon incrémentale dès le premier épisode peut très bien être la stratégie la plus efficace.

L'analyse des résultats a mis en évidence deux types d'erreur contribuant au DER inter-document : l'erreur due au regroupement intra-document, en majeure partie irréductible, et l'erreur due au regroupement inter-document, réductible. Sur le système HAC/PLDA, nous avons mis en évidence le fait que l'adaptation permettait de réduire une grande partie de cette erreur inter-document.

Par ailleurs, une étude sur une troisième collection, résultat de l'entrelacement chronologique des collections BFM et LCP, montre que les systèmes HAC/PLDA et CC/TR, sont, dans une configuration incrémentale stricte, les plus performants et les plus robustes, puisque les performances sur LCP et BFM varient très peu selon qu'elles sont traitées séparément ou ensemble. En revanche, dans la configuration incrémentale avec fusion et adaptation, seul le système CC/TR reste robuste, les performances se dégradant de quelques points avec la PLDA. L'analyse des regroupements inter-collection a montré qu'il n'y avait pas beaucoup à gagner à traiter les collections de façon entrelacée, et que cela pouvait représenter un risque si le seuil de regroupement n'est pas bien calibré. Pour traiter un grand nombre de données variées, il est donc peut-être plus judicieux de créer des collections qu'on traitera séparément avant d'envisager une stratégie de regroupement inter-collection.

L'ensemble de ces résultats nous incite donc à recommander le choix des systèmes CC/TR, avec ou sans adaptation, ou HAC/PLDA, avec adaptation, pour la SRL incrémentale de collection. Sur les collections séparées, leurs performances sont équivalentes et le choix du système HAC/PLDA adapté peut être plus sage compte tenu de la robustesse du regroupement HAC. En revanche, si l'objectif est de traiter les collections ensemble, de façon entrelacée, le système CC/TR est clairement le plus performant, car il préserve les performances inhérentes à chaque collection. Autre résultat remarquable, lorsque la contrainte de non regroupement des classes passées peut être levée, le choix du regroupement CC avec la compensation neuronale de variabilité se révèle pertinent, car il est équivalent au même système à regroupement

global et est donc aussi performant.

Chapitre 9

Conclusions et Perspectives

Les travaux présentés dans ce manuscrit concernent la segmentation et regroupement en locuteurs de collections d'archives audiovisuelles, ayant pour caractéristique la présence de locuteurs identiques dans plusieurs documents (appelés locuteurs récurrents, par opposition aux locuteurs ponctuels). Dans l'optique de proposer un nouveau moyen d'explorer les collections, via les locuteurs récurrents, nous avons choisi de travailler sur deux émissions télévisées d'une quarantaine d'épisodes, ce qui correspond au nombre d'épisodes généralement diffusé au cours d'une année (pause estivale comprise).

Le défi posé par la tâche de SRL de collection est l'adaptation au domaine des modèles de variabilité intra-locuteur/inter-document. En effet, même s'il existe une quantité croissante de données librement disponibles, la nécessité de disposer de données du domaine acoustique ciblé, annotées manuellement, pour entraîner les modèles pose actuellement problème. Cette thèse cherchait à répondre à deux problématiques :

- Peut-on exploiter des données non annotées du domaine acoustique cible pour améliorer les performances de SRL de collection ?
- Quelle est l'effet réel d'un système de SRL de collection sur les locuteurs de la collection ?

9.1 Conclusions

Pour répondre à ces problématiques, nous avons d'abord défini deux collections expérimentales : BFM et LCP, constituées à partir des données des campagnes d'évaluation ETAPE, ESTER et REPERE. Chacune comprend une quarantaine de documents télévisuels et contient quelques centaines de locuteurs. Parmi ces locuteurs, les récurrents y parlent plus de la moitié du temps, ce qui rend les deux collections particulièrement intéressantes dans le cadre d'une exploration via les locuteurs. A partir des annotations en rôle des locuteurs, nous avons constitué quatre classes de locuteurs pour l'analyse, selon qu'ils sont journalistes ou invités, ponctuels ou récur-

rents. Pour l'apprentissage des modèles du système de SRL, nous avons constitué un corpus d'apprentissage constitué de plusieurs centaines de documents issus d'émissions radiophoniques, afin de se placer dans un contexte propice à l'adaptation au domaine.

Dans une première partie, nous avons étudié un système de SRL de collection à l'état de l'art, reposant sur une architecture à regroupement global, en deux passes : SRL intra-document d'abord, puis regroupement inter-document. Nous avons pris le parti de ne pas remettre en cause les briques de segmentation et de travailler spécifiquement sur les problématiques de regroupement intra- et surtout inter-document. Pour ce faire, nous avons utilisé la modélisation *i-vector* et les compensations de variabilité inter-document WCCN ou PLDA, associées aux regroupements hiérarchique agglomératif ascendant (HAC), conservatif, ou en composantes connexes (CC), associatif. Les résultats ont montré que des progrès étaient réalisables au niveau du regroupement inter-document, tandis que l'analyse en locuteurs a mis en évidence que les invités récurrents étaient les plus mal regroupés et qu'il existait un seuil de regroupement optimal différent pour chaque type de locuteur. Nous avons également montré qu'il existait une part d'erreur due aux briques de segmentation, irréductible et non soluble par le regroupement.

Ce premier résultat nous a amené à proposer une nouvelle méthode de compensation de variabilité intra-locuteur/inter-document (TR) basée sur les *i-vectors* et utilisant un réseau de neurones appris selon le paradigme *triplet loss*. Les performances de SRL sur les deux collections cibles ont montré que l'approche, combinée à un regroupement CC, surpasse l'état de l'art, notamment la PLDA combinée à un regroupement HAC, même si le nombre de paramètres à optimiser est plus important avec la compensation TR. L'analyse des résultats a montré qu'avec le réseau de neurones, le seuil de regroupement est optimal pour les quatre types de locuteurs en même temps, et pour les deux collections, ce qui facilite la calibration.

Dans la seconde partie, nous avons proposé d'utiliser les données des collections à traiter pour améliorer l'estimation des modèles de compensation de variabilité intra-locuteur inter-document, selon un mécanisme d'adaptation au domaine. La méthode d'adaptation itérative proposée dépend d'un coefficient de pondération permettant d'équilibrer l'influence des données d'apprentissage initiales, annotées manuellement, et des données des collections cibles, pouvant contenir des erreurs. La méthode permet d'améliorer itérativement les performances, particulièrement celles de systèmes utilisant le regroupement HAC. La méthode présente l'avantage d'être robuste au choix du seuil de regroupement. Nous avons également montré qu'elle peut s'appliquer sur des collections de différentes tailles, et que pour les approches WCCN et PLDA, l'optimalité du coefficient d'adaptation dépend de la taille de la collection cible, exprimée en nombre de locuteurs récurrents. L'analyse en locuteurs a montré que le caractère itératif bénéficie principalement aux locuteurs récurrents.

En fonction des besoins applicatifs, le regroupement inter-document peut être global ou incrémental, selon que l'on souhaite traiter la collection en une fois ou en suivant sa chronologie. Les deux stratégies proposées pour améliorer la compensation de variabilité intra-locuteur/inter-document ont donc été également évaluées avec une architecture à regroupement incrémental et ont confirmé leur efficacité. Par rapport à un système à regroupement global, les approches proposées ont montré que la dégradation des performances était limitée. Si l'adaptation au domaine apporte peu au réseau de neurones, initialement très performant avec un regroupement CC, elle permet à la PLDA, avec un regroupement HAC, d'approcher ses performances, moyennant un paramètre supplémentaire à calibrer. L'analyse a montré qu'il existait deux types d'erreur : intra-document, irréductible, et inter-document, réductible, que l'adaptation au domaine permet de diminuer lors de regroupements incrémentaux.

Par ailleurs, pour comprendre le fonctionnement réel des systèmes de SRL, nous avons proposé différentes méthodes d'analyse en locuteurs et de visualisation des résultats. D'une part, pour faire le lien entre le taux d'erreur de SRL (DER) et les performances propres à chaque locuteur et chaque type de locuteur, nous avons défini les notions de taux d'erreur nominal et taux d'erreur classe, qui nous ont permis de visualiser les contributions réelles des locuteurs au taux d'erreur de SRL, par exemple à la figure 4.3. D'autre part, pour étudier l'évolution des performances individuelles lors de l'adaptation itérative des modèles, nous avons présenté une visualisation nominale de l'évolution des performances, par exemple à la figure 6.14.

