Segmentation et regroupement en locuteurs dans des documents audiovisuels, en interaction avec des annotateurs humains

Pierre-Alexandre Broux

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Le Mans Université

THESE DE DOCTORAT DE

LE MANS UNIVERSITE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Pierre-Alexandre BROUX

Segmentation et regroupement en locuteurs dans des documents audiovisuels, en interaction avec des annotateurs humains

Thèse présentée et soutenue à Le Mans, le 10/01/2020
Unité de recherche : Institut national de l'audiovisuel (INA) et Laboratoire informatique Le Mans Université (LIUM)
Thèse N° : 2020LEMA1001

Rapporteurs :

Jean-François BONASTRE	Professeur, LIA
Nicholas EVANS	Professeur, EURECOM

Examinateurs :

Régine ANDRÉ-OBRECHT	Professeur, Université Toulouse

Président :

Daniel Luzzati	Professeur émérite, LIUM

Directeur de thèse :

Sylvain MEIGNIER	Professeur, LIUM

Co-encadrants de thèse :

Simon PETITRENAUD	Maître de conférences, LIUM
Jean CARRIVE	Responsable du service de la recherche, INA

Déclaration

Je déclare être pleinement conscient que le plagiat de documents ou d'une partie d'un document publiés sur toutes formes de support, y compris l'internet, constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée. En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour écrire ce mémoire. Je déclare également que ce mémoire est original et n'a pas été soumis dans son entièreté ou en partie pour un autre diplôme ou une autre qualification.

Remerciements

Tout d'abord, je voudrais remercier M. Jean-François Bonastre et M. Nicholas Evans pour avoir accepté d'être rapporteurs ainsi que l'examinatrice Mme Régine André-Obrecht.

Ensuite, je tiens à remercier mes encadrants. Au LIUM, Sylvain Meignier et Simon Petitrenaud pour leurs confiances, leurs soutiens et leurs patiences. Plus particulièrement, Sylvain pour m'avoir poussé à m'améliorer et Simon pour sa disponibilité ainsi que son grand suivi, autant professionnel que personnel. À l'INA, je voudrais remercier Jean Carrive pour son soutien, sa patience, la prise de nouvelles et la surveillance de mon état de santé.

J'aimerais aussi remercier l'équipe du LIUM pour sa chaleureuse et bonne ambiance, que ce soit au Mans ou à l'étranger lors de conférences. Une pensée toute particulière pour Kévin Vythelingum et nos échanges interminables. Je voudrais également remercier Boris Dupin, Vincent Monard et Haolin Ren pour les bons moments passés à l'INA.

Enfin, je voudrais remercier ma famille, ma compagne, mes amis et les personnes du *REM-GYM* à Paris.

Abstract

The diarization task tries to determine the number of speakers as well as their interventions in an audio file. It is an interesting task for any enterprise willing to index its audiovisual contents. Especially, the French National Audiovisual Institute (INA) desires to apply this task on its archives so as to improve its accessibility and its annotation. However, the uses of the institute need a minimal quality which, most of the time, is not reached by the state-of-the-art automatic diarization systems yet. In order to reach the wanted effectiveness, a human can correct the output of a diarization system. Nevertheless, a human intervention is generally time-consuming and expensive. In order to reduce these costs, a possible solution is to use a computer-assisted system: a human gives some information to a system in order that it can improve its predictions so as to decrease its intervention cost.

The present manuscript revolves around the computer-assisted diarization. It proposes a metric so as to assess the human intervention cost to correct a diarization, a framework to evaluate the human corrections of a speaker diarization, an automaton simulating the human corrections to do for a diarization and some computer-assisted diarization systems decreasing the total human intervention cost. More precisely, the proposed computer-assisted diarization systems reassess either only the speaker clustering or the segmentation and the speaker clustering.

Résumé

La tâche de segmentation et de regroupement en locuteur (SRL) consiste à déterminer le nombre de locuteurs ainsi que leurs interventions dans un document audio. Cette tâche intéresse de nombreuses entreprises qui souhaitent indexer leurs contenus audiovisuels. En particulier, l'institut national de l'audiovisuel (INA) désire appliquer cette tâche sur ses archives afin d'en améliorer l'accessibilité mais également l'annotation. Cependant, les usages de l'institut requièrent une qualité minimum qui n'est, la plupart du temps, pas encore atteinte par les systèmes automatiques de SRL à l'état de l'art. Pour atteindre les performances voulues, un humain peut corriger la sortie d'un système de SRL. Néanmoins, une intervention humaine est généralement chronophage et coûteuse. Afin de réduire ces coûts, une solution possible est d'utiliser un système assisté par l'humain : un humain donne des informations à un système afin qu'il améliore ses prédictions pour faire décroître son coût de correction.

Le présent manuscrit s'articule autour de la SRL assistée par l'humain. Il propose une mesure afin d'évaluer le coût d'intervention humain pour corriger une SRL, un protocole pour évaluer les interactions d'un humain pour la SRL, un automate simulant les corrections humaines à faire pour une SRL et des systèmes de SRL assistés réduisant le coût d'intervention humain total. Plus précisément, les systèmes de SRL assistés présentés réévaluent soit uniquement le regroupement en locuteurs, soit la segmentation et le regroupement en locuteurs.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Les missions de l'INA

L'Institut national de l'audiovisuel (INA), créé le 6 janvier 1975, est un établissement public à caractère industriel et commercial (*EPIC*), en charge de la mémoire sonore et visuelle de l'audiovisuel français. Plus précisément, il s'occupe notamment de collecter, de sauvegarder, d'archiver, de documenter, de produire et de valoriser des contenus audiovisuels. L'INA est également un centre de recherche pluridisciplinaire sur l'audiovisuel et les médias.

Collecte, sauvegarde et archivage

L'INA a pour première mission la collecte et la conservation des contenus audiovisuels français :

- les chaînes publiques de radio et de télévision ainsi que les archives professionnelles depuis 1945 ;
- au titre du dépôt légal de la radio et de la télévision françaises, l'ensemble des programmes qu'émettent les diffuseurs nationaux. Limité aux émissions produites par les diffuseurs hertziens avant 1995, puis ensuite étendu aux chaînes du câble et du satellite ainsi qu'aux radios privées ;
- au titre du dépôt légal des sites web français, l'ensemble des sites en lien avec l'audiovisuel français.

Pour la collecte de ces fonds audiovisuels, l'INA a mis en place un processus de captation numérique. En 2017, 8600 comptes de plateformes vidéo, 13 000 comptes Twitter, 14 426 sites web médias, 66 chaînes de radio et 103 chaînes de télévision ont été captés en continu. Toutes les archives de l'INA cumulées représentaient alors une quantité approximative de 18 millions d'heures de radio et de télévision. L'INA conserve en outre d'autres fonds au gré

de contrats passés par l'institut auprès d'autres secteurs que l'audiovisuel (établissements culturels, entreprises privées, etc.).

L'INA assure également la conservation physique de ce patrimoine français. Au fil du temps, les supports de ce patrimoine audiovisuel, souvent uniques, se détériorent inexorablement. Afin de faire face à ce problème, l'INA a lancé un Plan de sauvegarde et de numérisation (*PSN*) en 1999. En 2017, environ 1,6 million d'heures ont été sauvées. L'INA prévoit d'avoir numérisé 90% de ses fonds en 2019 et 100% en 2021.

L'INA ne conserve pas uniquement des contenus audiovisuels français. Pour des raisons économiques ou politiques, l'INA conserve également des fonds étrangers. Par exemple, dans le cadre d'un accord de coopération signé le 20 décembre 2013, l'Afrique du Sud a collaboré avec l'INA afin de sauver les 256 heures d'enregistrements du procès de Nelson Mandela et des leaders du Congrès national africain.

Exploitation des archives

En plus de la collecte et de la conservation de ses fonds audiovisuels, l'INA a pour mission leur valorisation auprès de trois publics :

- le grand public par des interfaces de consultation disponibles en ligne sur le site ina.fr et ses plateformes associées. Cela vise à mieux faire connaître auprès d'internautes curieux ou nostalgiques la richesse des contenus collectés, en particulier les fonds historiques. Les fonds disponibles dans ce cadre sont des fonds libres de tout droit ou pour lesquels l'INA possède des droits d'exploitation. En 2017, ces fonds disponibles s'élevaient à 49 260 heures de programmes ;
- les professionnels qui accèdent à des archives par une interface de consultation en ligne disponible sur le site inamediapro.com. Cette interface est une plateforme de prévisualisation, de sélection ainsi que d'achat en ligne. Elle s'appuie sur des recherches de contenus au moyen de métadonnées produites par les documentalistes de l'institut et sur un dispositif de visionnage des contenus en ligne. En 2017, environ 1,8 million d'heures de programmes étaient accessibles en ligne pour ce type d'usagers ;
- les chercheurs et universitaires qui peuvent consulter l'ensemble des collections dans des espaces d'accueil qui leur sont dédiés à la Bibliothèque nationale de France (BnF) et dans une trentaine d'autres centres de consultation. Dans ces espaces, les chercheurs peuvent visionner ou écouter les documents. L'accès aux collections s'appuie sur la mise à disposition des métadonnées de description des médias, produites par les équipes de l'INA. En 2017, 18 millions d'heures de programmes

radio et télévision étaient disponibles pour ce public à la BnF et dans une trentaine d'autres centres pour une consultation *in situ*.

L'INA valorise d'autres manières ses archives : en produisant et en proposant des programmes d'information courts sur la TNT, le web, les applications mobiles et les réseaux sociaux ; en faisant des partenariats culturels et éducatifs ; en créant de nouvelles œuvres télévisuelles, radiophoniques ou cinématographiques.

À partir de ses collections, l'INA impulse, mène et accompagne de nombreuses actions de valorisation scientifique : débats, colloques, rencontres, publications, etc. Ces initiatives permettent de contribuer à la compréhension et au décryptage de l'univers médiatique contemporain ainsi qu'à l'analyse de l'évolution de l'écosystème audiovisuel et numérique.

Activités de recherche

L'INA possède un service de recherche dont l'activité s'inscrit dans une dynamique de partenariats avec de nombreux acteurs en France et à l'international. Cette activité de recherche sur l'audiovisuel et les médias recouvre un large éventail de thématiques : la restauration, la numérisation, l'analyse des images et des sons, la reconnaissance de formes, la fouille de données, le traitement sémantique des contenus ou encore l'analyse des médias.

Documentation des archives

Afin de pouvoir valoriser ses archives, l'INA doit les documenter. La documentation consiste notamment à associer aux collections des notices documentaires. Les notices documentaires sont des fiches descriptives qui contiennent divers champs tels qu'un titre, un résumé, les noms des intervenants, des références temporelles appelées "*timecodes*" ou encore des mots clefs nommés "*descripteurs*" contrôlés par un thésaurus. Ces champs sont obtenus à partir de différentes opérations telles que la "*segmentation*" et l'identification des locuteurs. L'INA nomme "*segmentation*" l'opération de repérage temporel qui vise à indiquer les temps de début et de fin d'une entité documentaire (reportage, extrait, prestation scénique, etc.). L'identification des locuteurs, quant à elle, est l'opération consistant à indiquer les intervenants associés à une notice documentaire en s'appuyant sur des listes d'autorité. Ces listes d'autorité sont remplies au fur et à mesure par les documentalistes lorsqu'ils découvrent de nouveaux locuteurs. Les notices documentaires permettent de retrouver efficacement des documents audiovisuels pertinents vis-à-vis des demandes des personnes souhaitant accéder aux archives.

Au moment de la rédaction de ce manuscrit, les notices documentaires produites à l'INA sont entièrement réalisées manuellement par des documentalistes professionnels. Certaines

tâches associées à cette annotation, telle que la segmentation temporelle, sont fastidieuses, chronophages et coûteuses. En raison du nombre limité des équipes documentaires et de l'accroissement constant du nombre d'heures stockées, une quantité importante d'archives reste à ce jour non documentée ou sous-documentée. Le niveau de détail des processus de documentation varie en fonction du type d'archives considérées : les émissions d'information sont généralement documentées finement, tandis que les autres (émissions variétés, documentaires, télé-réalités) sont généralement associées à un niveau de détail minimal. Le repérage temporel de nombreuses notices n'est pas disponible, par exemple les journaux télévisés des années 70 et 80 ou pour des contenus récemment collectés dont les composants de la notice sont directement fournis par les chaînes et pour lesquels la segmentation n'a pas encore été réalisée par l'INA.

Le processus d'annotation ayant débuté il y a plus de 40 ans, la méthodologie d'annotation a évolué en fonction des besoins. Par conséquent, les notices ne suivent pas toutes le même guide d'annotation et elles nécessiteraient une mise à jour afin de les uniformiser.

1.2 Les besoins de l'INA en l'annotation des locuteurs

Les attentes sur l'annotation des locuteurs

Les représentations variées des personnes dans les archives (entités nommées, audio, image, etc.) sont importantes dans le contexte de l'INA. En effet, les différents publics interrogeant les archives de l'institut, notamment les professionnels, ont des attentes élevées en ce qui concerne la qualité, l'accessibilité, mais également l'annotation des contenus. Ainsi, lorsqu'ils font une recherche sur une personne donnée, ils aimeraient avoir toutes les archives qui ont un lien avec elle d'une manière précise.

Pour répondre à une partie de ces exigences, l'INA désire avoir une annotation précise des locuteurs. Il souhaite que les opérations d'annotation manuelle de segmentation et d'identification des locuteurs soient les plus précises possibles afin qu'on puisse savoir "*Qui parle quand*" et "*Qui dit quoi*". Cependant, les notices documentaires actuelles ne vont pas à un tel niveau de détail. En effet, arriver à ce niveau rajouterait un temps de travail considérable aux documentalistes. Actuellement, une notice documentaire est associée à un niveau de détail moindre et décrit une scène ou une séquence (par exemple, un sujet de journal télévisé) dans sa globalité. À ce niveau de détail, la liste des intervenants est parfois donnée mais en général sans timecodes.

Une solution envisagée par l'INA est d'automatiser partiellement ou entièrement l'annotation des locuteurs.

Une automatisation répondant aux attentes de l'INA pour l'annotation des locuteurs permettra d'améliorer les usages actuels, voire les élargir. Voici une liste non exhaustive de ces usages :

- effectuer des recherches plus fines. Il sera possible d'avoir les instants de début et de fin de chaque intervention ;
- améliorer la navigation. En ayant toutes les interventions des locuteurs correctement timecodées, on peut facilement accéder aux parties d'une archive. Ne pas les avoir comme c'est le cas actuellement oblige les clients et usagers à parcourir toute l'émission pour retrouver les interventions ;
- effectuer des analyses ou des statistiques plus approfondies. Par exemple, il sera possible d'effectuer un décompte du temps de parole comme le fait le Conseil supérieur de l'audiovisuel (*CSA*) en période électorale ;
- aider les chercheurs. Par exemple, il sera possible de constituer des bases d'apprentissage de modèles de locuteurs ;
- aider les chercheurs en sciences humaines et sociales. Un niveau d'annotation élevé facilitera l'analyse et l'interprétation des documents.

L'automatisation des annotations des locuteurs

Un procédé automatique pour la création ou la mise à jour de notices documentaires pourrait utiliser les outils de transcription automatique de la parole, la segmentation et le regroupement en locuteurs ou encore l'identification des locuteurs.

Alquier *et al.* [2018] qui applique ces outils sur des archives de l'INA, met en évidence que les taux d'erreur associés sont généralement jugés comme acceptables dans beaucoup d'applications, mais pas pour celles de l'INA. En effet, les usages de l'institut requièrent une qualité minimum qui n'est, la plupart du temps, pas encore atteinte par ces tâches automatiques. Par conséquent, elles ne peuvent être intégrées telles quelles dans un processus métier de l'INA. En effet, les archives seraient mal documentées, ce qui ne répondrait pas aux exigences de qualité élevée des usagers.

1.3 Positionnement du problème

Afin de réduire le temps d'annotation en maintenant une qualité suffisante, l'humain peut être sollicité pour corriger les sorties d'un système automatique. De précédents travaux ont montré qu'il était plus rapide de corriger une transcription automatique que de transcrire intégralement le document à la main [Bazillon *et al.*, 2008].

La question à laquelle nous chercherons à répondre dans cette thèse est : "*Est-il possible d'utiliser les corrections de l'annotateur pour améliorer le système automatique ?*". Plus précisément, ce travail vise à proposer des outils d'aide à la documentation à même d'interagir en temps réel avec un annotateur, pour améliorer au maximum la qualité des annotations en un temps réduit. Nous nous intéresserons à la tâche de segmentation et de regroupement en locuteurs qui vise à déterminer le nombre de locuteurs présents dans un signal de parole ainsi que le moment de leurs interventions.

1.4 Plan du manuscrit

Ce manuscrit de thèse est composé de 6 chapitres.

Les chapitres 2 et 3 présentent des états de l'art. Le chapitre 2 se focalise sur la tâche de segmentation et regroupement en locuteurs tandis que le chapitre 3 se concentre sur les systèmes assistés par l'humain.

Les chapitres 4 et 5 exposent les contributions de la thèse. Le chapitre 4 propose un protocole pour évaluer les interactions d'un humain pour la SRL. Le chapitre 5 s'intéresse à la réduction du temps d'intervention humain pour obtenir une annotation sans erreurs.

Un dernier chapitre donnera un bilan de ces travaux et donnera des pistes pour les poursuivre.

Chapitre 2

État de l'art : segmentation et regroupement en locuteurs

Ce chapitre décrit la tâche de segmentation et de regroupement automatique en locuteurs. Après avoir présenté le principe, l'intérêt et les difficultés de la tâche, il présente les différentes parties contenues dans un système de segmentation et regroupement en locuteurs. Ce chapitre se termine par la présentation du système de référence qui sera utilisé dans les expériences.

2.1 Définition de la tâche

La tâche de segmentation et regroupement en locuteurs (*SRL*), aussi connue sous le nom de *speaker diarization*, vise à déterminer le nombre de locuteurs et leurs interventions dans un document audio. C'est une tâche qui s'appuie essentiellement sur la comparaison des voix des locuteurs. Pour pouvoir comparer les voix, il faut des segments de parole homogènes contenant la voix d'un seul locuteur. Un segment de parole est une zone de signal comportant de la parole délimitée par des frontières n'ayant aucune signification particulière. Il est à différencier d'un tour de parole. En effet, un tour de parole est une zone de signal contenant les paroles d'un seul locuteur et dont les frontières indiquent un changement de locuteur ou de nature de données (musique, silence, etc.). Un tour de parole peut ainsi correspondre à plusieurs segments de parole. Il peut arriver que plusieurs personnes parlent en même temps, on parle alors de parole superposée. Cette dernière est, quant à elle, soit représentée par des segments de paroles de locuteurs distincts se chevauchant soit par un seul segment de parole.

La tâche de SRL revient alors à découper le signal audio d'un document en segments de parole puis à les regrouper par locuteur [Anguera *et al.*, 2012]. C'est une tâche de

prétraitement pouvant être utile pour l'identification de locuteur [Bonastre *et al.*, 2000] ou la transcription de parole [Anguera *et al.*, 2012; Geoffrois, 2016] lorsqu'il y a plus d'un locuteur dans un enregistrement audio/vidéo.

Le procédé classique de SRL s'applique individuellement sur chacun des enregistrements audio d'un corpus sans utiliser de connaissance *a priori* sur les locuteurs : leur nombre ainsi que leur identité sont inconnus et aucun modèle ou échantillon de leur voix n'est disponible. La majorité des systèmes de SRL proposés jusqu'à très récemment ont suivi cette définition, où les émissions sont traitées et évaluées individuellement (SRL d'émissions, *single-show diarization*) [Barras *et al.*, 2006; Bredin et Poignant, 2013; Delgado *et al.*, 2015; Soldi *et al.*, 2015]. Dans ce cadre, les locuteurs détectés par les systèmes sont identifiés par des étiquettes anonymes propres à chaque enregistrement. Un même locuteur intervenant ainsi dans deux émissions différentes est donc identifié par deux étiquettes différentes.

La tâche de SRL existe également dans un contexte plus large, où elle vise à déterminer le nombre de locuteurs et leurs interventions non plus dans un seul document, mais dans un ensemble de documents (SRL de collection, *cross-show diarization*) [Delgado *et al.*, 2015; Dupuy, 2015; Dupuy *et al.*, 2014b].

Dans ce manuscrit, nous nous intéressons uniquement au procédé classique, soit la SRL d'émissions sans utiliser de connaissance *a priori* sur les locuteurs.

2.2 Difficultés de la tâche

Actuellement, il n'existe pas de système de SRL capable de traiter indifféremment et efficacement n'importe quel type de document audio. Chaque système de SRL est optimisé pour un type d'enregistrement particulier, voire pour un type de données comme c'est le cas pour les conférences DIHARD [Ryant *et al.*, 2018].

Les archives de l'INA, majoritairement constituées d'émissions de radio et de télévision, ont la particularité d'être très variées. Cette variété se joue sur l'époque, le type d'émission ou encore le type de conditions d'enregistrement. Habituellement, les corpus utilisés pour la tâche de SRL ne présentent pas une aussi grande variété. En effet, ils sont généralement constitués d'un seul type d'émission (principalement des émissions d'information) et correspondent à des enregistrements plus ou moins récents réalisés dans de bonnes et semblables conditions. Ainsi la variété des archives de l'INA est une difficulté à prendre en considération pour optimiser les systèmes de SRL.

Avant de décrire les difficultés de la tâche de SRL concernant les enregistrements d'émissions, il est nécessaire d'introduire la notion de degré de spontanéité. Le degré de spontanéité est lié à la fluidité des échanges entre locuteurs. Plus il est élevé, plus il est

difficile de distinguer qui parle et de comprendre les échanges comme dans les débats ou les interviews. À l'inverse, plus il est faible, plus les échanges ressemblent à des discours préparés comme les émissions d'information. Le degré de spontanéité implique divers phénomènes tels que des passages de parole superposée, les faux départs, etc. [Bazillon *et al.*, 2008; Dufour *et al.*, 2009]. Un degré élevé est une des difficultés majeures de la segmentation. Plus le degré de spontanéité est élevé, plus les tours de parole sont brefs et plus il est difficile de segmenter correctement la parole. Concernant le regroupement en locuteurs, plus il y a de locuteurs présents dans l'enregistrement à traiter, plus le risque de les confondre augmente. Ainsi, la difficulté des émissions de radio et de télévision consiste à éviter de mal segmenter la parole lorsque le degré de spontanéité est élevé, mais également à éviter une surestimation ou une sous-estimation du nombre de locuteurs.

Les émissions de radio et de télévision comportent un nombre de locuteurs très variable d'une émission à l'autre ainsi qu'une quantité non négligeable de silence, de bruit et de musique. Le degré de spontanéité, quant à lui, varie en fonction de l'émission. Par exemple, il est faible pour les documentaires, mais élevé pour les émissions de divertissement. Lors de leur transmission ou de leur archivage, les émissions sont également susceptibles d'être dégradées rendant ainsi leur traitement plus difficile.

2.3 Paramétrisation de la parole : MFCC

Bien qu'il existe d'autres paramètres pour modéliser le signal audio dans divers domaines du traitement de la parole (voir l'annexe A), les paramètres cepstraux continuent à être utilisés depuis une vingtaine d'années. L'intérêt majeur de ces paramètres est qu'ils sont décorrélés entre eux [Haton *et al.*, 2006; Larcher, 2009] ce qui permet de limiter le nombre de coefficients cepstraux requis pour définir l'espace à modéliser.

Les paramètres issus d'analyses cepstrales, obtenus à partir d'une analyse fréquentielle du signal, existent sous diverses formes [Furui, 1981]. Nous présentons ici la paramétrisation cepstrale la plus utilisée dans tous les domaines du traitement de la parole, la paramétrisation MFCC. Les coefficients cepstraux à fréquence Mel [Davis et Mermelstein, 1980; Mermelstein, 1976], *Mel-Frequency Cepstral Coefficients* (MFCC), sont produits par une suite de traitements. Tout d'abord, une analyse fréquentielle dans une fenêtre temporelle glissante est effectuée à partir d'une transformée de Fourier rapide afin de décomposer cette portion du signal en ses fréquences constituantes. Ensuite, le spectre produit par cette décomposition est modulé avant d'être filtré par un banc de filtres triangulaires en échelle Mel [Stevens *et al.*, 1937]. Ce banc de filtres permet de simuler la perception des fréquences par l'oreille humaine. Après ce filtrage, le logarithme des valeurs résultantes est

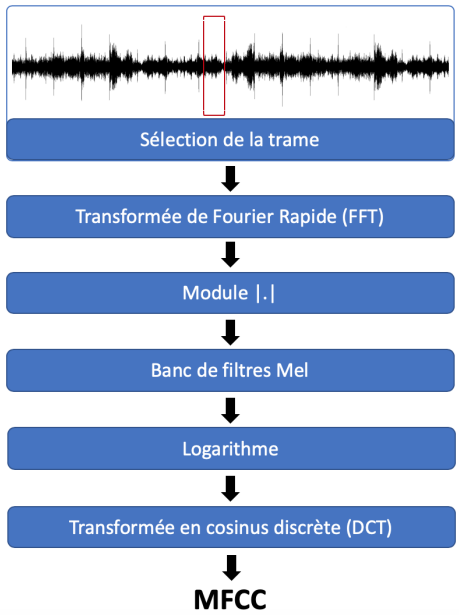


FIGURE 2.1 Les différentes étapes d'extractions des paramètres MFCC

calculé afin d'obtenir l'enveloppe spectrale en décibels. À ce stade, ces paramètres peuvent être utilisés tels quels pour le traitement de la parole. On parle alors de banc de filtres logarithmiques, *Log Filter Bank* (FB). Enfin, si l'on applique une transformée de Fourier inverse sur ces paramètres FB grâce à une transformée en cosinus discrète [Calliope et Fant, 1989], *Discrete Cosine Transform* (*DCT*), nous obtenons les paramètres MFCC. C'est la transformée en cosinus discrète qui permet une décorrélation des données. La figure 2.1 schématise l'extraction des paramètres MFCC.

Habituellement, un vecteur de paramètres acoustiques MFCC est extrait toutes les 10 millisecondes sur une fenêtre temporelle représentant une durée de 20 à 30 millisecondes. Le nombre de paramètres extrait par fenêtre varie généralement entre 12 et 19.

2.3.1 Enrichissement des données acoustiques

Les paramètres MFCC représentent la parole sous forme d'une information dite statique. Afin de tenir compte de la dynamique des paramètres, il n'est pas rare d'ajouter les dérivées temporelles premières et secondes [Furui, 1981]. Plus précisément, la dérivée première modélise la vitesse de variation du spectre alors que la seconde mesure son accélération.

L'énergie du signal présente dans la fenêtre analysée comme ses dérivées peuvent également être intégrées aux vecteurs acoustiques en raison de leurs valeurs discriminantes.

Sa valeur étant proche du coefficient cepstral 0 (c0) des MFCC, il est fréquent qu'elles le remplacent.

2.4 Modélisation du locuteur

Dès lors que nous avons extrait les informations jugées pertinentes du signal de parole, nous pouvons commencer à modéliser le locuteur. Dans cette partie, nous présentons les modèles les plus communément utilisés : le modèle mono-gaussien, le modèle GMM et le modèle i-vecteur.

2.4.1 Modèle : Mono-gaussien

Un des plus simples modèles pour représenter le locuteur est le modèle probabiliste mono-gaussien. Dans un système de SRL, les méthodes de regroupement en locuteurs utilisent fréquemment ce modèle en début de processus. Le modèle mono-gaussien est décrit par deux paramètres : un vecteur de moyenne μ et une matrice de covariance Σ respectivement de dimension D et $D \times D$, où D est la dimension d'un vecteur acoustique. Il n'est pas rare d'utiliser des matrices de covariance diagonale.

La densité de probabilité f qu'un vecteur acoustique v soit issu d'un modèle mono-gaussien $M(\mu, \Sigma)$ est obtenue par la formule suivante :

$$f(v|M(\mu,\Sigma)) = 2\pi^{\frac{-D}{2}}|\Sigma|^{\frac{-1}{2}} \exp\left(\frac{1}{2}(v-\mu)'\Sigma^{-1}(v-\mu)\right). \tag{2.1}$$

Pour une séquence de vecteurs acoustiques $V = \{v_1, \ldots, v_t, \ldots, v_T\}$, la vraisemblance que cette séquence soit issue de $M(\mu, \Sigma)$ est obtenue par le produit des densités :

$$p(V|M(\mu,\Sigma)) = \prod_{t=1}^{T} f(v_t|M(\mu,\Sigma)). \tag{2.2}$$

Plus les observations v_t d'une séquence V sont dans une région où la densité de probabilité est forte, plus il est vraisemblable qu'elles soient issues de cette distribution.

2.4.1.1 Rapports de vraisemblance

Nous présentons dans cette partie deux méthodes classiques pour comparer deux modèles gaussiens : GLR et BIC. D'autres méthodes existent dans la littérature telles que la divergence gaussienne [Barras *et al.*, 2006] ou la distance de Kullbach-Leibler [Delacourt, 2000]. Ces autres méthodes ne sont pas décrites dans ce document.

GLR [Gish *et al.*, 1991] pour rapport de vraisemblance généralisé (*Generalized Likelihood Ratio - GLR*). Pour deux ensembles de paramètres acoustiques V_i et V_j modélisés respectivement par les gaussiennes $M_i(\mu_i, \Sigma_i)$ et $M_j(\mu_j, \Sigma_j)$, la mesure GLR vise à exprimer le rapport de vraisemblance entre l'hypothèse $\mathcal{H}_0$ et l'hypothèse $\mathcal{H}_1$:

- $\mathcal{H}_0$: les deux ensembles V_i et V_j sont issus d'un seul et même locuteur. Ces deux ensembles seraient alors mieux représentés par un modèle unique $M_{ij}(\mu_{ij}, \Sigma_{ij})$;
- $\mathcal{H}_1$: les deux ensembles gaussiens sont issus de deux locuteurs distincts. Ces deux ensembles seraient alors mieux représentés par leurs modèles respectifs $M_i(\mu_i, \Sigma_i)$ et $M_j(\mu_j, \Sigma_j)$.