9.2 Limites

Les deux méthodes proposées dans ce manuscrit ont donné des résultats intéressants. D'une part le réseau de neurones de compensation de variabilité associé à un regroupement en composantes connexes s'est révélé plus efficace que les méthodes à l'état de l'art. Son seul inconvénient est le nombre plus élevé de paramètres pour son apprentissage. D'autre part, l'approche consistant à utiliser les données de la collection à traiter pour adapter les modèles de compensation a fait ses preuves avec la compensation PLDA et un regroupement hiérarchique agglomératif à saut maximum. Si elle nécessite un paramètre supplémentaire à optimiser (le coefficient d'adaptation), elle est peut-être plus robuste que le réseau de neurones quand l'écart de domaine acoustique avec les données d'apprentissage est grand.

Même si les deux approches ont permis de réduire les taux d'erreur de SRL de collection, on a pu voir que l'erreur inter-document n'était pas complètement supprimée et qu'il restait surtout une part d'erreur irréductible liée à la segmentation, qui nécessite des travaux complémentaires, notamment en ce qui concerne la gestion de la parole superposée. De plus, l'analyse en locuteurs a montré que les invités récurrents étaient les plus difficiles à regrouper, et que lorsqu'on traite les collections de

façon entrelacée, peu de regroupements inter-collection s'effectuaient. Ceci est problématique si on souhaite s'appuyer sur ces locuteurs pour explorer les données de proche en proche : en général, les locuteurs les plus intéressants pour l'exploration sont ces invités récurrents (hommes politiques, artistes...). Ces résultats mitigés s'expliquent par la faible durée de parole par document de ces individus, et par la grande variabilité inter-document qui les caractérise. Cela montre que la question de la modélisation/compensation de cette variabilité n'est pas encore complètement résolue, là où l'efficacité des méthodes de regroupement utilisées ne semble plus à prouver.

L'étude sur la taille des collections et leur influence sur les paramètres d'adaptation a également montré que les observations réalisées restent à confirmer sur des collections plus volumineuses. Cela pose également la question du temps de traitement. Le système de SRL proposé est relativement léger dans le sens où il ne nécessite de traiter l'audio qu'une fois, les regroupements s'effectuant ensuite sur la base de quelques dizaines d'*i-vectors* par document, ce qui est également vrai pour l'adaptation itérative. Cependant, nous n'avons pas quantifié la taille de collection limite pour notre approche de regroupement, de complexité quadratique. Remarquons simplement que les performances de SRL incrémentale restent proches des performances de SRL globale et permettent de réduire la complexité.

9.3 Perspectives

Si les travaux présentés dans ce manuscrit ont permis de faire progresser les performances des systèmes de SRL de collection, il reste encore des axes d'amélioration. Par exemple, nous avons posé la question de l'adaptation non supervisée des modèles de compensation de variabilité, mais la question pourrait se poser de vouloir construire un système de SRL de collection complètement non supervisé, n'utilisant pas de données annotées pour l'apprentissage. Imaginons devoir effectuer la SRL sur des enregistrements de réunions en langue estonienne, peut-être serait-il plus pertinent d'apprendre un système non supervisé de zéro sur le domaine acoustique cible plutôt que de chercher à adapter des modèles appris sur des données radiophoniques en langue française. Au cours de la thèse, nous avons construit un tel système non supervisé dans le cadre du challenge MGB. Les résultats sont prometteurs mais nécessitent d'être généralisés.

Ensuite, l'utilisation d'un réseau de neurones exploitant les *i-vectors* ayant donné de bons résultats, on pourrait poser la question de la remise en cause de la représentation *i-vector* en essayant de nouvelles formes de représentation issues du deep-learning. Ces dernières années, plusieurs méthodes basées sur des réseaux de neurones et visant à remplacer des briques de l'architecture MFCC/UBM/TV ont été proposées en reconnaissance du locuteur : on peut citer les *Time Delay Neural*

Networks [Snyder et al., 2015], qui remplacent le GMM/UBM, où les *Bottleneck Features* [Richardson et al., 2015], visant à remplacer les MFCCs.

En outre, dans nos travaux, la question de la temporalité des documents n'a pas été exploitée pour le regroupement. Par exemple, après un tour de parole t d'un locuteur dans un document, il est peut-être plus probable au tour $t + 2$ que ce soit le même locuteur qui parle, plutôt qu'un locuteur complètement nouveau, surtout si c'est sur une durée très courte. L'approche pourrait être complétée par l'exploitation de l'information des co-occurrences des locuteurs : si deux locuteurs ont parlé dans un même document par le passé, et si on retrouve l'un des deux dans un nouveau document, il est peut-être plus probable d'y retrouver le deuxième.

Enfin, la question de la mise en production d'un outil de SRL de collection se pose également : quelle interface utilisateur ? Comment gérer les erreurs ? Comment calibrer le système ? Par exemple, il est peut-être plus facile pour un utilisateur que plusieurs labels correspondent à un même locuteur plutôt que plusieurs locuteurs soient étiquetés par un même label. La question du coût de la correction des erreurs d'un système de SRL est par exemple un problème à part entière, et on ne sait pas si le seuil de regroupement donnant le résultat le plus facile à corriger est le seuil de regroupement minimisant les taux d'erreur.

Annexe A

Paramètres du système *baseline*

A.1 Modélisation de la Variabilité Totale

Ici, nous évaluons les performances du système de SRL *baseline* en utilisant seulement la Variabilité Totale, c'est-à-dire la modélisation *i-vector* avec la similarité cosine seule. La dimension de l'UBM est fixée à 256 ou 512, selon de la dimension de la matrice de TV testée. Le système de SRL *baseline* est évalué pour des matrices TV de dimension allant de 50 à 400. On teste le système sur les deux collections cibles complètes, en comparant les méthodes de regroupement HAC et CC (regroupement global).

Les résultats sont présentés dans la table A.1. Pour chaque expérience, on indique trois lignes de résultats (les DER intra- et inter-document pour les deux collections), à trois paires de seuils de regroupement (λ_I, λ_X) : une paire qui optimise les performances sur LCP, une paire dédiée à BFM, et une paire qui optimise les performances communes. Les deux dernières colonnes indiquent les performances moyennes entre les deux collections.

Les meilleures performances intra- et inter-document sont mises en valeur pour chaque type de regroupement. Les résultats montrent que la configuration $(256, 50)$ donne en moyenne les meilleurs résultats avec un DER inter-document moyen de 20.2% pour le regroupement HAC et de 17.9% pour le regroupement CC. De manière générale, on peut constater que le regroupement CC est plus performant que le regroupement HAC, sur la collection BFM. L'écart est moins flagrant quand on regarde la collection LCP. Le meilleur DER inter-document en regroupement CC sur LCP est de 19.2% et de 14.8% sur BFM (respectivement 19.5% et 19.0% pour le HAC). Notons enfin que, d'après la dernière colonne, l'écart des performances entre le meilleur système et le pire est de 1.5 point pour le regroupement HAC, et de 3.1 points pour le regroupement CC.

dim. UBM	dim. TV	rgp.	λ_I	λ_X	LCP		BFM		*Moyenne*	
					DER_I	DER_X	DER_I	DER_X	DER_I	DER_X
512	400	HAC	20	20	8.5	20.8	12.2	21.8		
			-10	10	9.5	24.0	13.5	**19.0**		
			20	20	8.5	20.8	14.1	21.8	11.3	21.3
		CC	20	0	8.6	20.4	12.5	17.2		
			0	0	10.0	21.7	12.2	16.3		
			20	0	8.6	20.4	12.5	17.2	10.6	18.8
256	200	HAC	-10	0	8.5	**19.5**	13.6	23.8		
			-40	-20	10.1	24.4	14.4	22.2		
			-10	0	8.5	**19.5**	13.6	23.8	11.1	21.7
		CC	-10	-30	8.5	22.6	12.4	19.3		
			-10	-30	8.5	22.6	12.4	19.3		
			-10	-30	8.5	22.6	12.4	19.3	10.5	21.0
256	100	HAC	-50	0	9.0	19.9	13.6	23.2		
			-50	-20	9.5	21.4	13.3	19.5		
			-50	-20	9.5	21.4	13.3	19.5	11.4	20.5
		CC	-70	-50	9.5	20.7	14.1	20.4		
			-40	-30	11.1	25.3	10.5	**14.8**		
			-40	-40	9.7	23.8	10.7	16.4	10.8	20.1
256	50	HAC	-40	-30	8.5	21.1	12.9	19.2		
			-40	-30	8.5	21.1	12.9	19.2		
			-40	-30	8.5	21.1	12.9	19.2	**10.7**	**20.2**
		CC	-50	-70	9.0	**19.2**	13.3	18.2		
			-40	-50	8.4	20.1	10.8	15.6		
			-40	-50	8.4	20.1	10.8	15.6	**9.6**	**17.9**

TABLE A.1: Influence de la dimension de l'UBM et de la matrice TV sur les performances de SRL de collection appliquée aux collections LCP et BFM, avec similarité cosine et regroupement HAC ou CC.