La mesure prend ainsi la forme suivante :

$$GLR(V_i, V_j) = \log \left(\frac{p(V_i, V_j | M_{ij}(\mu_{ij}, \Sigma_{ij}))}{p(V_i | M_i(\mu_i, \Sigma_i)) p(V_j | M_j(\mu_j, \Sigma_j))} \right). \tag{2.3}$$

BIC [Chen *et al.*, 1998; Schwarz *et al.*, 1978] pour critère d'information bayésien (*Bayesian Information Criterion - BIC*). La mesure BIC, correspondant à une fonction de vraisemblance pénalisée, vise à déterminer si une distribution V composée de N trames est issue d'un modèle gaussien $M(\mu, \Sigma)$. La pénalisation consiste à réduire la vraisemblance en prenant en considération la complexité du modèle :

$$BIC(V | M(\mu, \Sigma)) = \log p(V | M(\mu, \Sigma)) - \lambda \frac{1}{2} \#(M(\mu, \Sigma)) \log N, \tag{2.4}$$

où λ est le facteur déterminé expérimentalement régulant la pénalité et $\#(M(\mu, \Sigma))$ la complexité du modèle. En réalisant une différence entre les critères BIC de deux observations V_i et V_j, il est alors possible d'estimer la proximité entre ces deux dernières et ainsi de savoir si elles sont issues ou non d'un même locuteur. Une valeur positive de cette différence, notée ΔBIC, suggère que les deux observations sont issues d'un même locuteur et qu'elles seraient mieux représentées par un seul modèle $M_{ij}(\mu_{ij}, \Sigma_{ij})$. À l'inverse, une valeur négative indique que les deux observations sont produites par des locuteurs distincts et qu'il serait plus judicieux de les représenter par deux modèles $M_i(\mu_i, \Sigma_i)$ et $M_j(\mu_j, \Sigma_j)$. La formule ΔBIC après simplification [Delacourt, 2000] s'écrit comme suit pour les modèles mono-gaussiens :

$$\Delta BIC(V_i, V_j) = \frac{N_i + N_j}{2} \log |\Sigma_{ij}| - \frac{N_i}{2} \log |\Sigma_i| - \frac{N_j}{2} \log |\Sigma_j| + \lambda P, \tag{2.5}$$

où $|\Sigma_{ij}|$, $|\Sigma_i|$ et $|\Sigma_j|$ représentent respectivement les déterminants des matrices de covariance des modèles M_{ij}, M_i et M_j ; N_i et N_j le nombre de trames respectives des ensembles de vecteurs acoustiques V_i et V_j ; P le facteur de pénalité représentant

la complexité du modèle et λ le facteur régulant le poids de la pénalité déterminé expérimentalement. Le facteur de pénalité, dépendant de N_i et N_j ainsi que de D, s'écrit :

$$P = \frac{1}{2}\left(D + \frac{D(D+1)}{2}\right) + \log(N_i + N_j). \tag{2.6}$$

2.4.2 Modèle : GMM

Depuis son introduction en reconnaissance de la parole dans les années 1990 [Reynolds, 1993, 1995; Rose et Reynolds, 1990], l'approche utilisant des modèles statistiques fondés sur des mélanges de gaussiennes est devenue l'une des références pour modéliser le locuteur. Avec cette approche, lorsque la quantité de données est suffisante, nous obtenons une estimation plus robuste des distributions des vecteurs acoustiques que l'approche utilisant des modèles mono-gaussiens. Un modèle de mélange gaussien, *Gaussian Mixture Model* (*GMM*), est une combinaison linéaire pondérée de plusieurs gaussiennes.

Composé de G gaussiennes, un modèle GMM (θ) est défini par l'ensemble de ses paramètres $\theta = \{w_g, \mu_g, \Sigma_g\}$ avec $g \in \{1 \dots G\}$ où w_g, μ_g et Σ_g représentent respectivement le poids, la moyenne ainsi que la matrice de covariance de la $g^{\text{ème}}$ composante. Les poids sont tels que $w_g > 0$ pour tout $g \in \{1 \dots G\}$ et $\sum_{g=1}^{G} w_g = 1$.

La densité de probabilité qu'un vecteur acoustique v_t soit issu d'un modèle GMM (θ) est la moyenne pondérée des densités des composantes g :

$$y(v_t|\theta) = \sum_{g=1}^{G} w_g f_g(v_t|\mu_g, \Sigma_g). \tag{2.7}$$

Pour une séquence de vecteurs acoustiques $V = \{v_1, \dots, v_t, \dots, v_T\}$, le produit des densités permet d'estimer la vraisemblance que cette séquence soit issue par θ :

$$p(V|\theta) = \prod_{t=1}^{T} y(v_t|\theta). \tag{2.8}$$

L'estimation des paramètres $\theta = \{w_g, \mu_g, \Sigma_g\}$ est la difficulté majeure d'un modèle GMM. Cette étape a des conséquences directes sur la robustesse des modèles de locuteurs générés. Afin d'estimer la valeur de ces paramètres en fonction des données d'apprentissage, deux méthodes sont généralement employées : l'algorithme Espérance-Maximisation et l'adaptation Maximum A Posteriori à partir d'un modèle du monde.

2.4.2.1 Algorithme Espérance-Maximisation

L'algorithme Espérance-Maximisation [Dempster *et al.*, 1977; Laird, 1993], *Expectation-Maximisation* (*EM*), est un procédé itératif. À partir d'une séquence de vecteurs acoustiques V, il vise à estimer par maximum de vraisemblance, *Maximum Likelihood* (*ML*), les paramètres $\theta = \{w_g, \mu_g, \Sigma_g\}$. À chaque itération i, deux étapes se succèdent afin de déterminer de nouveaux paramètres θ^i assurant que la vraisemblance de ce nouveau modèle $p(V|\theta^i)$ soit supérieure à celle du modèle de l'itération précédente $p(V|\theta^{i-1})$.

Les valeurs initiales du modèle sont cruciales puisque l'algorithme Espérance-Maximisation est un procédé convergeant vers un maximum local. Ainsi l'optimalité des paramètres θ n'est pas garantie.

Pour une itération i, la première étape, nommée étape Espérance (*Expectation*) consiste à calculer la densité de probabilité a posteriori d'appartenance de tout vecteur acoustique v_t d'une séquence V à chaque gaussienne g des paramètres courants $\theta^i = \{w_g^i, \mu_g^i, \Sigma_g^i\}$ afin d'obtenir les statistiques d'ordre 0, 1 et 2.

La seconde étape, nommée Maximisation (*Maximization*) consiste à mettre à jour les paramètres du modèle $\theta^i = \{w_g^i, \mu_g^i, \Sigma_g^i\}$ à partir des statistiques calculées dans l'étape précédente. Ces paramètres mis à jour forment les paramètres du nouveau modèle θ^{i+1}. Afin d'éviter qu'une gaussienne ne se spécialise sur un faible nombre d'exemples, les covariances jugées comme étant trop faibles après cette étape peuvent être rétablies à une valeur minimale donnée.

L'initialisation des paramètres θ^1 est généralement réalisée en fixant les poids w_g^1 à $\frac{1}{G}$, en ayant les covariances Σ_g^1 identiques les unes des autres et égales à la covariance des données d'apprentissage. Les moyennes, quant à elles, sont initialisées avec des vecteurs acoustiques pris aléatoirement dans la séquence d'apprentissage V. Une autre initialisation courante est de procéder par divisions successives. Cela consiste à estimer d'abord une seule gaussienne sur l'ensemble des données. Ensuite, le nombre de gaussiennes est multiplié par deux en déviant chaque moyenne de manière aléatoire. Les nouvelles gaussiennes obtenues sont finalement calculées par l'algorithme Espérance-Maximisation. Les étapes de multiplication et d'apprentissage des gaussiennes se réitèrent jusqu'à atteindre le nombre voulu de gaussiennes.

La quantité de données disponibles dans la séquence d'apprentissage V, le nombre de composantes G et la dimension des vecteurs acoustiques D ont des conséquences directes sur la qualité de l'estimation de l'algorithme Espérance-Maximisation.

2.4.2.2 Adaptation Maximum A Posteriori

Lorsque la quantité de données d'apprentissage pour un locuteur n'est pas suffisante, la qualité du modèle GMM estimé par l'algorithme Espérance-Maximisation en est alors diminuée. Deux solutions peuvent alors être envisagées : réduire le nombre de gaussiennes dans le modèle ou adapter un modèle du monde aux données d'apprentissage. Un modèle du monde ou modèle générique de locuteur [Parris et Carey, 1992] (*Universal Background Model - UBM*) n'est autre qu'un modèle GMM estimé avec l'algorithme Espérance-Maximisation sur une quantité importante de données correspondant à de nombreux locuteurs. Bien souvent, cette quantité équivaut à plusieurs heures d'enregistrements. Un modèle du monde, fréquemment composé de 1024 ou 2048 gaussiennes, vise à modéliser une distribution des caractéristiques acoustiques indépendantes du locuteur. En fonction des données d'apprentissage exploitées et de l'objectif, un modèle du monde peut être spécialisé ou non pour un genre ou pour une bande de fréquences.

L'adaptation d'un modèle du monde est communément réalisée grâce au procédé que l'on appelle adaptation Maximum a Posteriori [Gauvain et Lee, 1994; Reynolds *et al.*, 2000], connu sous l'acronyme *MAP*. Ce procédé offre la possibilité d'adapter les paramètres $\theta_{ubm} = \{w_{ubm,g}, \mu_{ubm,g}, \Sigma_{ubm,g}\}$ d'un modèle du monde aux données d'un locuteur. Par cette méthode, les locuteurs disposant de peu de données pour l'apprentissage obtiennent ainsi des modèles GMM de meilleure qualité. Le modèle du monde $\theta_{ubm} = \{w_{ubm,g}, \mu_{ubm,g}, \Sigma_{ubm,g}\}$ est utilisé comme modèle initial. Après le calcul des statistiques à l'étape Espérance, l'étape de Maximisation met à jour les paramètres d'un modèle du locuteur en prenant en considération les paramètres du monde. Par ce procédé, il est possible d'adapter autant les moyennes que les covariances que les poids. Cependant, il a été montré expérimentalement que la simple adaptation des moyennes du modèle du monde [Reynolds *et al.*, 2000] permet d'obtenir de meilleurs modèles GMM. Pour une séquence de vecteurs acoustiques V, l'étape de maximisation correspond à :

$$\hat{\mu}_g^{i+1} = \frac{F_g^i(V) + \tau \mu_{ubm,g}}{N_g^i(V) + \tau}, \tag{2.9}$$

où $N_g^i(V)$ et $F_g^i(V)$ représente respectivement les statistiques d'ordre 0 et 1 de l'étape Espérance pour la gaussienne g à l'itération i, $\hat{\mu}_g^{i+1}$ la moyenne adaptée de la gaussienne g à l'itération $i+1$, $\mu_{ubm,g}$ la moyenne de la gaussienne g du modèle du monde et τ le facteur d'adaptation fixé expérimentalement ($8 \leq \tau \leq 20$). Plus la valeur de ce facteur est élevée, plus l'estimation des nouvelles moyennes du locuteur est proche des moyennes du modèle du monde. Réciproquement, plus la valeur de ce facteur est faible,

plus les nouvelles moyennes du locuteur se rapprochent d'une estimation de l'algorithme Espérance-Maximisation classique.

2.4.2.3 Rapports de vraisemblance

Dans cette partie, nous présentons deux méthodes communes utilisées en SRL pour comparer deux modèles GMM. Même si les mesures GLR et BIC présentées en 2.4.1.1 existent également pour comparer des modèles GMM [He *et al.*, 2010; Moraru, 2004; Stadelmann et Freisleben, 2006], nous ne les décrivons pas dans ce document.

CE [Le *et al.*, 2007; Reynolds, 1995; Solomonoff *et al.*, 1998] pour entropie croisée (*Cross Entropy - CE*) est une mesure visant à estimer la similarité entre deux ensembles de paramètres acoustiques V_i et V_j à partir de leurs modèles respectifs θ_j et θ_i :

$$CE(V_i, V_j) = \frac{1}{N_i} \log \frac{p(V_i, |\theta_i)}{p(V_i, |\theta_j)} + \frac{1}{N_j} \log \frac{p(V_j, |\theta_j)}{p(V_j, |\theta_i)}, \tag{2.10}$$

où N_i et N_j représentent respectivement le nombre de données de l'ensemble V_i et V_j.

CLR [Barras *et al.*, 2006; Reynolds *et al.*, 1998] pour rapport de vraisemblance croisé (*Cross Likelihood Ratio - CLR*) est une mesure qui reprend la mesure de l'entropie croisée, mais qui est envisagée dans le cadre où les deux modèles évalués θ_j et θ_i ont été adaptés au moyen de l'algorithme maximum a posteriori avec un modèle du monde θ_{ubm}. La formule du CLR prend la forme suivante :

$$CLR(V_i, V_j) = \frac{1}{N_i} \log \frac{p(V_i, |\theta_{ubm})}{p(V_i, |\theta_j)} + \frac{1}{N_j} \log \frac{p(V_j, |\theta_{ubm})}{p(V_j, |\theta_i)}. \tag{2.11}$$

Ce score fournit des résultats très proches de l'entropie croisée. Le choix d'utiliser le rapport de vraisemblance croisé ou l'entropie croisée dépend du corpus utilisé [Le *et al.*, 2007].

2.4.3 Modèle : i-vecteur

Depuis leur invention en 2009, les i-vecteurs sont des modèles couramment utilisés dans le domaine de la reconnaissance du locuteur et de la SRL. Avant d'être introduite dans le domaine de la SRL pour le regroupement en locuteurs [Rouvier et Meignier, 2012; Shum *et al.*, 2011], la modélisation i-vecteur est apparue dans celui de la reconnaissance du locuteur [Dehak *et al.*, 2011]. En s'inspirant des méthodes de réduction de dimensionnalité [Kenny,

2010; Kenny *et al.*, 2003; Kuhn *et al.*, 1998; Matejka *et al.*, 2007; Matrouf *et al.*, 2007], notamment de la méthode JFA [Kenny *et al.*, 2007], et des méthodes de décomposition de facteurs, cette approche consiste à réduire les multiples données acoustiques d'un locuteur en un vecteur de dimension réduite. Dans cette partie, nous expliquons la méthode JFA avant d'introduire le modèle i-vecteur.

2.4.3.1 Méthode JFA

Après l'adaptation maximum a posteriori (voir 2.4.2.2), chaque composante g d'un modèle GMM est représentée par un vecteur de moyennes adaptées. À partir de ces vecteurs de moyennes adaptées, il est possible de définir un supervecteur $m = \{\mu_1, \ldots, \mu_g, \ldots, \mu_G\}$ [Betser *et al.*, 2004]. Cette représentation a été initialement employée avec un classifieur de type séparateur à vaste marge (*Support Vector Machine - SVM*) [Campbell *et al.*, 2006]. Par la suite, elle a permis la conception des méthodes de réduction de dimensionnalité et de décomposition en facteurs [Kenny, 2010; Kenny *et al.*, 2003; Kuhn *et al.*, 1998; Matejka *et al.*, 2007; Matrouf *et al.*, 2007]. Ces méthodes et les supervecteurs ont ensuite participé à la conception de la méthode JFA.

La méthode JFA [Kenny *et al.*, 2007], pour *Joint Factor Analysis*, suppose qu'un supervecteur m est le produit de trois composantes indépendantes entre elles : Cc, Ll et Rr. Ces composantes représentent respectivement le canal, le locuteur et le résidu. Pour une session d'enregistrement h appartenant à un locuteur s, la modélisation JFA prend alors la forme suivante :

$$m_{(s,h)} = m_{ubm} + Cc_h + Ll_s + Rr_{s,h}, \tag{2.12}$$

où $m_{(s,h)}$ représente le supervecteur de dimension $G \times D$ dépendant du locuteur s et de la session h ; m_{ubm} le supervecteur d'un modèle du monde (GMM-UBM) indépendant du locuteur et de la session ; C la matrice de variabilité du canal de transmission (*Eigenchannel Matrix*) de dimension $(G \times D, Z)$; c_h le vecteur correspondant au facteur canal de la session h de dimension Z ; L la matrice de variabilité liée au locuteur (*Eigenvoice Matrix*) de dimension $(G \times D, Z)$; l_s le vecteur correspondant au locuteur de la session h de dimension Z ; R la matrice diagonale représentant le sous-espace résiduel du locuteur (*Diagonal Residual*) de dimension $(G \times D, G \times D)$ et $r_{s,h}$ le vecteur résiduel de dimension Z associé au locuteur et à la session.

Concernant la mise en application de la modélisation JFA, elle se résume à estimer les composantes Cc, Ll et Rr à partir de données d'apprentissage annotées en locuteur avant de pouvoir extraire les vecteurs c, l et r d'une observation donnée.

2.4.3.2 Du JFA au i-vecteur

En partant de l'approche JFA, les auteurs de Dehak [2009]; Dehak *et al.* [2011]; Matrouf *et al.* [2008] supposent que la composante canal C possède en outre des informations sur le locuteur. Ils proposent ainsi une approche où la composante locuteur L et la composante canal C sont représentées dans un seul et même espace nommé espace de variabilité totale (*total variability space*). Cet espace est une matrice contenant l'ensemble des variabilités importantes du locuteur. Puisque les composantes L et C sont représentées dans un espace commun, la composante résiduelle R n'a plus lieu d'être. La modélisation i-vecteur se formule alors ainsi :

$$m_{(s,h)} = m_{ubm} + \mathcal{T}\omega_{s,h}, \tag{2.13}$$

où $m_{(s,h)}$ représente le supervecteur dépendant du locuteur s et de la session h ; m_{ubm} le supervecteur d'un modèle du monde (GMM-UBM) de dimension $G \times D$ indépendant du locuteur et de la session ; $\mathcal{T}$ la matrice rectangulaire de dimension réduite $(G \times D, B)$ représentant l'espace de variabilité totale et $\omega_{s,h}$ le i-vecteur, un vecteur de dimension B représentant le locuteur et le canal et qui suit une loi normale $\mathcal{N}(0, I)$. La dimension B varie habituellement entre 50 et 400.

Concernant l'estimation des paramètres, l'espace de variabilité totale $\mathcal{T}$ est réalisé à partir de données d'apprentissage contenant plusieurs sessions de plusieurs locuteurs. Quant au i-vecteur, il est obtenu grâce à l'algorithme *Factor Analysis* (*FA*) en prenant en considération la particularité de l'espace de variabilité totale $\mathcal{T}$. Cet algorithme impliquant une réduction de dimensionnalité ainsi qu'une décomposition en facteurs permet ainsi d'éliminer une quantité d'information conséquente.

2.4.3.3 Rapports de vraisemblance

Il existe différentes possibilités pour comparer des i-vecteurs. Dans cette partie, nous présentons une des méthodes les plus classiques qui est utilisée en SRL : la méthode PLDA. D'autres méthodes sont données en annexe A. La méthode PLDA [Jiang *et al.*, 2012] pour analyse discriminante linéaire probabiliste (*Probabilistic Linear Discriminant Analysis*) a initialement été proposée dans le domaine de l'image pour la reconnaissance des visages [Prince et Elder, 2007]. Elle a été adaptée au domaine de la vérification du locuteur afin de modéliser la distribution des modèles i-vecteur [Kenny, 2010] avant d'être largement répandue dans les systèmes de reconnaissance du locuteur [Jiang *et al.*, 2012; Matějka *et al.*, 2011]. L'approche PLDA peut être envisagée comme un cas spécial mono-gaussien de l'approche JFA [Kenny *et al.*, 2008]. L'apprentissage du modèle PLDA requiert d'avoir à

sa disposition un corpus d'apprentissage représentant plusieurs enregistrements de chacun des locuteurs. Pour un enregistrement ou une observation h d'un locuteur s, le i-vecteur $\omega_{(s,h)}$ peut être résumé par la somme de trois composantes :

$$\omega_{(s,h)} = \mu + Cc_{s,h} + Ll_s + \epsilon_{s,h}, \tag{2.14}$$

où μ est la moyenne globale des i-vecteurs du corpus d'apprentissage, indépendante du locuteur s et de l'observation h ; L l'espace de variabilité du locuteur représenté par une matrice ; l_s le facteur locuteur (*Eigenvoice Factor*) ; C la matrice désignant l'espace de variabilité de l'observation ; $c_{s,h}$ le facteur canal de l'observation h pour le locuteur s et $\epsilon_{s,h}$ l'espace de variabilité résiduel modélisé par une gaussienne à matrice de covariance pleine. Les facteurs l_s et $c_{s,h}$ sont considérés comme indépendants et suivent une loi normale $\mathcal{N}(0, I)$. L'estimation des paramètres du modèle PLDA $\{\mu, L, C, \epsilon\}$ est réalisée grâce à l'algorithme Espérance-Maximisation. À partir de deux i-vecteurs ω_i et ω_j extraits de données de test, deux hypothèses sont testées :

- $\mathcal{H}_0$: les deux modèles i-vecteur sont issus par un seul et même locuteur. Ces deux modèles partagent alors un même facteur locuteur l ;
- $\mathcal{H}_1$: les deux modèles i-vecteur sont issus par deux locuteurs distincts. Ces deux modèles possèdent en conséquence un facteur locuteur l différent.

Le score de vraisemblance PLDA entre deux modèles i-vecteur ω_i et ω_j est calculé comme suit :

$$score_{PLDA}(\omega_i, \omega_j) = \log \frac{p(\omega_i, \omega_j|\mathcal{H}_0)}{p(\omega_i, \omega_j|\mathcal{H}_1)}. \tag{2.15}$$

Habituellement, les i-vecteurs ω_i et ω_j subissent une normalisation EFR [Bousquet *et al.*, 2012] ou SNN [Bousquet *et al.*, 2012] avant d'être évalués par le score de vraisemblance PLDA.

2.4.4 Discussion

La modélisation x-vecteur, proposée en 2016 par [Snyder *et al.*, 2016] et décrite en annexe A, a obtenu de meilleures performances que la modélisation i-vecteur lors de la campagne d'évaluation NIST 2018 [NIST, 2018]. Au moment de la rédaction de ce manuscrit, la plupart des systèmes de SRL récemment développés et étudiés reposent sur la fusion de systèmes à base de x-vecteurs avec des systèmes à bases de i-vecteurs [Huang *et al.*, 2018; Patino *et al.*, 2018b; Sell *et al.*, 2018].

La thèse a commencé en décembre 2015, c'est-à-dire la période avant l'apparition des x-vecteurs. Nos recherches s'appuient sur les systèmes à l'état de l'art de cette époque, soit les systèmes à base de i-vecteurs.

2.5 Segmentation du signal audio

Dans cette section, nous présentons une méthode classique de segmentation du signal audio : la segmentation GLR/BIC. La version développée au LIUM, que nous détaillons ici, s'inspire de l'approche proposée dans Delacourt et Wellekens [2000]. La segmentation GLR/BIC est réalisée en deux passes. La première passe vise à détecter les ruptures correspondant aux changements de locuteurs. Elle est réalisée par l'utilisation de la mesure GLR (voir 2.4.1.1). La seconde passe, de son côté, consiste à améliorer la sortie de la première passe en regroupant des segments consécutifs au moyen de la mesure BIC (voir 2.4.1.1). Il n'est pas rare que la mesure GLR soit remplacée par la divergence gaussienne [Barras *et al.*, 2006] qui fournit également de bons résultats.

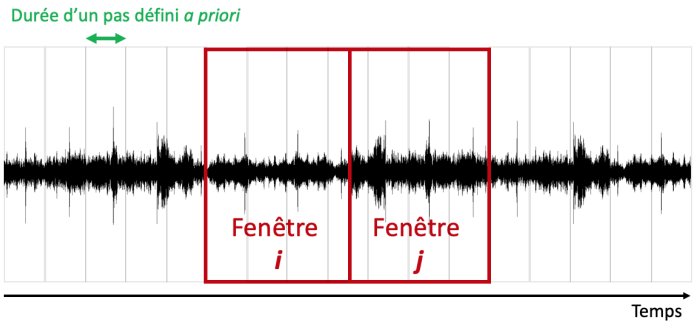


FIGURE 2.2 Deux fenêtres glissantes i et j se décalant d'un pas défini *a priori* pour la détection des points de rupture acoustique

La passe de détection de ruptures repose sur l'utilisation de deux fenêtres glissantes temporelles consécutives i et j. Les deux fenêtres de taille fixe et identique parcourent le signal audio sur toute sa longueur selon un pas défini a priori. La figure 2.2 illustre ces deux fenêtres ainsi que leurs déplacements. Les fenêtres, contenant un certain nombre de trames, sont modélisées par des modèles mono-gaussiens à matrices de covariance pleine. Pour un instant donné t, dès que le score GLR associé atteint un maximum local, un changement de locuteur est détecté entre les deux fenêtres. Cependant, cette segmentation a des difficultés à déterminer correctement les frontières lorsque les tours de parole ont une durée inférieure à la somme des durées des deux fenêtres glissantes.

Après la passe de détection de ruptures, la passe de regroupement de segments consécutifs est appliquée. Comme pour la première passe, des modèles mono-gaussiens à matrice de covariance pleine sont utilisés pour représenter les segments. Au lieu d'utiliser le rapport de vraisemblance généralisé, on utilise le critère d'information bayésien (*Bayesian Information Criteria - BIC*) présenté en 2.4.1.1, car contrairement à la première passe, les gaussiennes ne sont pas apprises sur des quantités de données identiques. Pour chaque paire de segments consécutifs s_i et s_{i+1}, la seconde passe consiste à évaluer la différence entre leurs critères d'information bayésien respectifs, ΔBIC, afin d'en estimer leur proximité et ainsi savoir s'ils sont issus ou non d'un même locuteur. Si le score de ΔBIC est positif, alors la paire de segments s_i et s_{i+1} est jugée comme appartenant à un même locuteur et les segments sont fusionnés. À l'inverse, un score négatif suppose que les segments de la paire appartiennent à deux locuteurs différents et les segments ne sont alors pas regroupés. À l'issue de cette seconde passe, les segments produits sont suffisamment longs pour utiliser des modèles de représentation du locuteur plus complexe.

2.6 Regroupement en locuteurs

Dans cette section, nous commençons par définir le regroupement en locuteurs. Ensuite, nous présentons des regroupements qui sont devenus des procédés classiques dans le domaine de la SRL.

2.6.1 Définition du regroupement en locuteurs

À l'issue d'une segmentation en locuteurs, le signal audio est représenté par une suite de segments en principe homogènes correspondant à des locuteurs différents. L'objectif du regroupement en locuteurs est alors de regrouper les segments présents dans l'ensemble du signal audio par locuteur. Pour décrire un ensemble de segments, partageant des caractéristiques communes, on utilise communément le terme de classe.

Diverses approches de regroupement en locuteurs existent dans la littérature, mais toutes offrent la possibilité d'être utilisées de manière complémentaire. L'approche la plus commune repose sur l'évaluation d'une matrice de similarité entre chaque paire de segments présents dans le document audio. Habituellement, l'approche mono-gaussienne/BIC [Dupuy *et al.*, 2012] est utilisée en tant que pré-requis avant de pouvoir utiliser des représentations plus complexes telles que le modèle GMM ou le i-vecteur, nécessitant plus qu'un segment de parole par locuteur. Avec une matrice de similarités, il est possible d'appliquer des méthodes classiques de regroupement non supervisé comme la classification ascendante hiérarchique

[Chen *et al.*, 1998; Gish *et al.*, 1991; Siegler *et al.*, 1997; Siu *et al.*, 1992; Solomonoff *et al.*, 1998], des méthodes de regroupement combinatoire comme la méthode K-moyennes (*K-Means*) [Shum *et al.*, 2011] ou encore des méthodes à base de graphes [Shum *et al.*, 2013].

Dans la sous-partie qui suit, nous détaillons uniquement la classification ascendante hiérarchique qui est la méthode la plus répandue.

2.6.1.1 Classification ascendante hiérarchique

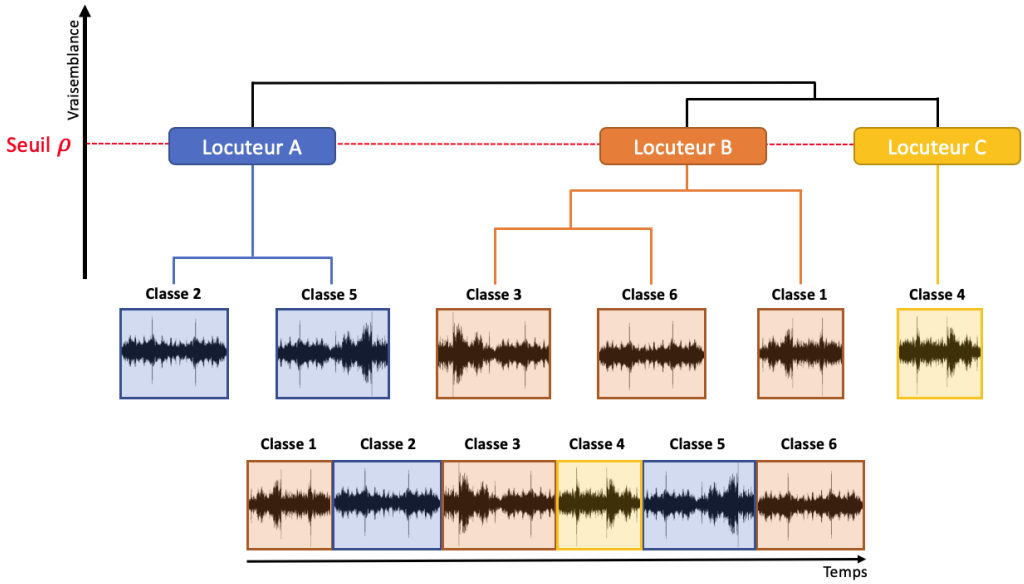


FIGURE 2.3 Classification ascendante hiérarchique avec un critère d'arrêt ρ

La classification ascendante hiérarchique (*Hierarchical Agglomerative Clustering - HAC*) est une méthode itérative visant à regrouper successivement plusieurs classes initiales en fonction d'un score caractérisant leur similarité. Pour une itération donnée, les deux classes "*les plus proches*" sont regroupées en une nouvelle classe. Le score entre cette classe et les autres est ensuite calculé avant de mettre à jour la matrice de scores. Ce procédé se réitère tant qu'il reste des classes à fusionner ou que les conditions d'un critère d'arrêt ne sont pas remplies. Le critère d'arrêt peut soit prendre la forme d'un nombre de classes à atteindre soit un score ρ à dépasser. La figure 2.3 représente un dendrogramme, une représentation d'une classification ascendante hiérarchique sous forme d'arbre où la hauteur de la jonction des arcs d'une classe correspond au score entre les deux classes avant la fusion. La mise à jour des scores peut être réalisée de deux manières. La première consiste à réestimer un nouveau modèle sur les données des classes fusionnées, puis à estimer les

scores avec les autres modèles. La seconde manière consiste à ne réestimer aucun modèle, mais à effectuer la mise à jour à partir des scores entre les éléments des classes. Cette seconde manière est réalisée au moyen de mesures inter-classes telles que le saut minimum (*single-linkage*) [Sibson, 1973], le saut maximum (*complete-linkage*) [Defays, 1977], le lien moyen (*average-linkage*) [Wilks, 2011] ou encore le critère de Ward (*Ward-linkage*) [Ward Jr, 1963]. Dans cette partie, nous ne détaillons que le saut minimum et maximum.

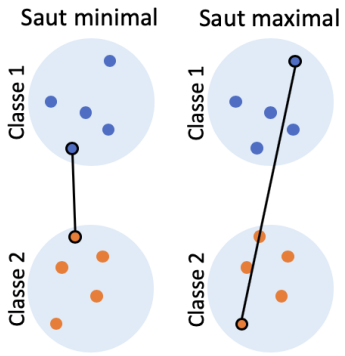


FIGURE 2.4 Illustration du saut minimal et maximal entre deux classes

Saut minimum Le saut minimum entre 2 classes A et B consiste à prendre le minimum de toutes les distances possibles entre les segments de A et B. Il est défini par :

$$D_{min}(A,B) = \min_{z \in A, y \in B}(score(z,y)). \tag{2.16}$$

Cette mesure de similarité inter-classe, sensible aux individus bruités, a tendance à produire des classes générales par effet de chaînage [Bruynooghe, 1978]. La figure 2.4 illustre le saut minimum.

Saut maximum Le saut maximum entre 2 classes A et B consiste à prendre la distance la plus grande entre les segments de A et B. Il est défini par :

$$D_{max}(A,B) = \max_{z \in A, y \in B}(score(z,y)). \tag{2.17}$$

Cette mesure, également sensible aux individus bruités, a tendance à créer des classes spécifiques puisqu'elle ne regroupe que des classes très proches. La figure 2.4 illustre le saut maximum.

Les modèles utilisés pour calculer les scores entre les segments à regrouper peuvent être soit des gaussiennes avec ΔBIC, soit des GMM avec CE ou CLR, soit des i-vecteurs avec PLDA.