A.2 Apport de la WCCN

L'expérience précédente est maintenant rejouée avec ajout de la compensation de variabilité intra-locuteur/inter-document WCCN pour le calcul des similarités, toutes choses égales par ailleurs. Les résultats sont présentés dans la table A.2.

Cette fois, on constate que les performances sont très proches entre toutes les configurations, hormis la configuration $(256, 50)$. Les trois configurations $(512, 400)$, $(256, 200)$ et $(256, 100)$ donnent un DER-X moyen aux alentours de 20.1% pour le HAC et 19.3% pour le CC.

Si l'on s'intéresse aux performances dédiées, on constate que la WCCN améliore les performances sur le corpus BFM, avec une baisse de 14.8% à 14.1% pour la meilleure configuration CC et une basse de 19.0% à 17.6% pour la meilleure configuration HAC, mais à des dimensions différents selon la méthode de regroupement. En revanche, on constate sur LCP une très légère amélioration de 19.5% à 19.4% pour le HAC et une dégradation de 19.2% à 21.3% pour le CC. L'évolution peut surprendre étant donné que la WCCN est censée compenser la variabilité intra-locuteur/inter-document, mais la différence d'environnement acoustique entre les données d'apprentissage (radio) et la collection LCP (télévision) est peut-être trop grande pour que la WCCN soit efficace.

dim. UBM	dim. TV	rgp.	λ_I	λ_X	LCP		BFM		*Moyenne*	
					DER_I	DER_X	DER_I	DER_X	DER_I	DER_X
512	400	HAC	-30	-20	9.3	20.0	12.5	20.2		
			-50	-40	9.8	23.0	13.3	**17.6**		
			-30	-20	9.3	20.0	12.5	20.2	**10.9**	20.1
		CC	-30	-40	9.1	23.2	14.5	19.0		
			-50	-40	9.5	23.7	11.6	15.7		
			-40	-40	9.4	23.5	11.7	15.9	10.6	19.7
256	200	HAC	-30	-30	9.6	19.7	11.9	24.2		
			-50	-40	9.8	23.0	13.3	**17.6**		
			-50	-30	9.9	20.6	13.0	19.0	11.5	**19.8**
		CC	-40	-50	8.9	22.6	11.6	15.8		
			-30	-50	14.0	27.5	10.8	15.0		
			-50	-50	9.3	22.9	11.2	15.4	10.3	19.2
256	100	HAC	-50	-40	9.0	**19.4**	12.8	24.5		
			-60	-50	8.8	20.7	13.5	19.8		
			-60	-50	8.8	20.7	13.5	19.8	11.2	20.3
		CC	-60	-70	9.1	**21.3**	12.2	18.3		
			-70	-60	9.6	23.6	10.4	**14.1**		
			-70	-60	9.6	23.6	10.4	**14.1**	**10.0**	**18.9**
256	50	HAC	-70	-70	9.4	21.8	12.8	27.1		
			-60	-60	10.3	23.2	13.3	21.7		
			-60	-60	10.3	23.2	13.3	21.7	11.8	22.5
		CC	-70	-70	15.2	29.4	12.1	18.9		
			-70	-70	15.2	29.4	12.1	18.9		
			-70	-70	15.2	29.4	12.1	18.9	13.7	24.2

TABLE A.2: Influence de la dimension de l'UBM et de la matrice TV sur les performances de SRL de collection appliquée aux collections LCP et BFM, avec similarité cosine+WCCN et regroupement HAC ou CC.

A.3 Performances de la PLDA

Dorénavant, nous fixons la dimension de l'UBM à 256 et travaillons avec des matrices TV de dimensions 100 ou 200. Nous nous intéressons à l'influence de la dimension de la PLDA sur les performances. Les *i-vectors* sont normalisés par la normalisation sphérique.

Encore une fois, comme on peut le voir dans la table A.3, les performances sont assez similaires entre les dimensions, avec un DER inter-document optimal moyen de 17.4% pour la configuration $(200, 100, HAC)$ et de 16.5% pour la configuration $(200, 150, CC)$. Ces valeurs indiquent que la PLDA apporte un gain de compensation de variabilité inter-document.

Du côté des performances dédiées, on constate également une amélioration par rapport à la cosine seule ou avec WCCN, avec un gain par rapport à la WCCN de 14.1% à 13.0% pour la meilleure configuration (BFM, CC) et de 17.6% à 15.7% pour (BFM, HAC). Sur la collection LCP, on passe de 21.3% à 18.8% pour le CC et de 19.4% à 18.0% pour le HAC.

Pour les expériences présentées dans le manuscrit, nous fixerons le système *baseline* avec les paramètres suivants : GMM/UBM à 256 composantes, matrice TV de dimension 200 et matrice PLDA de dimension 100. Ce choix a été arrêté après les expériences préliminaires qui portaient uniquement sur le regroupement HAC. En effet, la configuration retenue est la plus performante pour le regroupement inter-

document avec les mesures de similarité WCCN et PLDA.

dim. TV	dim. PLDA	rgp.	λ_I	λ_X	LCP		BFM		*Moyenne*	
					DER_I	DER_X	DER_I	DER_X	DER_I	DER_X
200	150	HAC	-10	0	8.3	**18.0**	11.4	20.2		
			0	10	9.2	19.3	10.6	16.2		
			0	10	9.2	19.3	10.6	16.2	**9.9**	17.8
		CC	-10	-30	8.3	**18.8**	10.1	14.5		
			0	-20	14.1	24.3	9.6	**13.0**		
			-10	-20	8.4	19.5	10.0	13.4	**9.2**	**16.5**
	100	HAC	-10	10	8.3	18.1	11.1	21.5		
			10	10	10.0	19.1	10.6	**15.7**		
			10	10	10.0	19.1	10.6	**15.7**	10.3	**17.4**
		CC	-10	-30	8.4	20.1	10.1	15.4		
			0	-10	15.0	24.9	9.6	13.4		
			-10	-20	8.7	21.2	9.9	13.6	9.3	17.4
	50	HAC	0	0	8.9	19.7	10.4	16.9		
			0	0	8.9	19.7	10.4	16.9		
			0	0	8.9	19.7	10.4	16.9	9.7	18.3
		CC	-10	-20	9.1	23.4	11.4	16.6		
			0	-10	13.6	23.6	9.6	14.1		
			0	-10	13.6	23.6	9.6	14.1	11.6	18.9
100	100	HAC	-20	-10	8.6	18.3	12.1	20.6		
			-10	-10	9.2	18.6	11.5	16.6		
			-10	-10	9.2	18.6	11.5	16.6	10.4	17.6
		CC	-20	-30	8.6	20.0	10.1	14.9		
			-20	-20	14.2	25.2	9.8	13.7		
			-20	-30	8.6	20.0	10.1	14.9	9.4	17.5
	50	HAC	-20	0	9.0	18.2	12.5	19.2		
			-20	-10	9.3	22.0	12.5	16.4		
			-20	0	9.0	18.2	12.5	19.2	11.3	18.7
		CC	-20	-20	9.2	19.7	10.5	14.6		
			-10	-20	8.5	20.1	9.9	14.3		
			-20	-20	9.2	19.7	10.5	14.6	9.9	17.2

TABLE A.3: Influence de la dimension de la PLDA et de la matrice TV sur les performances de SRL de collection appliquée aux collections LCP et BFM, avec similarité PLDA et regroupement HAC ou CC. La dimension de l'UBM est fixée à 256.

Annexe B

Sensibilité au seuil des systèmes de SRL

Les figures de cette annexe complètent la discussion de la section 4.3.5. Elles représentent l'évolution des taux d'erreur classe en fonction du seuil de regroupement, pour les systèmes *baseline* HAC/WCCN, CC/WCCN, HAC/PLDA et CC/PLDA, à regroupement global. Le DER est aussi illustré par la courbe mauve. Pour chaque système de SRL on observe clairement des optima différents pour chaque type de locuteur. Par définition, les taux d'erreur sur les locuteurs ponctuels sont bas pour des seuils bas et ont tendance à augmenter à la faveur des regroupements. Concernant les locuteurs récurrents, la dynamique est différente, les taux d'erreur au seuil le plus bas ($\lambda_X = -90$) sont élevés, ce qui est normal car aucun regroupement inter-document n'a eu lieu. Ensuite, les regroupements successifs font baisser les taux pour atteindre un minimum, avant de remonter. Chaque figure illustre des dynamiques de taux d'erreur différentes selon les types de locuteurs. Idéalement, un regroupement qui optimise le taux d'erreur des quatre classes pour un même seuil favorisera un DER bas : le système CC/TR, dont nous avons discuté à la section 5.5.2, approche ce cas de figure.