2.6.2 Regroupement i-vecteur/ILP

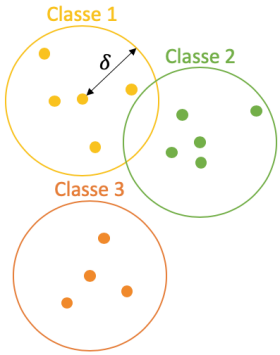


FIGURE 2.5 Schématisation du regroupement i-vecteur/ILP

Le regroupement par programmation linéaire en nombres entiers (*Integer Linear Programming - ILP*) a été proposé pour la première fois dans Rouvier et Meignier [2012]. Ce procédé extrait un i-vecteur (voir 2.4.3) pour chaque classe et évalue ensuite leurs scores de similarité entre eux. Le regroupement i-vecteur/ILP est semblable à un problème de sélection de médoïdes. Un médoïde correspond au point le plus proche du centre d'une classe. Plus précisément, le regroupement i-vecteur/ILP consiste à chercher K médoïdes couvrant tous les i-vecteurs tels que chaque i-vecteur soit attribué à un seul médoïde et ait un score inférieur à δ de leur médoïde. Le nombre K, lui, est à déterminer automatiquement.

En schématisant, le regroupement i-vecteur/ILP vise à positionner un minimum de disques de rayon δ dont chacun est centré sur un i-vecteur et où chaque locuteur est au moins représenté par un disque. Cette schématisation est illustrée par la figure 2.5.

Le regroupement i-vecteur/ILP vise à minimiser la formule suivante tout en respectant diverses contraintes :

$$\text{Minimiser : } \sum_{i=1}^{Y} x_{\omega_i,\omega_i} + \frac{1}{F+1} \sum_{i=1}^{Y} \sum_{j=1}^{Y} score(\omega_i, \omega_j) x_{\omega_i,\omega_j}, \tag{2.18}$$

$$\text{Contraintes : } x_{\omega_i,\omega_j} \in \{0,1\} \qquad i \in \{1,\ldots,Y\}, j \in \{1,\ldots,Y\}, \tag{2.19a}$$

$$\sum_{i=1}^{Y} x_{\omega_i,\omega_j} = 1, \tag{2.19b}$$

$$x_{\omega_i,\omega_j} - x_{\omega_i,\omega_i} \leq 0, \tag{2.19c}$$

$$score(\omega_i, \omega_j) x_{\omega_i,\omega_j} < \delta. \tag{2.19d}$$

où $\mathcal{C} = \{1, \ldots, Y\}$ est l'ensemble des classes initiales, F un facteur de normalisation et $score(\omega_i, \omega_j)$ le score de similarité entre les médoïdes ω_i et ω_j. La contrainte 2.19a indique que les variables sont binaires : lorsque $x_{\omega_i,\omega_j} = 1$, l'i-vecteur ω_j appartient à la classe dont le médoïde est ω_i et lorsque $x_{\omega_i,\omega_j} = 0$, ω_j n'appartient pas à cette classe. La contrainte 2.19b signifie que chaque i-vecteur est contraint d'être affecté à un seul médoïde ω_i. La contrainte 2.19c induit que si la classe ayant pour médoïde ω_i est non vide alors elle contient au minimum le i-vecteur ω_i.

Dans l'équation 2.18 que l'on cherche à minimiser, la première partie $\sum_{i=1}^{Y} x_{\omega_i,\omega_i}$ illustre le nombre de médoïdes tandis que la deuxième partie représente la dispersion de chaque solution normalisée par le facteur $\frac{1}{F+1}$. Pour que cette seconde partie soit comprise dans l'intervalle $[0; 1[$, F correspond à la somme de tous les scores de similarité inférieurs au seuil δ. Ceci a pour conséquence de choisir l'affectation des points aux médoïdes ayant la plus faible dispersion pour un nombre K identique. Ce problème de regroupement est résolu par une méthode générique pour la résolution de problèmes d'optimisation discrète, l'algorithme *Branch & Bound*. Cette méthode explore chaque solution de manière exhaustive. L'algorithme *Branch & Bound* est un procédé qui ne s'exécute pas en temps polynomial. Ainsi dans certains cas de figure, le temps de calcul peut devenir déraisonnable.

Afin de réduire le nombre de solutions à explorer et donc d'économiser du temps d'exécution, Dupuy *et al.* [2014b] propose une méthode simple. Les auteurs montrent que les scores de similarité inférieurs à δ peuvent être représentés sous la forme d'un graphe où les i-vecteurs représentent les noeuds et les scores les longueurs des arêtes. Chaque composante connexe de ce graphe peut être vue comme un sous-problème de plus faible complexité pouvant être traité indépendamment. Si une composante connexe forme une étoile ou est composée uniquement d'un noeud, alors nous obtenons une classe. Un graphe étoile correspond à un ensemble de noeuds connectés seulement à un noeud commun. Si une composante connexe ne forme pas une étoile alors il faut appliquer un regroupement i-vecteur/ILP.

2.7 Traitements complémentaires

Dans cette section, nous présentons des traitements complémentaires aux méthodes de segmentation et de regroupement décrites précédemment.

2.7.1 Resegmentation avec l'algorithme de Viterbi

Les frontières des segments peuvent être remises en question au moyen de l'algorithme de Viterbi [Viterbi, 1967]. Cet algorithme s'applique sur un modèle de Markov caché (*Hidden Markov Model - HMM*) [Baum et Petrie, 1966] où chaque état représente une classe-locuteur. Les modèles de Markov cachés sont définis comme étant des automates probabilistes à états finis. Ils visent à déterminer la probabilité d'émission d'une séquence d'observations donnée. Au moyen d'un modèle de Markov caché, l'algorithme de Viterbi permet de segmenter un signal audio en fonction de ses caractéristiques acoustiques [Barras *et al.*, 2004b; Meignier *et al.*, 2001]. Pour représenter chaque classe-locuteur, soit un état dans un modèle de Markov caché, un modèle GMM (voir 2.4.2) est utilisé. Un modèle est estimé à partir de l'algorithme Espérance-Maximisation (voir 2.4.2.1) sur l'ensemble des segments associés à la classe. Les modèles GMM sont utilisés préférentiellement à la place des modèles mono-gaussiens en raison d'un grand nombre de données disponibles pour les locuteurs. La pénalité associée à un changement de classe lors d'un décodage, quant à elle, est fixée expérimentalement.

2.7.2 Détection de la parole

La détection d'activité vocale (*Speech Activity Detection - SAD*) consiste à identifier la nature des trames afin d'extraire celles qui contiennent de la parole. Cette technique de détection de parole repose le plus souvent sur un décodage de Viterbi avec un modèle de Markov caché à deux états au minimum [Le Lan, 2017] : plusieurs modèles pour la parole et d'autres pour le silence et le bruit.

2.8 Évaluation

Dans cette section, nous présentons les différentes campagnes d'évaluation liées au domaine de la SRL ainsi que les corpus associés que nous utilisons dans cette thèse. Nous présentons également comment évaluer un système de SRL.

2.8.1 Campagnes d'évaluation

Les campagnes d'évaluation permettent d'évaluer ainsi que de comparer des méthodes du domaine de la parole dans des conditions d'exploitation contrôlées. Depuis 1996, diverses campagnes d'évaluation nécessitant de segmenter et de regrouper les locuteurs ont eu lieu. Les campagnes les plus importantes sont brièvement décrites dans cette partie.

RT pour *Rich Transcription* sont les campagnes d'évaluation les plus importantes pour la tâche de SRL organisées par le *National Institute of Standards and Technology* (*NIST*), un organisme proposant des campagnes d'évaluation américaines. La première campagne date de 2002 et la dernière de 2009. Les campagnes *Rich Transcription* de 2002 à 2004 [NIST, 2002, 2003, 2004a] concernent principalement le traitement des émissions radio et télévisées alors que celles des années suivantes s'intéressent sur le traitement des réunions [NIST, 2004b, 2005, 2006, 2007, 2009].

ESTER est une campagne d'évaluation française organisée par l'Association francophone de la communication parlée (*AFCP*), la Direction générale de l'armement (*DGA*) et l'Agence de distribution des ressources du langage et des évaluations (*Evaluations and Language resources Distribution Agency - ELDA*). Il y en a eu deux : ESTER 1 en 2003 et ESTER 2 en 2009. Ces campagnes ont permis d'évaluer des systèmes de transcription enrichie sur des émissions de radio française. Concernant la SRL, les campagnes ESTER intègrent une évaluation de la tâche.

ETAPE est une campagne d'évaluation proposée en 2012, s'inscrivant dans la continuité des campagnes ESTER. Cette campagne est la première qui offre en plus la possibilité d'évaluer une collection d'émissions. Par rapport aux données de la campagne ESTER, celles de la campagne d'ETAPE ont un degré de spontanéité plus élevé.

REPERE pour défi ANR REPERE est une campagne d'évaluation organisée par le Laboratoire national de métrologie et d'essais (*LNE*), la DGA et l'ELDA. Cette campagne, ayant pour thème la reconnaissance des personnes dans des émissions télévisées françaises de BFM et LCP, a été proposée au cours de trois années : REPERE 0 en 2012 (test à blanc), REPERE 1 en 2013 et REPERE 2 en 2014. Le thème de cette campagne vise plus précisément à indiquer à chaque instant l'identité des personnes qui parlent ou sont visibles ainsi que les noms incrustés dans une émission télévisée. Lors de cette campagne [Galibert et Kahn, 2013], les systèmes de SRL ont quant à eux été évalués en plus de l'identification des personnes.

MGB est une campagne d'évaluation se concentrant sur la SRL, la reconnaissance de la parole, la détection de dialecte et l'alignement des sous-titres à l'audio pour des enregistrements d'émissions télévisées multi-genres en anglais et en arabe [Ali *et al.*, 2016, 2017; Bell *et al.*, 2015]. Cette campagne a été proposée au cours de trois années : MGB 1 en 2015, MGB 2 en 2016 et MGB 3 en 2017. En fonction des années, cette campagne a été intégrée en tant que session spéciale soit à la conférence *ASRU* soit à la conférence *SLT*.

DIHARD est une campagne d'évaluation introduite en 2018 [Ryant *et al.*, 2018] en tant que session spéciale à la conférence *Interspeech*. Cette évaluation se concentre

sur la tâche de SRL appliquée à des données difficiles telles que des vidéos YouTube ou des enregistrements dans des restaurants. Grâce au succès rencontré, elle est reconduite en 2019.

Cette liste n'est pas exhaustive. En effet, on peut également ajouter les campagnes d'évaluations Albayzin [Zelenák *et al.*, 2012] et SITW [McLaren *et al.*, 2016a].

Dans cette thèse, nous utilisons 4 corpus provenant des campagnes d'évaluations ESTER, ETAPE et REPERE. Ces corpus sont brièvement détaillés dans le tableau 2.1. Pour tous ces corpus, nous traitons l'intégralité des enregistrements sans utiliser l'information a priori sur les zones annotées données par les fichiers *UEM* (*Un-Partitioned Evaluation Map*) associés aux corpus.

Campagne	ESTER		ETAPE	REPERE
Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1
Nature	radio	radio	radio+TV	TV
Langue	français	français	français	français
Parole préparée	oui	oui	oui	oui
Degré de spontanéité	faible	faible	fort	moyen
Durée des enregistrements	10h07	7h12	8h39	14h16
Durée de l'annotation	9h52	7h10	6h58	2h57
# de stations	6	4	3	2
# d'émissions	9	8	11	7
# d'enregistrements	18	26	15	28
# de locuteurs	342	250	148	158
Moyenne # de locuteurs par émission	20,17	11,53	10,33	7,5

TABLE 2.1 Détails des corpus utilisés dans le manuscrit de thèse

2.8.2 Mesure d'évaluation

La mesure d'évaluation typique pour évaluer la qualité d'un processus de SRL est le *Diarization Error Rate*, plus connue sous le sigle (*DER*). Introduite par le NIST pour les campagnes d'évaluations RT [NIST, 2003], elle est définie comme la proportion de temps de parole qui n'est pas correctement attribuée au bon locuteur. Le calcul de cette mesure nécessite de comparer l'hypothèse générée par un système de SRL à une vérité terrain ou référence, tout en considérant la meilleure correspondance entre cette hypothèse et cette référence. La meilleure correspondance est celle qui maximise les temps de cooccurrences entre les locuteurs de la référence et les locuteurs de l'hypothèse [Galibert, 2013].

La mesure du DER se résume à la somme de trois types d'erreurs, illustrées dans la figure 2.6 :

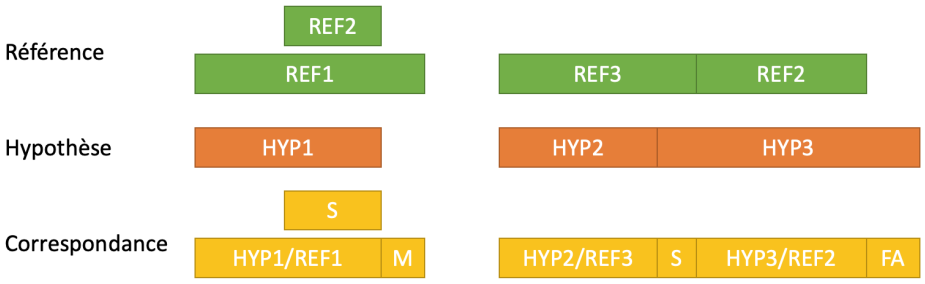


FIGURE 2.6 Illustration des types d'erreurs pour le calcul du DER

M pour *Miss*, correspond aux détections manquées de locuteur. Elle représente la durée des zones de l'hypothèse associées à aucun locuteur. Cette erreur correspond au cas où le système n'a pas détecté de parole alors qu'il y en a.

FA pour *False Alarm*, correspond aux fausses détections de locuteur. Elle représente la durée des zones de l'hypothèse attribuées à tort à un locuteur. Cette erreur correspond au cas où le système a détecté de la parole alors qu'il n'y en a pas.

S pour *Substitution*, correspond aux substitutions entre locuteurs. Elle représente la durée des zones de parole de l'hypothèse associées au mauvais locuteur, en considérant l'appariement optimal entre les locuteurs de la référence et ceux de l'hypothèse.

La mesure du DER est définie comme la somme des trois erreurs détaillées précédemment divisée par T la durée totale de parole dans la référence :

$$DER = \frac{M + FA + S}{T}. \tag{2.20}$$

Lorsqu'on veut calculer le DER, il faut au préalable savoir si l'on veut accorder une tolérance de quelques millisecondes aux frontières des segments de la référence et si l'on prend en considération les zones de parole superposée. Le degré de spontanéité des données à évaluer permet de nous aider dans ces choix. En effet, plus les tours de parole sont courts, plus la tolérance aux frontières de la référence doit être faible. De même, plus les zones de parole superposée sont conséquentes, plus les écarter de l'évaluation rend le DER non significatif. Pour des documents contenant de la parole préparée comme les journaux radiophoniques, il est commun d'admettre une tolérance de 250 ms et d'ignorer les zones de parole superposée. Pour des documents contenant un fort degré de spontanéité, au contraire, il est courant d'avoir une tolérance aux frontières moindre et d'évaluer la parole superposée.

Les mesures de DER présentes dans ce manuscrit sont obtenues à partir de l'outil du LNE [Galibert, 2013], mais avec une légère modification. L'outil du LNE a été développé au cours des campagnes d'évaluation ETAPE [Gravier *et al.*, 2012] et REPERE [Galibert et Kahn, 2013]. Il s'appuie sur l'algorithme hongrois [Munkres, 1957] pour trouver la meilleure correspondance entre les locuteurs de la référence et ceux de l'hypothèse. La modification, quant à elle, concerne le traitement des zones de tolérance. Au lieu d'utiliser le traitement du LNE, c'est celui du NIST qui est appliqué. Contrairement à l'outil du LNE, les zones de tolérances ne sont pas évaluées avec l'outil du NIST.

Le DER est une évaluation dépendante de la durée de prise de parole de chaque locuteur. Plus précisément, elle favorise la détection des locuteurs intervenant le plus. Par exemple, si un locuteur parlant 80% du temps est correctement détecté et qu'une dizaine d'autres ne le sont pas alors le DER sera de 20%. D'autres mesures d'évaluation complémentaires existent pour évaluer une sortie d'un système de SRL. On peut citer la précision, le rappel ou encore la F-mesure qui sont des mesures connues pour l'évaluation des frontières des tours de parole [Cettolo *et al.*, 2005; Vandecatseye *et al.*, 2004].