B.1 Système HAC/WCCN

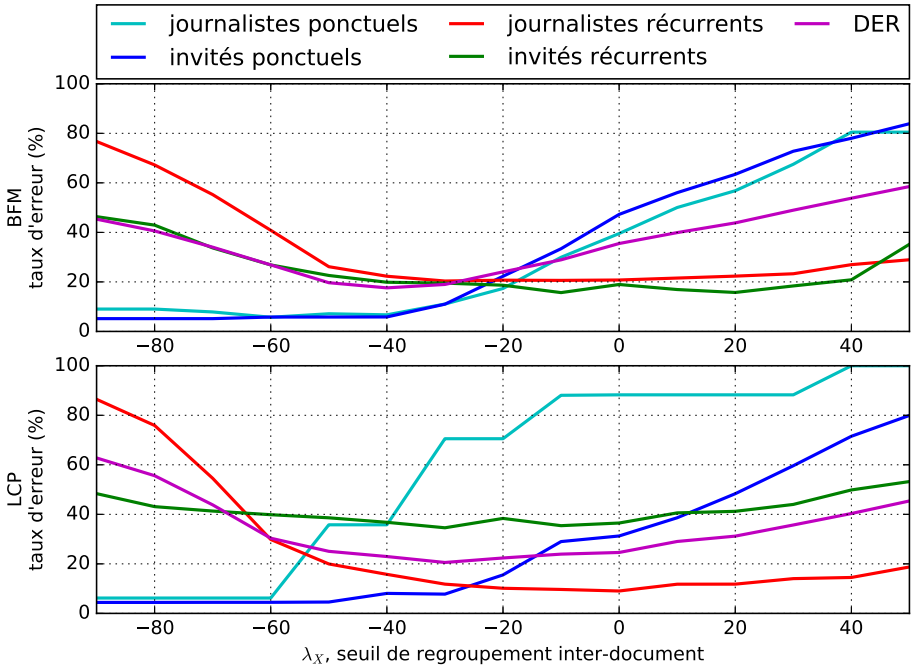

FIGURE B.1 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement HAC λ_X, avec la similarité cosine/WCCN.

B.2 Système CC/WCCN

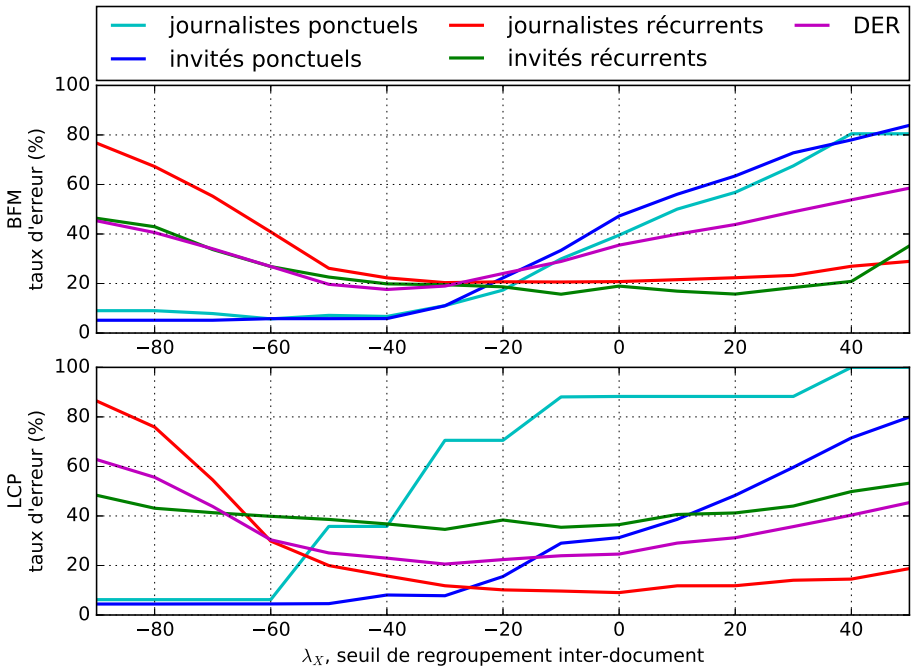

FIGURE B.2 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement CC λ_X, avec la similarité cosine/WCCN.

B.3 Système HAC/PLDA

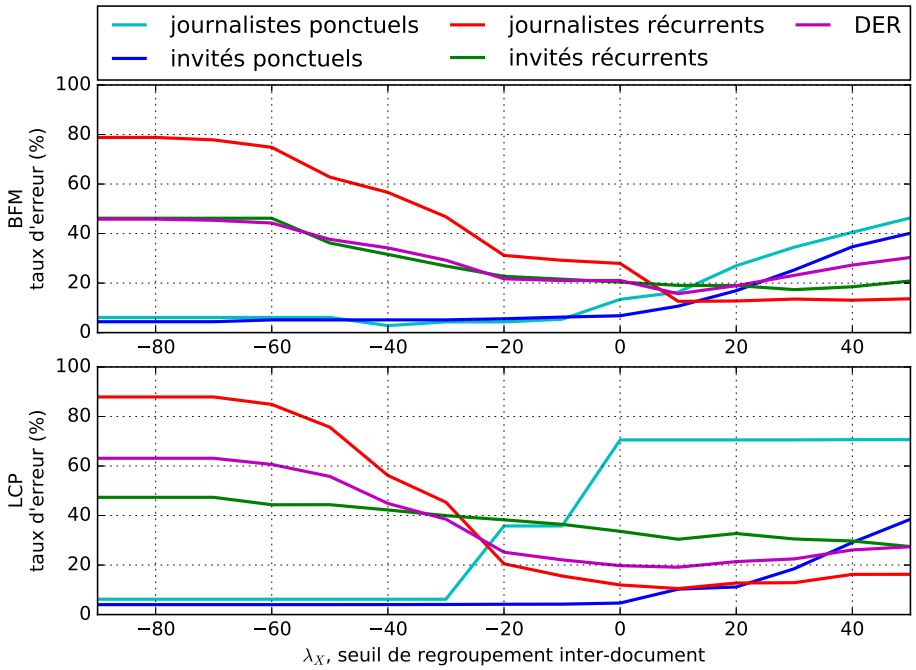

FIGURE B.3 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement HAC λ_X, avec la similarité PLDA.

B.4 Système CC/PLDA

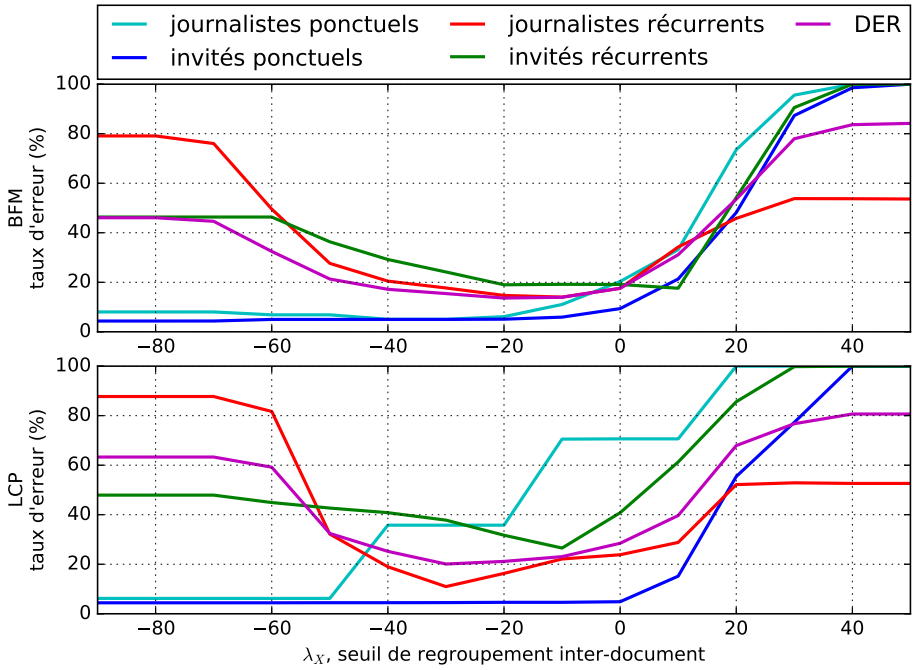

FIGURE B.4 – Evolution des taux d'erreur par type de locuteur, en fonction du seuil du seuil de regroupement CC λ_X, avec la similarité PLDA.