2.9 Architecture du système SRL de référence

Le système de SRL à base de i-vecteurs auquel nous nous référons lors de nos expériences et qui est encore couramment utilisé à l'heure actuelle est illustré par la figure 2.7. Ce système correspond quasiment à celui présenté dans Dupuy *et al.* [2014a]. La seule différence est que l'étape de la détection de la parole est effectuée dans les premières étapes et non dans les dernières. Dans cette partie, nous décrivons le système de SRL de référence optimisé pour les émissions radiophoniques et télévisées.

Le système de SRL de référence se divise en 7 étapes (figure 2.7). La première est la paramétrisation de la parole. Elle consiste à extraire des vecteurs de paramètres acoustiques correspondant à 12 MFCC auxquels on ajoute un coefficient pour l'énergie, le tout complété par leurs dérivées premières et secondes. La seconde étape est une détection de parole, utilisant un algorithme de Viterbi avec 8 modèles GMM à 64 gaussiennes de covariance diagonale : trois modèles de parole (parole propre, parole avec du bruit et parole avec une musique de fond), un modèle de parole téléphonique, deux modèles de silences (bande passante étroite et large), un modèle pour les jingles et un autre pour la musique. Ces modèles sont obtenus à partir des données extraites du corpus de la campagne ESTER 1 au moyen de l'algorithme Espérance-Maximisation (voir 2.4.2.1). La troisième étape est une étape de segmentation réalisée au moyen d'une segmentation GLR/BIC, utilisant des modèles mono-gaussiens avec des matrices de covariance diagonale. Les modèles de la

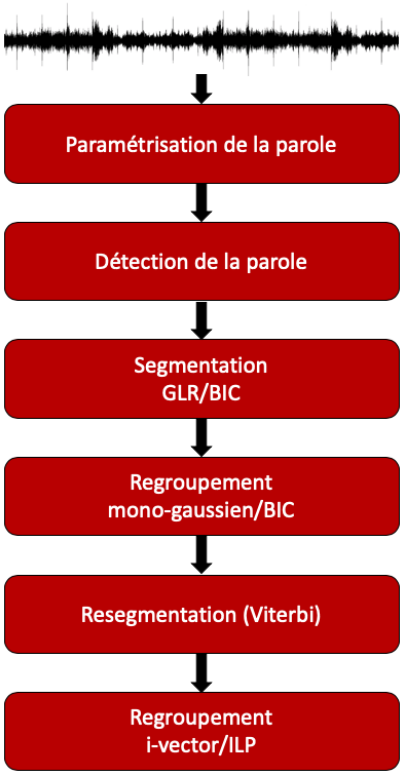


FIGURE 2.7 Différentes étapes du système de SRL de référence

fenêtre de gauche et de la fenêtre de droite sont estimés sur une fenêtre de 5 secondes (2,5 secondes pour chaque modèle) glissante sur tout le signal. La seconde passe de segmentation est effectuée grâce au critère ΔBIC avec un facteur λ fixé expérimentalement à 2. L'étape suivante est un regroupement HAC mono-gaussien/BIC avec un facteur λ fixé expérimentalement à 3 pour le ΔBIC. Ce regroupement utilise des matrices de covariance pleines pour ses modèles mono-gaussiens. La cinquième étape correspond à une resegmentation en utilisant un décodage de Viterbi avec des modèles GMM à 8 gaussiennes de matrice de covariances diagonale et une pénalité d'insertion de 250. La dernière étape, quant à elle, consiste à appliquer un regroupement i-vecteur/ILP, plus complexe, mais plus performant du point de vue du DER qu'un regroupement GMM/CLR-CE. Le regroupement i-vecteur/ILP utilise la PLDA pour comparer les i-vecteurs. Le tableau 2.2 illustre les meilleurs résultats pour divers corpus. Remarquons que la variation du DER des corpus est proportionnelle à la proportion du degré de spontanéité dans les corpus (tableau 2.1). Les tableaux 2.3 et 2.4, quant à eux, montrent les temps d'exécution pour chaque étape. Ces temps ont été obtenus à partir des serveurs du LIUM. Ces serveurs ne sont pas tous identiques. Ainsi les temps d'exécution peuvent légèrement varier d'un serveur à un autre.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil δ optimal	50	40	40	40	40
DER	5,44	7,10	27,04	10,44	11,54

TABLE 2.2 Évaluation du système de SRL de référence sur divers corpus avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible DER

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Segmentation GLR/BIC	0,21	0,14	0,17	0,29	0,81
Regroupement mono-gaussien/BIC	0,64	0,33	0,84	1,28	3,09
Resegmentation	3,68	2,13	2,94	5,35	14,10
Regroupement i-vecteur/ILP	12,75	9,31	12,22	18,57	52,85
Toutes étapes confondues	17,78	11,91	16,17	25,49	71,35

TABLE 2.3 Temps d'exécution en minutes des différentes étapes du système de SRL de référence sur divers corpus

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Segmentation GLR/BIC	0,04	0,03	0,04	0,16	0,05
Regroupement mono-gaussien/BIC	0,12	0,08	0,20	0,72	0,19
Resegmentation	0,68	0,49	0,70	3,02	0,87
Regroupement i-vecteur/ILP	2,35	2,16	2,92	10,49	3,27
Toutes étapes confondues	3,28	2,77	3,87	14,40	4,41

TABLE 2.4 Ratio du temps d'exécution des différentes étapes du système de SRL de référence par rapport à la durée du corpus

2.10 Bilan

Dans ce chapitre, nous avons présenté la tâche de segmentation et regroupement en locuteurs ainsi que ses difficultés. Nous avons montré comment modéliser la voix d'un locuteur, mais également comment la comparer à celle d'autres locuteurs. En outre, ce chapitre nous a permis de détailler la structure typique d'un système de SRL. Il a montré comment évaluer un système de SRL et a présenté celui sur lequel nous allons nous appuyer dans le reste du manuscrit. Il a également présenté les corpus sur lesquels nous allons effectuer nos expériences.

Concernant le cœur de la thèse, ce chapitre permet de donner un aperçu des limites d'un système entièrement automatique. Il a mis en évidence qu'un système automatique, au moyen du système de référence appliqué sur divers corpus, présente toujours des erreurs. Dans le chapitre suivant, nous présentons le concept de systèmes assistés par l'humain.

Après une description du concept, nous nous intéressons aux travaux en SRL dans ce domaine.

Chapitre 3

État de l'art : Systèmes assistés par l'humain

Ce chapitre expose les besoins liés aux systèmes assistés par l'humain. Il montre comment tirer avantage d'une information fournie par un humain dans une interaction homme-machine afin d'obtenir des performances supérieures à un système entièrement automatique. Il détaille également comment un humain peut être intégré dans un système assisté.

Après une introduction aux systèmes assistés par l'humain, ce chapitre introduit le cas de la SRL assistée par l'humain. Or, ce sujet ayant très peu été abordé, la littérature associée reste presque inexistante. Par conséquent, ce chapitre présente d'autres domaines connexes afin de faire émerger des pistes.

3.1 Principe de la tâche

FIGURE 3.1 Schématisation de l'intervention humaine sur la sortie d'un système automatique

Les systèmes entièrement automatiques sont développés dans le but principal de pouvoir remplacer entièrement l'humain dans des tâches complexes nécessitant des efforts psychiques ou physiques. De nos jours, ces systèmes n'arrivent pourtant pas toujours à égaler ou surpasser les performances d'un être humain. Pour parvenir à un tel niveau de performance,

un humain peut intervenir sur la sortie d'un système automatique (figure 3.1). Cependant, cette intervention est dans la plupart des cas onéreuse et chronophage. Afin de pallier ces problèmes, différents types d'assistance ont vu le jour. Ils peuvent être séparés en deux catégories complémentaires : l'assistance à l'humain et l'assistance au système automatique (figure 3.2).

Assistance à l'humain Cette assistance vise à améliorer l'environnement où l'humain intervient sur la sortie d'un système automatique [Cardinal *et al.*, 2007; Nanjo *et al.*, 2006; Ogata *et al.*, 2007]. Plus précisément, cela consiste à proposer divers modules complémentaires visant en définitive à réduire le temps d'intervention humaine. Ces modules peuvent avoir différentes fonctions. Par exemple, un module peut fournir de nouvelles données qui n'étaient pas initialement disponibles dans l'environnement de base afin de réduire le temps de réflexion humain ou remplacer certaines composantes de la sortie par d'autres éléments plus probables sans avoir recours au clavier, réduisant ainsi les interactions physiques.

Assistance au système automatique Elle permet à l'humain de fournir des informations au système automatique afin qu'il améliore ses performances et réduise ainsi l'effort et le temps d'intervention humain total. En anglais, on emploie le terme de *Computer assisted system* [Toselli *et al.*, 2011e] ou de *Interactive system* [Geoffrois, 2016] pour décrire les systèmes automatiques intégrant ce type d'assistance. En français, il n'y a pas de terme qui fasse consensus. Ainsi, nous décidons de parler de système *assisté par l'humain* pour faire référence à un système automatique intégrant cette assistance.

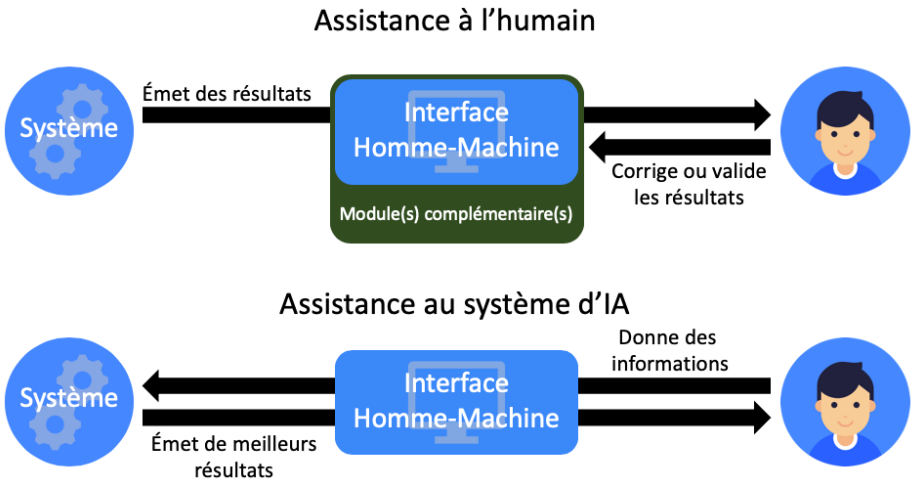


FIGURE 3.2 Schématisation de l'assistance à l'humain et de l'assistance au système automatique

Dans ce manuscrit, nous nous focalisons sur l'assistance au système automatique, soit les systèmes assistés par l'humain. Cette assistance, correspondant à la capacité d'un système à apprendre d'une interaction humaine, est actuellement un besoin important pour divers domaines [Geoffrois, 2016]. Dans la reconnaissance de la parole par exemple, un sujet de recherche actif [Orosanu et Jouvet, 2015] est de permettre aux utilisateurs de fournir de nouveaux mots à un système afin qu'il puisse les reconnaître.

La plupart des technologies pour le traitement de l'audio, de la vidéo ou du texte actuellement développées se concentrent sur une implémentation entièrement automatique et ignorent les possibilités d'amélioration que peut offrir l'exploitation des interventions humaines. La principale raison associée à ce choix est que, dans la majorité des cas, le taux d'erreur est jugé acceptable pour l'application voulue.

3.2 Conception des systèmes assistés

La conception classique d'un système automatique s'appuie sur trois ensembles de données annotés manuellement : les données d'apprentissage, les données de développement et les données de test. Le coût d'annotation de ces données n'est pas pris en compte dans la mesure d'évaluation. *A contrario*, dans la conception de systèmes assistés, la mesure d'évaluation doit prendre en considération l'effort et le temps d'annotation mis par l'humain. En effet, on vise à solliciter le moins possible l'humain et à réduire le plus possible le taux d'erreurs.

3.3 Besoins pour les systèmes assistés

En 1974 dans le magazine américain *IEEE* [Jarvis, 1974], un magazine dédié à la recherche dans le domaine de l'informatique, les qualités d'un système assisté ont été formulées ainsi :

> "*An interactive computer environment is one which attempts to facilitate the interplay between man and machine in pursuit of a goal defined by man. Presumably, to be effective, this environment should allow the calculating speed, precision, and structured logical/iterative skill of the machine to serve the conceptual, intuitive, highly associative, and contextually sensitive attributes of human mental function in the solution of problems [...] The areas of graphics, image processing, and pattern recognition are particularly appropriate candidates for this interactive approach.*"

> "*Un environnement informatique interactif est un environnement qui tente de faciliter l'interaction entre l'homme et la machine dans un but défini par l'homme. Vraisemblablement, pour être efficace, cet environnement devrait prévoir la vitesse de calcul, la précision et l'aptitude itérative/logique structurée de la machine pour être au service des éléments conceptuels, intuitifs, associatifs et sensibles contextuellement de l'activité mentale humaine dans la solution des problèmes [...] Les domaines du traitement graphique, du traitement d'image et de la reconnaissance de formes sont des candidats particulièrement appropriés pour cette approche interactive.*"

Depuis cet article et au vu du faible nombre de publications sur le sujet, il est fort probable que les systèmes assistés n'ont exploité qu'une très faible partie des possibilités d'amélioration que peuvent offrir les interventions humaines. Il est donc fort probable que nombre de qualités restent encore à identifier.

La précision et l'efficacité d'un système assisté importent peu dès lors que le temps total nécessaire à l'obtention de résultats jugés corrects par un humain est trop élevé. En effet, il ne faut pas qu'un humain mette moins de temps à effectuer manuellement le travail souhaité qu'un système assisté, dans ce cas un système assisté perd tout intérêt.

L'humain, lui aussi, est plus ou moins précis et efficace : la psychologie de l'individu, les compétences de l'individu et les aspects sociaux sont des facteurs à prendre en compte [Dix, 2009] pour évaluer l'économie de temps et d'effort d'une intervention humaine dans un système assisté. En outre, nous devons garder à l'esprit qu'un humain n'est pas parfait et peut donc faire des erreurs. Par conséquent, il faut qu'un système assisté puisse détecter une intervention humaine porteuse d'erreurs et prendre les mesures nécessaires.

3.4 Charges physiques et psychiques demandées

Lorsqu'un humain donne des informations à un système assisté, cela requiert une certaine charge physique et psychique. Ces charges dépendent avant tout de l'application et de l'objectif. L'ergonomie de l'application joue un rôle important. En effet, plus une application est ergonomique, plus les charges physiques et psychiques sont réduites [Dix, 2009].

La charge physique dépend des formes prises par les interactions de l'homme avec la machine. Pour transmettre de l'information, l'humain interagit habituellement avec le système informatique en utilisant la souris et le clavier. Ces périphériques ont l'avantage d'être des interactions déterministes, c'est-à-dire qu'ils informent directement de l'action voulue de l'humain. En utilisant d'autres moyens d'interaction non déterministes comme une caméra ou un microphone, une phase d'interprétation est alors nécessaire pour extraire

l'action voulue de l'humain [Toselli *et al.*, 2011e], ce qui peut engendrer une contrainte physique supplémentaire. En effet, si la phase d'interprétation extrait une action autre que celle voulue, l'humain doit soit réitérer l'interaction jusqu'à ce que la phase d'interprétation soit comprise correctement, soit utiliser un autre moyen d'interagir avec le système assisté. Par exemple, pour un segment mal labellisé dans une SRL, un humain peut énoncer à haute voix au système de SRL assisté le bon label. Dans ce cadre, si la phase d'interprétation comprend un autre label, l'humain doit soit prononcer à nouveau le label adéquat, soit le taper directement au moyen d'un clavier.

La charge psychique, quant à elle, peut grandement varier pour une même tâche. Par exemple pour la tâche de reconnaissance de locuteur, en fonction de l'application, l'humain peut répondre à "*Est-ce bien le bon locuteur qui parle ?*" ou à "*Quel est le nom du locuteur ?*".

Notons qu'une application peut proposer plusieurs actions ou combinaisons d'actions, associées à des charges physiques et psychiques différentes, pour transmettre une même information.