Annexe C

Adaptation du réseau de neurones

Concernant l'adaptation du réseau de neurones pour le calcul de scores TR, pour reprendre le protocole d'adaptation utilisé pour la WCCN ou la PLDA, nous proposons de poursuivre l'apprentissage du réseau *baseline* avec les classes-locuteurs générées par la SRL *baseline*. De la même façon que pour la WCCN et la PLDA, nous introduisons un paramètre d'adaptation α qui permet de pondérer l'influence des classes sources et cibles dans le calcul de la loss.

$$\mathcal{L}(\mathcal{T}) = \alpha \sum_{i}^{N_{cible}} max(0, \Delta_i + \beta) + (1 - \alpha) \sum_{i}^{N_{source}} max(0, \Delta_i + \beta) \qquad \text{(C.1)}$$

Les paramètres à étudier pour l'adaptation sont le nombre d'époques à réaliser ainsi que le choix de l'époque initiale : vaut-il mieux adapter un réseau pré-entraîné ou entraîner un nouveau réseau de zéro, sachant que l'initialisation est aléatoire ?

C.1 Poursuite de l'apprentissage

Une première expérience consiste à figer un réseau *baseline* (ie. appris sur les données sources) après un certain nombre d'époques d'apprentissage, l'utiliser pour réaliser la SRL *baseline*, puis exploiter les classes-locuteurs générées par cette première passe de SRL pour adapter le réseau. Dans ce cas, l'adaptation consiste à poursuivre l'apprentissage avec un mélange de données sources et cibles, la pondération s'effectuant sur le calcul de la *loss*. La figure C.1 (respectivement C.2) présente l'effet de l'adaptation en fonction du nombre d'époques cumulées (époques *baseline* + époques d'adaptation), avec un coefficient d'adaptation $\alpha = 0.5$ (resp. $\alpha = 0.9$). L'adaptation est réalisée à partir de l'époque 0 ou 1500, pour un nombre d'époques cumulées allant jusqu'à 2000. Remarquons que réaliser l'adaptation à partir de l'époque 0 consiste simplement à réaliser un apprentissage conjoint complet à partir de la même initialisation que la *baseline*. Les courbes présentées sont des courbes moyennes : réalisées à partir de 20 réseaux *baseline* différents. Les bornes

(min, max) ne sont représentées que pour l'adaptation à partir de l'époque 1500, pour des raisons de lisibilité.

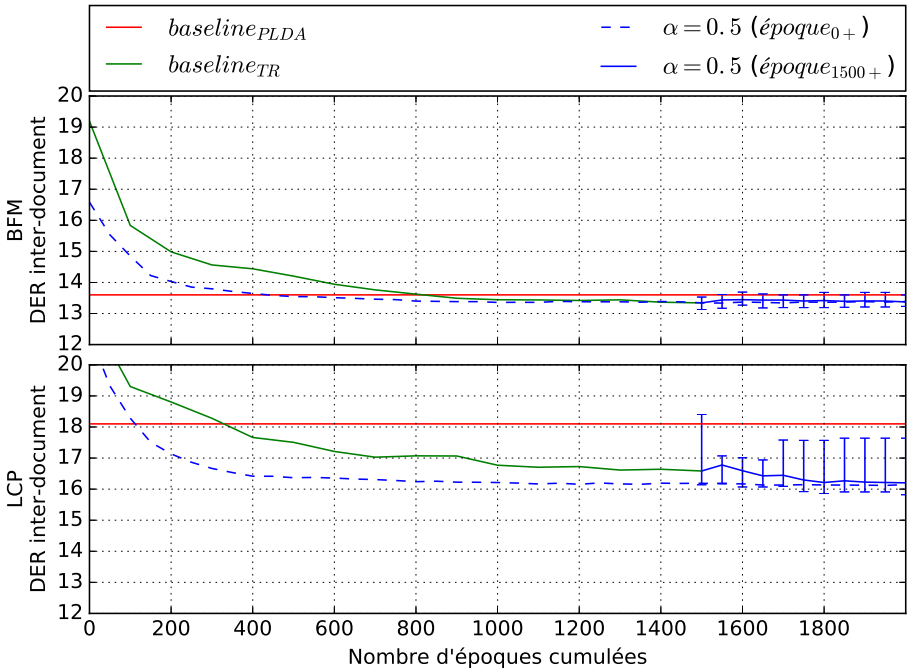

FIGURE C.1 – Analyse de l'évolution du DER en fonction du nombre d'époques cumulées (époques *baseline* + époques d'adaptation), pour $\alpha = 0.5$.

Sur la figure C.1, pour $\alpha = 0.5$, on ne constate pas de différence significative entre les deux stratégies d'adaptation. Les deux courbes bleues semblent converger vers le même point. Pour BFM, l'adaptation ne semble apporter aucune amélioration, tandis que sur la collection LCP, on observe un gain par rapport à la *baseline*, où l'on passe de 16.6% à 16.1%.

La figure C.2 est plus informative, elle montre une différence de comportement lorsqu'on donne plus d'importance aux données cibles dans le calcul de la *loss* : en adaptant à partir de l'époque 0, le DER converge en environ 400 époques vers son minimum, qui pour BFM est 0.5 point au dessus de sa *baseline* et pour LCP au même niveau. Au-delà, les performances de dégradent légèrement. L'adaptation à partir de l'époque 1500 donne des résultats plus intéressants. Si pour BFM on n'observe pas d'amélioration significative (ni de dégradation), les performances sur la collection LCP atteignent les 16.0% en moyenne à l'époque 1750. Phénomène intéressant, l'intervalle (min, max) s'élargit à mesure qu'on approche des 2000 époques, ce qui montre que le DER peut parfois passer à 15.5%... ou remonter.

Pour résumer, l'approche qui consiste à adapter le réseau *baseline* déjà appris sur les données cibles seules semble légèrement meilleure qu'un apprentissage conjoint de zéro. Au vu des performances, nous proposons de fixer le nombre d'époques cumulées à 1750 pour les expériences d'adaptation itératives, c'est-à-dire 250 époques d'adaptation d'un réseau *baseline* figé à 1500 époques.

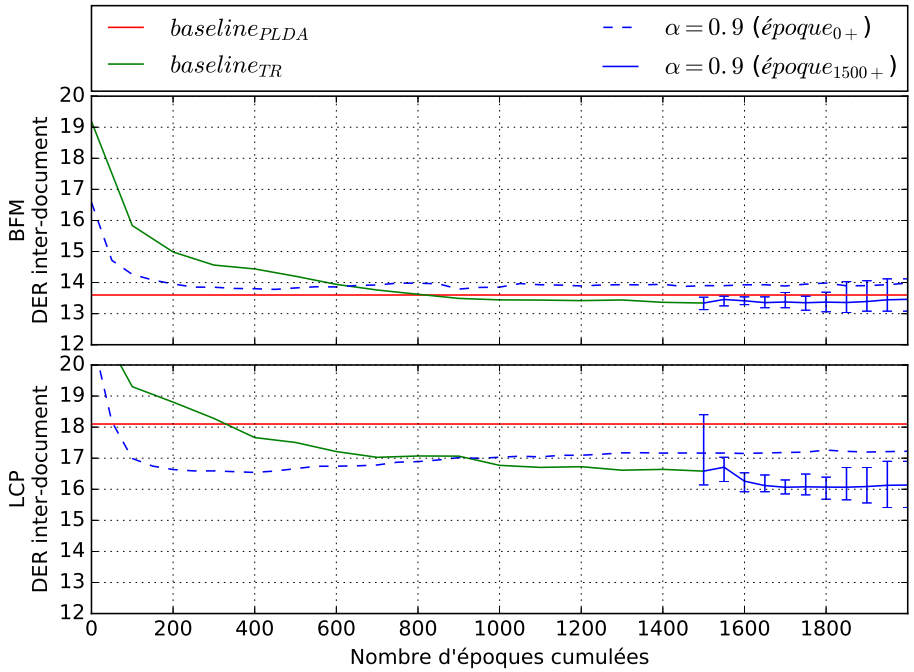

FIGURE C.2 – Analyse de l'évolution du DER en fonction du nombre d'époques cumulées (époques *baseline* + époques d'adaptation), pour $\alpha = 0.9$.

C.2 A propos de la calibration

Lors des expériences, nous avons observé un phénomène qui mérite quelques commentaires. Sur la figure C.3, nous avons représenté l'expérience d'adaptation avec $\alpha = 0.9$ d'une autre perspective : soit le seuil d'adaptation λ_X (ie. le seuil de regroupement CC donnant la performance *baseline*), on s'autorise à utiliser un autre seuil de regroupement $\lambda_{X'}$, tel que $\lambda_{X'} = \lambda_X + 20$, après adaptation.

Sur la figure, lorsqu'on regarde la courbe bleue pleine (adaptation à partir de 1500 époques), on observe des gains moyens significatifs par rapport à la courbe bleue en pointillés, qui représente les performances quand le seuil de regroupement après adaptation est le même que celui utilisé pour réaliser l'adaptation. Le DER après adaptation, à ce nouveau seuil $\lambda_{X'}$, atteint les 12.9% sur BFM et 14.5% sur LCP, en moyenne, avec des gains respectifs de 0.5 et 1.5 points par rapport à la configuration d'adaptation/regroupement initiale.

Intuitivement, cela revient à s'autoriser à "regrouper plus" après adaptation, car les mesures de similarités sont plus précises. Ainsi, là où des regroupements supplémentaires faisaient augmenter le DER sur le système *baseline* ("mauvais regroupements"), ils sont censés être de "bons regroupements" après adaptation. L'inconvénient de la méthode, c'est la variabilité des performances. Comme le montre la figure, si en moyenne, on améliore significativement les performances, dans le pire des cas l'adaptation peut dégrader le DER jusqu'à 3 points pour BFM, et ne rien améliorer du tout pour LCP.

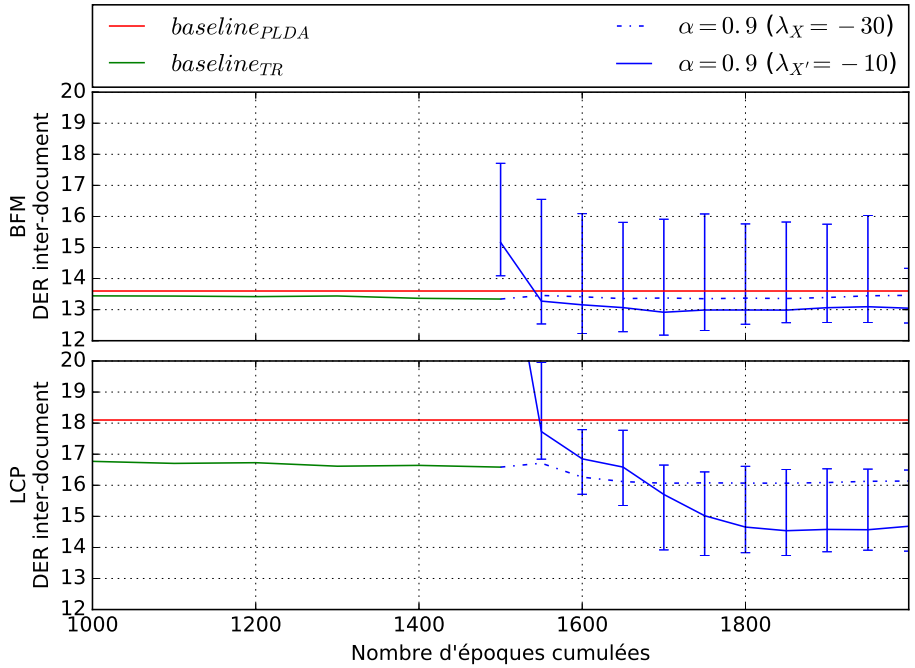

FIGURE C.3 – Analyse de l'évolution du DER en fonction du nombre d'époques cumulées (époques *baseline* + époques d'adaptation), pour $\alpha = 0.9$, à un point de fonctionnement différent de celui d'adaptation ($\lambda_{X'} = \lambda_X + 20$).

Liste des figures

Liste des tableaux

Références bibliographiques

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. (2012). Speaker diarization : A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2) :356–370.

Aronowitz, H. (2014). Inter dataset variability compensation for speaker recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pages 4002–4006. IEEE.

Barras, C., Zhu, X., Meignier, S., and Gauvain, J. (2006). Multi-stage speaker diarization of broadcast news. IEEE Transactions on Speech and Audio Processing, 14(5) :1505–1512.

Bell, P., Gales, M., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., and Woodland, P. (2015). The mgb challenge : Evaluating multi-genre broadcast media transcription. In IEEE ASRU.

Ben, M., Betser, M., Bimbot, F., and Gravier, G. (2004). Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted gmms. In Proc. ICSLP, volume 2004.

Bhattacharya, G., Alam, M. J., Kenny, P., and Gupta, V. (2016). Modelling speaker and channel variability using deep neural networks for robust speaker verification. In 2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16, 2016, pages 192–198.

Bonastre, J.-F., Delacourt, P., Fredouille, C., Merlin, T., and Wellekens, C. (2000). A speaker tracking system based on speaker turn detection for nist evaluation. In Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, volume 2, pages II1177–II1180. IEEE.

Bourlard, H., Ferras, M., Pappas, N., Popescu-Belis, A., Renals, S., McInnes, F., Bell, P., and Guillemot, M. (2013). Processing and Linking Audio Events in Large Multimedia Archives : The EU inEvent Project. In Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM), Marseille, France.

Bousquet, P.-M., Larcher, A., Matrouf, D., Bonastre, J.-F., and Plchot, O. (2012). Variance-spectra Based Normalization for i-vector Standard and Probabilistic Linear Discriminant Analysis. In Odyssey : The Speaker and Language Recognition Workshop, Singapore, Singapore, pages 157–164.

Bousquet, P.-M., Matrouf, D., and Bonastre, J.-F. (2011). Intersession Compensation and Scoring Methods in the I-vectors Space for Speaker Recognition. In Proceedings of Interspeech, Florence, Italia.

Bredin, H. (2016). Tristounet : Triplet loss for speaker turn embedding. arXiv preprint arXiv :1609.04301.

Bredin, H., Poignant, J., Fortier, G., Tapaswi, M., Le, V.-B., Roy, A., Barras, C., Rosset, S., Sarkar, A., Yang, Q., et al. (2013). Qcompere@ repere 2013. In SLAM 2013-First Workshop on Speech, Language and Audio for Multimedia, pages 49–54.

Burget, L., Matejka, P., Schwarz, P., Glembek, O., and Cernocky, J. (2007). Analysis of Feature Extraction and Channel Compensation in a GMM Speaker Recognition System. IEEE Transactions on Audio, Speech, and Language Processing, 15(7) :1979–1986.

Campbell, W. M., Sturim, D. E., and Reynolds, D. A. (2006). Support vector machines using gmm supervectors for speaker verification. IEEE signal processing letters, 13(5) :308–311.

Carey, M. and Parris, E. S. (1992). Speaker Verification Using Connected Words. Proceedings of Institute of Acoustics, 14 :p95–p100.

Chen, L., Lee, K. A., Ma, B., Guo, W., Li, H., and Dai, L. R. (2015). Channel adaptation of plda for text-independent speaker verification. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 5251–5255. IEEE.

Chen, S. and Gopalakrishnan, P. (1998). Speaker, Environment and Channel Change Detection and Clustering Via the Bayesian Information Criterion. In Proc. DARPA Broadcast News Transcription and Understanding Workshop, page 8. Virginia, USA.

Chollet, F. (2017). Keras.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1) :30–42.

Defays, D. (1977). An efficient algorithm for a complete link method. The Computer Journal, 20(4) :364–366.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-End Factor Analysis for Speaker Verification. Audio, Speech, and Language Processing, IEEE Transactions on, 19(4) :788–798.

Delacourt, P., Kryze, D., and Wellekens, C. J. (1999). Detection of Speaker Changes in an Audio Document. In Sixth European Conference on Speech Communication and Technology.

Delgado, F. H., Serrano, G. J., and Anguera, M. X. (2015). Fast cross-session speaker diarization.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological), pages 1–38.

Doddington, G. (2012). The effect of target/non-target age difference on speaker recognition performance. In Odyssey 2012-The Speaker and Language Recognition Workshop.

Dupuy, G., , Meignier, S., and Estève, Y. (2014a). Is incremental cross-show speaker diarization efficient to process large volumes of data ? In Proceedings of Interspeech, Singapore.

Dupuy, G. (2015). Speaker diarization : the voluminous collections of audiovisual recordings. Theses, Université du Maine.

Dupuy, G., Meignier, S., Deléglise, P., and Estève, Y. (2014b). Recent improvements towards ILP-based clustering for broadcast news speaker diarization. In Speaker Odyssey Workshop.

Dupuy, G., Rouvier, M., Meignier, S., and Estève, Y. (2012a). I-vectors and ILP Clustering Adapted to Cross-Show Speaker Diarization. In Proceedings of Interspeech, Portland, Oregon, USA.

Dupuy, G., Rouvier, M., Meignier, S., and Estève, Y. (2012b). Segmentation et Regroupement en Locuteurs d'une collection de documents audio. In Proceedings of 29e Journées d'Études sur la Parole (JEP'12), Grenoble, France.

Favre, B., Damnati, G., Bechet, F., Bendris, M., Charlet, D., Auguste, R., Ayache, S., Bigot, B., Deltei, A., Dufour, R., et al. (2013). Percoli : a person identification system for the 2013 repere challenge. In First Workshop on Speech, Language and Audio in Multimedia.