3.5 Utilisation des interactions entre l'homme et la machine

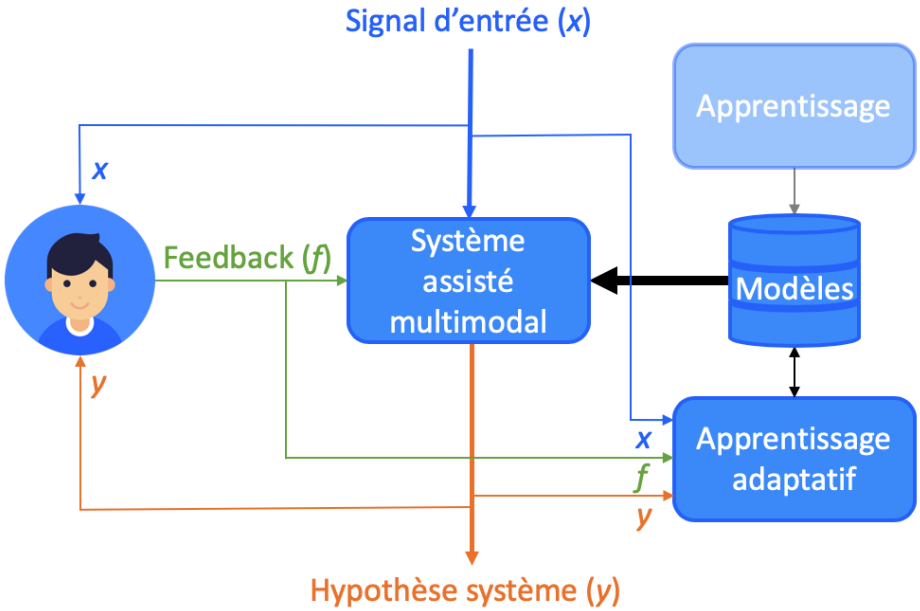


FIGURE 3.3 Illustration détaillée d'un système assisté

Le passage des systèmes entièrement automatiques à celui d'assistés par l'humain met en évidence trois axes de recherche possibles :

Feedback : À chaque interaction homme-machine, le système assisté vise à extraire les informations utiles de l'intervention humaine afin d'améliorer directement ses performances initiales.

Multimodalité : Le système assisté vise à harmoniser les différentes natures de données entrant afin d'améliorer ses performances.

Adaptation : Le système assisté réentraîne et adapte ses modèles grâce aux données extraites du feedback.

La figure 3.3 illustre ces trois possibilités de recherche. Dans cette figure, x représente le signal d'entrée, qui peut par exemple être du son, une image ou du texte ; y l'hypothèse fournie par le système assisté ; f le feedback, c'est-à-dire l'information que l'humain donne au système. Lorsque l'humain détecte une absence de correspondance entre le signal d'entrée x et l'hypothèse y, il intervient en proposant un feedback f qui peut aider le système à améliorer ses prédictions. L'humain continue d'intervenir jusqu'à ce qu'il considère que le système assisté ne commet plus d'erreurs ou commet des erreurs acceptables. Le système assisté, quant à lui, détermine une hypothèse à partir du feedback f et des modèles qu'il a à sa disposition. Les modèles sont initialement obtenus à partir de données d'apprentissage de paires signal d'entrée/hypothèse, comme pour les systèmes entièrement automatiques classiques. Cet apprentissage est illustré en bleu clair dans la figure. À chaque interaction homme-machine, les modèles sont adaptés grâce à un apprentissage adaptatif qui tire avantage des feedbacks humains.

3.5.1 Feedback

Le feedback d'un humain peut prendre différentes formes. Il peut s'agir d'informations visant à améliorer la prise de décision du système ou de simples directives pour améliorer des paramètres perceptifs du système tel que la mise au point de caméra. Le feedback humain f est rarement de la même nature que le signal d'entrée x.

Dans le cas d'informations visant à améliorer la prise de décision, les interactions homme-machine offrent, en elles-mêmes, la possibilité d'améliorer l'efficacité du système assisté sans même réaliser une adaptation des modèles. En effet, il est possible de tirer directement avantage des informations utiles qu'elles contiennent afin que le système puisse proposer des hypothèses plus pertinentes. Cependant, estimer la meilleure hypothèse y à associer à un signal d'entrée x en prenant en considération, en plus des modèles, un

feedback f rend la prise de décision plus complexe que dans les systèmes entièrement automatiques classiques ne prenant en compte que les modèles.

Dans la plupart des systèmes assistés, un historique des feedbacks se forme au fur et à mesure des interventions humaines. Cet historique est souvent utilisé dans son ensemble pour améliorer la prédiction des hypothèses du système. L'utilisation d'un historique au lieu d'un seul feedback pour déterminer la meilleure hypothèse y à partir d'un signal d'entrée x ne rend pas la prise de décision nécessairement plus complexe [Toselli *et al.*, 2011e].

3.5.2 Multimodalité

Dans les systèmes assistés par l'humain, la multimodalité des données existe à deux niveaux. Le premier niveau concerne le signal d'entrée qui peut être un ensemble complexe de divers types de données. Par exemple, le signal d'entrée peut être du texte avec du son. Le deuxième niveau concerne la nature du signal d'entrée qui est différente de celle du feedback humain. Par exemple, dans le cadre d'un système de reconnaissance de texte dans un document audiovisuel, il est très peu probable que le feedback humain soit sous la forme d'une image, mais plutôt sous la forme de mouvements de souris et de touches clavier. La principale difficulté d'une multimodalité des données est de réussir à atteindre une bonne coopération entre chaque modalité permettant ainsi de tirer un avantage optimal de chacune d'elles.

Dans ce manuscrit, cet axe de recherche n'est pas étudié.

3.5.3 Adaptation

En plus des avantages fournis en 3.5.1, les feedbacks offrent naturellement la possibilité d'utiliser les informations qu'ils contiennent afin d'adapter les modèles du système. Cette adaptation peut se faire à deux niveaux. Le premier niveau concerne l'adaptation des modèles utilisés pour associer une hypothèse y à un signal d'entrée x à partir du feedback. Dans la figure 3.3, ceci est illustré par l'apprentissage adaptatif. Le second niveau, quant à lui, concerne l'adaptation des modèles utilisés pour l'interprétation d'un feedback non déterministe à partir des données de feedback interprétées.

Nous approfondissons essentiellement cet axe de recherche dans ce manuscrit.

3.6 Apprentissage assisté par l'humain

Dans cette partie, nous présentons différentes adaptations existantes dans la littérature permettant d'améliorer le système assisté.

3.6.1 Apprentissage en ligne

Au fur et à mesure des interactions humaines, le système assisté acquiert un certain nombre d'hypothèses corrigées ou validées liées à des signaux d'entrée. L'apprentissage en ligne, aussi connu sous le nom de *Online Learning* [Barras *et al.*, 2004a; Soldi *et al.*, 2015], consiste à apprendre et à adapter le système à partir des interactions. Il considère les interactions humaines comme un flux de données d'apprentissage continu.

L'approche la plus basique de l'apprentissage en ligne consiste à combiner les données d'apprentissage avec celles des corrections et validations humaines pour obtenir de nouvelles données d'apprentissage pour les modèles du système. Puisque ces données sont incrémentées au fur et à mesure des interactions humaines, cette approche s'est vu attribuer le nom d'apprentissage incrémental (*Incremental learning*) [Koshinaka *et al.*, 2012; Ross *et al.*, 2008].

3.6.2 Apprentissage actif

L'humain peut interagir de deux manières avec le système, soit de manière passive soit de manière active. La manière passive consiste à demander à l'humain de corriger toutes les hypothèses proposées y pour chaque signal d'entrée x tout en suivant leur ordre d'apparition. La manière active, quant à elle, consiste à laisser le système décider quelles hypothèses nécessitent une supervision humaine et dans quel ordre. En outre, le système assisté peut décider d'arrêter de faire appel à l'humain dès lors que la qualité est estimée suffisante. L'apprentissage actif [Dasgupta, 2009; Zhang *et al.*, 2013], aussi connu sous le nom d'*Active Learning*, s'appuie sur l'interaction active. Cet apprentissage concerne les contextes où le système n'arrive pas à trouver les hypothèses correctes de certains signaux. Ces hypothèses sont alors obtenues à partir d'une intervention humaine. L'apprentissage actif se résume alors à minimiser le nombre de demandes d'assistance à l'humain et plus particulièrement l'effort et le temps d'intervention humain, tout en maximisant les performances du système assisté. Cela est généralement réalisé en appliquant un critère de sélection sur tous les signaux d'entrée afin de sélectionner celui qui diminue le plus les erreurs du système. Ce signal est alors proposé à l'humain pour obtenir l'hypothèse adéquate. Ce processus se réitère jusqu'à ce que le système considère que toutes les hypothèses sont acceptables.

3.6.3 Apprentissage semi-supervisé

Dans un système assisté, il existe deux types de sorties : les sorties validées ou corrigées par un humain et les sorties estimées par le système sans l'aide d'une correction humaine.

Ces deux types de sorties sont respectivement nommés sorties supervisées et sorties non supervisées pour le reste de cette partie. Les sorties supervisées offrent la possibilité d'améliorer le système, par exemple, au moyen d'un apprentissage en ligne. Néanmoins, elles ne sont pas les seules. En effet, les sorties non supervisées, qui peuvent contenir d'éventuelles erreurs, apportent également des informations utiles pouvant rendre le système plus efficace. L'apprentissage semi-supervisé (*Semi-Supervised learning*) [Chapelle *et al.*, 2009] consiste à tirer avantage de ces deux types de sorties pour fournir de meilleures prédictions.

3.6.4 Apprentissage avec des feedbacks limités

Dans la section 3.4, nous avons expliqué que la charge psychique d'une correction humaine dépend de l'application. Par exemple, dans le domaine de la transcription assistée, les questions posées à un humain peuvent être "*Est-ce la bonne transcription ?*" ou "*Quelle est la transcription correcte ?*". Ainsi, la quantité d'informations fournie par le feedback peut être plus ou moins importante.

Dans le cas où l'humain ne donnerait pas directement la solution, le système peut utiliser les informations contenues dans le feedback pour proposer une nouvelle hypothèse. Plus précisément, les informations contenues dans le feedback sont utilisées pour adapter les modèles afin que cette erreur de prédiction ne se reproduise plus. Pour ce cas de figure, on parle d'apprentissage avec des feedbacks limités [Kakade *et al.*, 2008], *Learning with limited-feedback* en anglais. Ce type d'apprentissage fait partie des apprentissages par renforcement (*Reinforcement learning*) [Jaksch *et al.*, 2010]. Les apprentissages par renforcement visent à renforcer les prédictions d'un système au fil du temps en tirant le maximum de bénéfice qu'il peut obtenir de son environnement.

3.7 SRL assistée par l'humain

Dans cette partie, nous présentons la SRL assistée par l'humain. Nous définissons la tâche avant d'étudier les systèmes assistés dans la littérature.

3.7.1 Définition de la tâche

Bien que les systèmes SRL entièrement automatiques actuels offrent de bonnes performances, ils n'atteignent toujours pas les performances d'un humain. En effet, il reste toujours un nombre important d'erreurs présentes dans la SRL dû à divers phénomènes tels que la parole superposée ou encore des interventions de très courte durée. Selon l'application

visée, les performances des systèmes entièrement automatiques peuvent ne pas suffire. Ainsi, une intervention humaine corrigeant les erreurs des systèmes automatiques peut alors être appliquée. Cependant, une intervention humaine coûte cher, demande du temps et surtout un effort humain considérable [Bazillon *et al.*, 2008]. Une solution adéquate est donc d'avoir un système qui prend en considération des interventions humaines pour améliorer la SRL. Plus précisément, cela consiste à appliquer le concept des systèmes assistés à la tâche de SRL.

3.7.2 Systèmes assistés dans la littérature

La SRL assistée par l'humain reste quasiment inexistante dans la littérature. La seule étude notable est celle de Budnik *et al.* [2014]. Cet article utilise cependant des systèmes multimodaux et n'est donc pas uniquement dédié à la SRL assistée. Les auteurs utilisent un apprentissage actif conjointement avec des systèmes de reconnaissance automatique du locuteur, de texte et du visage. L'utilisation de l'apprentissage actif dans leur système montre qu'il permet de diminuer le nombre de corrections humaines.

Même s'il y a une absence de SRL assistée par l'humain dans la littérature, de nombreux articles d'autres domaines peuvent servir de pistes. En effet, ces articles proposent des méthodes pouvant être facilement appliquées au domaine de la SRL assistée.

Par exemple, dans Basu *et al.* [2004] dans le contexte du regroupement de textes, les auteurs proposent deux méthodes. La première est une méthode d'apprentissage actif qui tente d'interroger le moins souvent possible un humain afin d'obtenir le plus de contraintes de regroupement informatives. Les contraintes prennent la forme de données devant ou ne devant pas être regroupées. La deuxième méthode est une méthode de regroupement utilisant un apprentissage semi-supervisé (voir 3.6.3). Plus précisément, cette méthode vise à regrouper des données en fonction d'une mesure qui prend en considération la similarité des données et le coût de violer des contraintes de regroupement. Dans l'article, ces deux méthodes ont montré qu'elles améliorent significativement le regroupement de textes pour une faible quantité d'information humaine fournie.

Un autre exemple est celui de Soldi *et al.* [2015] où les auteurs proposent une méthode de SRL entièrement automatique pour des données de réunion où les intervenants se présentent en début de chaque enregistrement. Pour un enregistrement et en s'appuyant sur la vérité terrain associée, une certaine quantité de données de chaque locuteur de la présentation est extraite. La méthode proposée dans l'article, appliquée sur le reste du document, utilise alors ces données afin d'améliorer ses performances en SRL. Dans l'article, cette méthode a montré une amélioration des performances.

Dans les parties qui suivent, nous présentons un panorama de systèmes assistés dans des domaines connexes à la SRL. Outre le fait que des articles provenant d'autres domaines peuvent proposer des méthodes pouvant être applicable au domaine de la SRL assistée, ils peuvent également proposer des mesures ou donner des pistes pour évaluer le coût d'intervention humain dans un système de SRL assistée.

3.8 Traduction assistée

Dans cette partie, nous présentons les mesures d'erreurs utilisées pour évaluer la traduction assistée. Nous détaillons par la suite plusieurs systèmes de traduction assistés. Les systèmes de traduction assistés ont déjà été étudiés depuis au moins 1985 pour résoudre divers problèmes d'ambiguïté que ce soit au niveau lexical, syntaxique ou sémantique [Barrachina *et al.*, 2009; Slocum, 1985; Whitelock *et al.*, 1986]. Cependant, peu d'articles sur ce domaine sont disponibles dans la littérature. La plupart des publications s'articulent principalement autour du projet *TransType* [Foster *et al.*, 1997, 2002; Langlais *et al.*, 2000; Langlais et Lapalme, 2002; Nepveu *et al.*, 2004; Patry et Langlais, 2009; Simard et Isabelle, 2009] et de son successeur *TransType2* [Barrachina *et al.*, 2009; Bender *et al.*, 2005; Casacuberta *et al.*, 2009; Civera *et al.*, 2004a,b; Cubel *et al.*, 2004, 2003; Och *et al.*, 2003; Tomás et Casacuberta, 2006]. Ce projet incorpore un système de traduction complet intégré dans un environnement d'édition interactif permettant de générer la suite la plus probable de chaque phrase cible en cours de traduction.

3.8.1 Définition de la tâche

Au moyen des technologies de traduction automatique (*Machine-Translation*) à l'état de l'art [Wu *et al.*, 2016], il n'est pas toujours possible d'obtenir des traductions de qualité comparable à celle d'un humain entre n'importe quelles paires de langage. Dans certains cas, cette qualité peut être suffisante pour l'application visée. Dans les autres cas, la sortie de ces technologies doit être post-traitée afin d'obtenir la qualité désirée. Ainsi, la traduction est un domaine adéquat pour pouvoir intégrer des solutions tirant avantage des interactions humaines afin d'améliorer les performances. Les systèmes intégrant ce genre de méthode sont communément appelés systèmes de traduction assistés par l'humain (*Interactive Machine Translation - IMT* ou *Computer-Assisted Translation - CAT*) [Barrachina *et al.*, 2009; Bender *et al.*, 2005].