Ferràs, M. and Bourlard, H. (2012). Speaker Diarization and Linking of Large Corpora. In Proceedings of IEEE Workshop on Spoken Language Technology, Miami, Florida (USA).

Ferras, M., Madikeri, S., and Bourlard, H. (2016a). Speaker diarization and linking of meeting data. IEEE/ACM Transactions on Audio, Speech, and Language Processing, PP(99) :1–1.

Ferras, M., Madikeri, S., Motlicek, P., and Bourlard, H. (2016b). System fusion and speaker linking for longitudinal diarization of tv shows. In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, pages 5495–5499. IEEE.

Furui, S. (1981). Cepstral Analysis Technique for Automatic Speaker Verification. Acoustics, Speech and Signal Processing, IEEE Transactions on, 29(2) :254–272.

Galibert, O. (2013). Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. In INTERSPEECH, pages 1131–1134.

Galibert, O. and Kahn, J. (2013a). The first official repere evaluation. In Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM).

Galibert, O. and Kahn, J. (2013b). The First Official REPERE Evaluation. In Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM), Marseille, France.

Galibert, O., Leixa, J., Gilles, A., Choukri, K., and Gravier, G. (2014). The ETAPE Speech Processing Evaluation. In Conference on Language Resources and Evaluation, Reykyavik, Iceland.

Galliano, S., Gravier, G., and Chaubard, L. (2009). The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. In Proceedings of Interspeech, Brighton, Royaume Uni.

Garcia-Romero, D. and Espy-Wilson, C. Y. (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems. In Interspeech, pages 249–252.

Garcia-Romero, D. and McCree, A. (2014). Supervised domain adaptation for i-vector based speaker recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pages 4047–4051. IEEE.

Garcia-Romero, D., McCree, A., Shum, S., Brummer, N., and Vaquero, C. (2014). Unsupervised domain adaptation for i-vector speaker recognition. In Proceedings of Odyssey : The Speaker and Language Recognition Workshop.

Gauvain, J.-L., Lamel, L., and Adda, G. (1998). Partitioning and transcription of broadcast news data. In ICSLP, volume 98, pages 1335–1338.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. IEEE transactions on Speech and audio processing, 2(2) :291–298.

Ghaemmaghami, H., Dean, D., and Sridha, S. (2013). Speaker Attribution of Australian Broadcast News Data. In Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM), Marseille, France.

Ghaemmaghami, H., Dean, D., and Sridharan, S. (2015). A cluster-voting approach for speaker diarization and linking of australian broadcast news recordings. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4829–4833. IEEE.

Ghaemmaghami, H., Dean, D., Vogt, R., and Sridharan, S. (2012). Speaker attribution of multiple telephone conversations using a complete-linkage clustering approach. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pages 4185–4188. IEEE.

Gish, H., Siu, M.-H., and Rohlicek, R. (1991). Segregation of Speakers for Speech Recognition and Speaker Identification. In IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 873–876. IEEE.

Glembek, O., Ma, J., Matějka, P., Zhang, B., Plchot, O., Bürget, L., and Matsoukas, S. (2014). Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems. In 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 4032–4036. IEEE.

Kanagasundaram, A., Dean, D., and Sridharan, S. (2015). Improving out-domain plda speaker verification using unsupervised inter-dataset variability compensation approach. In Proceedings of 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2015), pages 4654–4658. IEEE.

Karam, Z. N. and Campbell, W. M. (2013). Graph Embedding for Speaker Recognition. In Graph Embedding for Pattern Analysis, pages 229–260. Springer.

Karanasou, P., Gales, M. J., Lanchantin, P., Liu, X., Qian, Y., Wang, L., Woodland, P. C., and Zhang, C. (2015). Speaker diarisation and longitudinal linking in multi-genre broadcast data. In Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on, pages 660–666. IEEE.

Kenny, P. (2010). Bayesian speaker verification with heavy tailed priors. In Speaker Odyssey Workshop.

Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition. IEEE Transactions on Audio, Speech, and Language Processing, 15(4) :1435–1447.

Khoury, E., El Shaffey, L., Ferras, M., and Marcel, S. (2014). Hierarchical speaker clustering methods for the nist i-vector challenge. In Speaker Odyssey Workshop.

Kuhn, R., Junqua, J.-C., Nguyen, P., and Niedzielski, N. (2000). Rapid speaker adaptation in eigenvoice space. IEEE Transactions on Speech and Audio Processing, 8(6) :695–707.

Kuhn, R., Nguyen, P., Junqua, J.-C., Goldwasser, L., Niedzielski, N., Fincke, S., Field, K., and Contolini, M. (1998). Eigenvoices for Speaker Adaptation. In ICSLP, volume 98, pages 1774–1777.

Lance, G. N. and Williams, W. T. (1967). A general theory of classificatory sorting strategies i. hierarchical systems. The computer journal, 9(3) :373–380.

Larcher, A., Aik Lee, K., and Meignier, S. (2016). An extensible speaker identification sidekit in python. In International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE.

Le, V. B., Mella, O., and Fohr, D. (2007). Speaker Diarization Using Normalized Cross Likelihood Ratio. In Proceedings of Interspeech'07, volume 7, pages 1869–1872.

Le Lan, G., Charlet, D., Larcher, A., and Meignier, S. (2016a). Autoapprentissage pour le regroupement en locuteurs : premières investigations. In Journées d'Études sur la Parole (JEP'16), pages 80–82. AFCP.

Le Lan, G., Charlet, D., Larcher, A., and Meignier, S. (2016b). First investigations on self trained speaker diarization. In Proceedings of Odyssey : The Speaker and Language Recognition Workshop.

Le Lan, G., Charlet, D., Larcher, A., and Meignier, S. (2016c). Iterative plda adaptation for speaker diarization. In Interspeech, pages 2175–2179.

Le Lan, G., Charlet, D., Larcher, A., and Meignier, S. (2017). A triplet ranking-based neural network for speaker diarization and linking. In Interspeech.

Le Lan, G., Meignier, S., Charlet, D., and Deléglise, P. (2016d). Speaker diarization with unsupervised training framework. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5560–5564.

Matrouf, D., Scheffer, N., Fauve, B. G., and Bonastre, J.-F. (2007). A Straightforward and Efficient Implementation of the Factor Analysis Model for Speaker Verification. In Proceedings of Interspeech'07.

Matveev, Y. (2013). The problem of voice template aging in speaker recognition systems. In International Conference on Speech and Computer, pages 345–353. Springer.

Meignier, S., Bonastre, J.-F., Fredouille, C., and Merlin, T. (2000). Evolutive hmm for multi-speaker tracking system. In Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, volume 2, pages II1201–II1204. IEEE.

Meignier, S., Bonastre, J.-F., and Magrin-Chagnolleau, I. (2002). Speaker utterances tying among speaker segmented audio documents using hierarchical classification : towards speaker indexing of audio databases. In Seventh International Conference on Spoken Language Processing.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. Pattern recognition and artificial intelligence, 116 :374–388.

NIST (2000). The 2000 NIST Speaker Recognition Evaluation Plan.

NIST (2003). The NIST Rich Transcription Spring 2003 (RT-03S) Evaluation Plan.

Prince, S. J. and Elder, J. H. (2007). Probabilistic Linear Discriminant Analysis for Inferences About Identity. In IEEE 11th International Conference on Computer Vision, pages 1–8. IEEE.

Rabiner, L. R. and Juang, B.-H. (1993). Fundamentals of Speech Recognition, volume 14. PTR Prentice Hall Englewood Cliffs.

Reynolds, D., Singer, E., Carlson, B., O'Leary, G., Mc Laughlin, J., and Zissman, M. (1998). Blind clustering of speech utterances based on speaker and language characteristics. In Proceedings of International Conference on Spoken Language Processing, Sydney, Australia.

Reynolds, D. A. (1992). A Gaussian Mixture Modeling Approach to Text-independent Speaker Identification. In Thèse de doctorat. Georgia Institute of Technology.

Reynolds, D. A. (1995). Speaker identification and verification using gaussian mixture speaker models. Speech communication, 17(1) :91–108.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. Digital signal processing, 10(1) :19–41.

Richardson, F., Reynolds, D., and Dehak, N. (2015). A unified deep neural network for speaker and language recognition. arXiv preprint arXiv :1504.00923.

Rose, R. C. and Reynolds, D. A. (1990). Text Independent Speaker Identification Using Automatic Acoustic Segmentation. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 293–296. IEEE.

Rouvier, M., Dupuy, G., Gay, P., Khoury, E., Merlin, T., and Meignier, S. (2013). An open-source state-of-the-art toolbox for broadcast news diarization. Technical report, Idiap.

Rouvier, M. and Meignier, S. (2012). A Global Optimization Framework for Speaker Diarization. In Proceedings of Odyssey 2014 : The Speaker and Language Recognition Workshop, Singapore.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet : A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823.

Schwarz, G. et al. (1978). Estimating the Dimension of a Model. The annals of statistics, 6(2) :461–464.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D., and Glass, J. (2011a). Exploiting Intra-Conversation Variability for Speaker Diarization. In Proceedings of Interspeech, Florence, Italy.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D. A., and Glass, J. R. (2011b). Exploiting intra-conversation variability for speaker diarization. In interspeech, volume 11, pages 945–948.

Shum, S. H., Campbell, W. M., and Reynolds, D. A. (2013a). Large-scale Community Detection on Speaker Content Graphs. In International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7716–7720. IEEE.

Shum, S. H., Dehak, N., Dehak, R., and Glass, J. R. (2013b). Unsupervised methods for speaker diarization : An integrated and iterative approach. IEEE Transactions on Audio, Speech, and Language Processing, 21(10) :2015–2028.

Shum, S. H., Reynolds, D. A., Garcia-romero, D., and Mccree, A. (2014a). Unsupervised clustering approaches for domain adaptation in speaker recognition systems.

Shum, S. H., Reynolds, D. A., Garcia-Romero, D., and McCree, A. (2014b). Unsupervised clustering approaches for domain adaptation in speaker recognition systems.

Sibson, R. (1973). Slink : an optimally efficient algorithm for the single-link cluster method. The computer journal, 16(1) :30–34.

Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic Segmentation, Classification and Clustering of Broadcast News Audio. In Proceedings of the DARPA Broadcast News Workshop, page 11.

Silovsky, J. and Prazak, J. (2012). Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pages 4193–4196. IEEE.

Siu, M.-H., Yu, G., and Gish, H. (1992). An Unsupervised, Sequential Learning Algorithm for the Segmentation of Speech Waveforms with Multiple Speakers. In IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 189–192. IEEE.

Snyder, D., Garcia-Romero, D., and Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition. In Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on, pages 92–97. IEEE.

Solomonoff, A., Mielke, A., Schmidt, M., and Gish, H. (1998). Clustering Speakers by Their Voices. In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, volume 2, pages 757–760. IEEE.

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. The Journal of the Acoustical Society of America, 8(3) :185–190.

Tran, V.-A., Le, V. B., Barras, C., and Lamel, L. (2011a). Comparing multi-stage approaches for cross-show speaker diarization. In INTERSPEECH, number 1, pages 1053–1056.

Tran, V.-A., Le, V. B., Barras, C., and Lamel, L. (2011b). Comparing Multi-Stage Approaches for Cross-Show Speaker Diarization. In Proceedings of Interspeech, Florence, Italy.

Tranter, S. E. and Reynolds, D. A. (2006). An overview of automatic speaker diarization systems. IEEE Transactions on audio, speech, and language processing, 14(5) :1557–1565.

Valente, F. and Wellekens, C. (2004). Variational bayesian speaker clustering. In ODYSSEY04-The Speaker and Language Recognition Workshop.

Van Leeuwen, D. A. (Proc. Odyssey 2010). Speaker linking in large data sets.

Van Leeuwen, D. A. and Brümmer, N. (2007). An introduction to application-independent evaluation of speaker recognition systems. In Speaker classification I, pages 330–353. Springer.

Villalba, J. and Lleida, E. (2014). Unsupervised adaptation of plda by using variational bayes methods. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pages 744–748. IEEE.

Villalba, J., Ortega, A., Miguel, A., and Lleida, E. (2015). Variational bayesian plda for speaker diarization in the mgb challenge. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pages 667–674.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1386–1393.

Ward Jr, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. Journal of the American statistical association, 58(301) :236–244.

Yang, Q., Jin, Q., and Schultz, T. (2011a). Investigation of cross-show speaker diarization. In INTERSPEECH, pages 2925–2928.

Yang, Q., Jin, Q., and Schultz, T. (2011b). Investigation of Cross-show Speaker Diarization. In Proceedings of Interspeech, Florence, Italy.

Zeiler, M. D. (2012). ADADELTA : an adaptive learning rate method. CoRR, abs/1212.5701.

Liste des contributions

- **Conférences d'audience internationale avec comité de relecture**

Le Lan, G., Charlet, D., Larcher, A., & Meignier, S. (2017, August). *A triplet ranking-based neural network for speaker diarization and linking*. In Interspeech 2017 (Vol. , pp.).

Le Lan, G., Charlet, D., Larcher, A., & Meignier, S. (2016, September). *Iterative PLDA adaptation for speaker diarization*. In Interspeech 2016 (Vol. 2016, pp. 2175-2179).

Le Lan, G., Meignier, S., Charlet, D., & Deléglise, P. (2016, March). *Speaker diarization with unsupervised training framework*. In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on (pp. 5560-5564). IEEE.

- **Workshops d'audience internationale avec comité de relecture**

Le Lan, G., Meignier, S., Charlet, D., & Larcher, A. (2016, June). *First investigations on self trained speaker diarization*. In Proceedings of Odyssey : The Speaker and Language Recognition Workshop.

- **Conférences d'audience nationale avec comité de relecture**

Le Lan, G., Meignier, S., Charlet, D., & Larcher, A. (2016). *Autoapprentissage pour le regroupement en locuteurs : premières investigations*. In Journées d'Études sur la Parole (JEP'16) (pp. 80-82). AFCP.

UNIVERSITE BRETAGNE LOIRE

orange

Le Mans Université

Thèse de Doctorat

Gaël LE LAN

Analyse en locuteurs de collections de documents multimédia

Speaker analysis of multimedia data collections

Résumé

La segmentation et regroupement en locuteurs (SRL) de collection cherche à répondre à la question « qui parle quand ? » dans une collection de documents multimédia. C'est un prérequis indispensable à l'indexation des contenus audiovisuels. La tâche de SRL consiste d'abord à segmenter chaque document en locuteurs, avant de les regrouper à l'échelle de la collection. Le but est de positionner des labels anonymes identifiant les locuteurs, y compris ceux apparaissant dans plusieurs documents, sans connaître à l'avance ni leur identité ni leur nombre. La difficulté posée par le regroupement en locuteurs à l'échelle d'une collection est le problème de la variabilité intra-locuteur/inter-document : selon les documents, un locuteur peut parler dans des environnements acoustiques variés (en studio, dans la rue…). Cette thèse propose deux méthodes pour pallier le problème. D'une part, une nouvelle méthode de compensation neuronale de variabilité est proposée, utilisant le paradigme de *triplet-loss* pour son apprentissage. D'autre part, un procédé itératif d'adaptation non supervisée au domaine est présenté, exploitant l'information, même imparfaite, que le système acquiert en traitant des données, pour améliorer ses performances sur le domaine acoustique cible. De plus, de nouvelles méthodes d'analyse en locuteurs des résultats de SRL sont étudiées, pour comprendre le fonctionnement réel des systèmes, au-delà du classique taux d'erreur de SRL (*Diarization Error Rate* ou DER). Les systèmes et méthodes sont évalués sur deux émissions télévisées d'une quarantaine d'épisodes, pour les architectures de SRL globale ou incrémentale, à l'aide de la modélisation locuteur à l'état de l'art.

Mots clés
Segmentation et regroupement en locuteurs, réseau de neurones, adaptation au domaine, apprentissage supervisé, apprentissage non supervisé

Abstract

The task of speaker diarization and linking aims at answering the question "who speaks and when?" in a collection of multimedia recordings. It is an essential step to index audiovisual contents. The task of speaker diarization and linking firstly consists in segmenting each recording in terms of speakers, before linking them across the collection. Aim is, to identify each speaker with a unique anonymous label, even for speakers appearing in multiple recordings, without any knowledge of their identity or number. The challenge of the cross-recording linking is the modeling of the within-speaker/across-recording variability: depending on the recording, a same speaker can appear in multiple acoustic conditions (in a studio, in the street…). The thesis proposes two methods to overcome this issue. Firstly, a novel neural variability compensation method is proposed, using the *triplet-loss* paradigm for training. Secondly, an iterative unsupervised domain adaptation process is presented, in which the system exploits the information (even inaccurate) about the data it processes, to enhance its performances on the target acoustic domain. Moreover, novel ways of analyzing the results in terms of speaker are explored, to understand the actual performance of a diarization and linking system, beyond the well-known Diarization Error Rate (DER). Systems and methods are evaluated on two TV shows of about 40 episodes, using either a global, or longitudinal linking architecture, and state of the art speaker modeling (*i-vector*).

Key Words
speaker diarization and linking, neural network, domain adaptation, unsupervised training, supervised training