3.8.2 Mesures d'erreurs utilisées

En raison des nombreuses traductions possibles acceptables pour une phrase source donnée, l'évaluation d'un système de traduction est un sujet faisant débat [Callison-Burch *et al.*, 2010]. Dans le cas d'un système de traduction assisté, le problème de l'évaluation est davantage accentué puisque l'évaluation doit prendre en compte en plus le coût d'intervention humain. Actuellement pour les systèmes de traductions assistés, diverses mesures ont été utilisées ou mises en places pour évaluer l'effort et le temps d'interaction nécessaire à un traducteur humain pour obtenir une traduction de référence.

WSR [Tomás et Casacuberta, 2006; Toselli *et al.*, 2011d] pour *Word Stroke Ratio* est la mesure la plus souvent utilisée pour estimer l'effort d'un traducteur humain dans un cadre d'un système de traduction assisté. Le WSR correspond au nombre de mots à corriger nécessaires pour obtenir la traduction de référence, divisé par le nombre total de mots présents dans cette traduction de référence. Le WSR est égal au Word Error Rate (*WER*) [Toselli *et al.*, 2011d] lorsqu'aucune méthode d'aide à la traduction n'est mise en place. Le WER est un taux d'erreur des différences entre les traductions d'hypothèse et celles de référence. Ce taux est calculé en sommant le nombre d'insertions, de suppressions et de substitutions de mots à effectuer dans l'hypothèse puis en divisant ce résultat par le nombre de mots présents dans la référence. L'utilisation du WER et du WSR offre alors la possibilité de comparer le nombre de mots à corriger avec ou sans aide à la traduction.

KSR [Barrachina *et al.*, 2009; Bender *et al.*, 2005; Tomás et Casacuberta, 2006] pour *Keyboard Stroke Ratio* est une mesure initialement intégrée dans les systèmes de communication assistés (*Augmentative and Alternative Communication*) destinés aux personnes handicapées. Il est calculé à partir du nombre de frappes au clavier effectué par l'utilisateur pour écrire un message. Dans Barrachina *et al.* [2009], les auteurs l'adaptent au domaine de la traduction assistée. Le KSR représente alors le nombre de frappes au clavier nécessaires pour obtenir la traduction de référence divisé par le nombre total de caractères de la référence.

MAR [Barrachina *et al.*, 2009] pour *Mouse-Action Ratio* est une mesure peu utilisée qui s'appuie sur le principe du KSR. En effet, au lieu de se concentrer sur le clavier, cette mesure se concentre sur la souris. Le MAR représente le nombre de mouvements de pointeurs souris plus le nombre de phrases, divisé par le nombre total de caractères de la traduction de référence. Le nombre de phrases a été rajouté afin de pouvoir simuler l'action de l'humain nécessaire pour accepter la traduction finale.

KSMR [Barrachina *et al.*, 2009] pour *Keyboard Stroke and Mouse-Action Ratio* est une mesure également peu utilisée qui n'est autre que l'addition du KSR avec le MAR.

Puisque le KSR estime les actions de l'humain à partir du clavier tandis que le MAR à partir de la souris, ces mesures s'appuient donc sur des types d'effort physique différents [Macklovitch, 2006]. Dans Barrachina *et al.* [2009], les auteurs utilisant le KSMR supposent que les actions du KSR et du MAR le composant requièrent des efforts physiques humains similaires.

3.8.3 Systèmes assistés dans la littérature

Dans cette partie, nous présentons quelques systèmes assistés liés au domaine de la traduction.

Dans Toselli *et al.* [2011d], les auteurs proposent un système de traduction assisté générant un graphe de mots. Ce graphe est simplement un espace de recherche restreint pour la traduction d'une phrase donnée [Barrachina *et al.*, 2009]. Il est généré une fois pour chaque phrase. Ce graphe de mots est un graphe orienté acyclique et pondéré. Chaque nœud représente la position dans la phrase à traduire tandis que chaque arc est étiqueté avec un mot de la phrase cible associée à un poids correspondant aux scores des modèles de langage et de traduction. Dès qu'une correction humaine est effectuée sur la traduction proposée par le système, cette correction ainsi que les mots précédents sont appliqués sur le graphe de mot. Une recherche semblable à celle du décodage de Viterbi est ensuite appliquée sur ce graphe de mots afin de proposer au correcteur une nouvelle traduction plus probable. Le WSR obtenu pour un tel système assisté sur les corpus Xerox [Cubel *et al.*, 2004; SchlumbergerSema, 2001] allant de l'anglais à l'espagnol et European Union [SchlumbergerSema, 2001] allant du français à l'anglais est respectivement d'environ 30% et de 40%. Les WER de ces corpus ont des valeurs presque identiques à celui du WSR. Ainsi, cette méthode interactive ne fournit ni d'amélioration ni de dégradation. Cependant, en modifiant légèrement la qualité de la traduction finale attendue, une réduction intéressante des interactions humaines est remarquée. En effet, pour le corpus European Union, en permettant de fournir des traductions avec un WER autour de 10%, il est possible de faire passer le WSR à une valeur en dessous de 40%. Des résultats similaires sont obtenus avec le corpus Xerox.

Dans Barrachina *et al.* [2009], l'étude a été effectuée dans le cadre du projet TransType2 [SchlumbergerSema, 2001] avec des traducteurs professionnels sur les corpus Xeros et European Union. Les auteurs évaluent des systèmes assistés s'appuyant sur différentes technologies classiques de traduction. En comparant le WSR au WER, les auteurs remarquent

qu'ils évoluent de la même manière. Pour finir, ils obtiennent un gain d'interactions comparable à Toselli *et al.* [2011d].

3.9 Transcription de la parole assistée

Dans cette partie, nous présentons les mesures d'erreurs utilisées pour évaluer la transcription de la parole assistée avant d'étudier des systèmes assistés présents dans la littérature. Le nombre d'études associées à cette tâche reste faible dans la littérature.

3.9.1 Définition de la tâche

La reconnaissance automatique de la parole (*Automatic Speech Recognition - ASR*) a été largement utilisée ces dernières années, mais elle n'est pas toujours efficace pour obtenir une transcription d'une qualité comparable à celle d'un humain [Yu et Deng, 2016]. En effet, en raison de divers phénomènes tels qu'un grand vocabulaire ou de la parole superposée, un nombre conséquent d'erreurs peuvent apparaître dans la transcription. Pour atteindre une meilleure transcription, une intervention humaine est généralement requise pour corriger les erreurs du système ASR. L'utilisation de système d'assistance à la transcription de la parole (*Computer Assisted Transcription of Speech - CATS*) [Laurent, 2010; Toselli *et al.*, 2011a], prenant en considération des interventions humaines pour améliorer la transcription, semble alors être une bonne solution pour réduire le travail de l'humain.

Pour la suite de cette partie, il est nécessaire de définir deux termes : le préfixe et le suffixe. Le préfixe est la partie de la transcription hypothèse correspondant aux mots de la dernière correction ou validation humaine et aux mots qui les précèdent. Le suffixe, quant à lui, correspond aux mots suivants et non vérifiés.

3.9.2 Mesures d'erreurs utilisées

Pour estimer le coût d'intervention humain nécessaire à l'obtention d'une transcription correcte, il existe diverses mesures dans la littérature. Les principales mesures sont brièvement expliquées dans cette partie.

WSR [Civera *et al.*, 2004b; Laurent, 2010; Laurent *et al.*, 2011; Rodríguez *et al.*, 2007; Toselli *et al.*, 2011a] pour *Word Stroke Ratio* est une mesure utilisée à l'origine dans les systèmes de traduction assistés par l'humain. Elle représente le nombre de mots à corriger sur le nombre total de mots dans la référence. Dans le cas de la transcription, un mot à corriger représente alors un mot mal transcrit. Comme cela a été décrit en 3.8.2, le WSR est égal au Word Error Rate (*WER*) [Laurent, 2010;

McCowan *et al.*, 2004] dans le cas où aucune méthode d'aide à la transcription n'est présente. L'analyse du WSR avec le WER permet donc d'obtenir une estimation approximative de la différence de coût d'interaction humain entre la présence et l'absence d'une aide à la transcription.

KSR [Laurent, 2010; Laurent *et al.*, 2011; Trost *et al.*, 2005; Wandmacher et Antoine, 2007; Wood et Lewis, 1996]. Cette mesure, ayant été également adaptée dans le domaine de la traduction assistée (voir 3.8.2), a été adaptée au domaine de la transcription de la parole assistée. Dans Laurent *et al.* [2011], les auteurs proposent la formule suivante :

$$KSR = (1 - \frac{k_p}{k_a}) \times 100, \tag{3.1}$$

où k_p exprime le nombre d'actions réalisées par l'utilisateur pour corriger l'hypothèse d'un système de reconnaissance automatique de la parole et k_a le nombre d'actions qui auraient été nécessaires en partant d'une hypothèse ne contenant aucun mot.

3.9.3 Systèmes assistés dans la littérature

Dans cette partie, nous présentons quelques systèmes de transcription assistés.

Dans Toselli *et al.* [2011a], les auteurs proposent et combinent deux méthodes prenant en compte les interactions humaines pour améliorer la transcription hypothèse. Tout d'abord, la première méthode consiste à utiliser un graphe de mots. Ce dernier peut être vu comme un résultat annexe du processus de décodage de la parole. En effet, il contient les meilleures probabilités des modèles linguistiques et acoustiques pour chaque partie de l'hypothèse [Ney *et al.*, 1997; Ortmanns *et al.*, 1997]. Dans un graphe de mots, chaque nœud représente la position temporelle dans l'enregistrement à transcrire et chaque arc représente un mot associé avec une probabilité. À partir d'un graphe de mots et lorsque le préfixe change, les auteurs proposent d'appliquer ce préfixe sur le graphe de mots. Ceci permet d'obtenir un ensemble de nœuds qui se rapproche de la meilleure segmentation du signal. À partir de cet ensemble de nœuds, un système cherche le meilleur chemin afin de proposer à l'utilisateur une nouvelle hypothèse pour le suffixe. La seconde méthode nécessite d'avoir un préfixe, une représentation de plusieurs phrases avec leur probabilité (un graphe de mots) et d'appliquer ce préfixe sur cette représentation. Elle requiert également de pouvoir calculer le coût total de générer une hypothèse donnée à partir d'une autre en utilisant trois opérations : substituer, supprimer et insérer un mot, chaque opération a un coût associé choisi en fonction de l'objectif. À partir de ces pré-requis, cette méthode consiste alors à calculer le coût total de chaque nouvelle hypothèse possible dans le graphe de mots à partir de l'hypothèse avant l'intervention humaine. Lorsque tous les coûts ont été évalués, une

recherche semblable à celle du décodage de Viterbi est appliquée sur le graphe de mots afin de proposer à l'utilisateur le suffixe le moins coûteux. La combinaison de ces deux méthodes permet de réduire les interactions humaines. En effet, par rapport à une simple correction après un décodage d'un système de transcription de la parole entièrement automatique, le WSR passe de 22,9% à 19,3% pour le corpus Xerox [Cubel *et al.*, 2004].

Dans Laurent *et al.* [2011] et Laurent [2010], les auteurs présentent une méthode prenant en considération les corrections humaines pour produire une nouvelle transcription sans faire appel à un nouveau décodage du signal audio. Plus précisément, elle consiste à réévaluer la meilleure hypothèse contenue dans un réseau de confusion (*confusion network*) [Mangu *et al.*, 2000] en fonction des corrections effectuées. Ce réseau de confusion est obtenu à partir d'un graphe de mots construit pendant la phase de décodage d'un système de transcription de la parole entièrement automatique. Un réseau de confusion est constitué des mêmes éléments qu'un graphe de mots. La figure 3.4 illustre la transformation d'un graphe de mots en réseau de confusion. La méthode proposée permet de réduire significativement le nombre d'actions nécessaire à la correction d'une sortie d'un système. En effet, testée sur le corpus composé d'enregistrements d'émissions radio françaises ESTER 2 [Galliano *et al.*, 2009], elle a permis de décroître le nombre d'actions avec un KSR passant de 87,7% à 90,3% et un WSR passant de 19,2% à 15,8%.

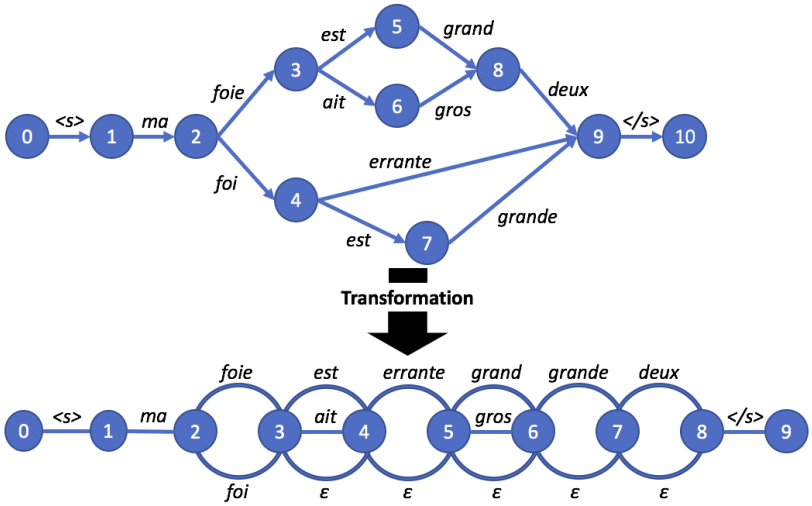


FIGURE 3.4 Exemple d'une transformation d'un graphe de mots en réseau de confusion (les probabilités ne sont pas représentées)

Dans Rodríguez *et al.* [2007], les auteurs présentent une méthode qui relance le décodage après chaque correction humaine. Elle décode une nouvelle hypothèse de reconnaissance de mots en considérant le préfixe. Cette méthode s'appuie sur l'utilisation de deux modèles de langage. Un modèle est utilisé pour forcer le système à décoder à nouveau le préfixe. Ce

modèle va ensuite être utilisé par un autre modèle afin de rechercher le nouveau suffixe. Cette méthode fait décroître le nombre de corrections de 19% pour le corpus EuTrans [Amengual *et al.*, 2000] et de 14% pour le corpus Albayzin Geographic [Verdejo *et al.*, 1998]. Cependant, cette méthode présente un grand risque d'avoir un effet négatif sur le temps de réponse du système.

3.10 Vérification du locuteur assistée

Dans cette partie, nous exposons les mesures d'erreurs utilisées pour évaluer la vérification du locuteur assistée. Nous étudions ensuite des systèmes assistés présents dans la littérature. Tout comme pour le domaine de la transcription de la parole assistée, le nombre d'études concernant la vérification du locuteur assistée reste faible dans la littérature.

3.10.1 Définition de la tâche

En 2010, les meilleurs systèmes de vérification du locuteur entièrement automatique obtenaient des taux d'erreur en dessous de 2% d'*EER* (*Equal Error Rate*) sous certaines conditions [Greenberg *et al.*, 2011a]. Depuis, ces systèmes se sont encore nettement améliorés Kahn [2011]. Bien que cette tâche propose des performances acceptables, la vérification du locuteur assistée par l'humain (*Human Assisted Speaker Recognition - HASR*) a été envisagée dans l'optique d'évaluer la complémentarité qu'il peut exister entre une expertise humaine et le traitement automatique [Greenberg *et al.*, 2011b; Kahn, 2011]. L'idée finale consiste à extraire des informations des interactions d'un système de vérification du locuteur avec un humain afin de proposer de meilleures performances.

3.10.2 Mesures d'erreurs utilisées

Il n'existe pas, à notre connaissance, de mesure d'erreurs pour évaluer le coût d'intervention humain dans le domaine de la vérification du locuteur assistée. Dans Greenberg *et al.* [2010], les auteurs adoptent une approche d'évaluation simple. En effet, ils n'utilisent pas les fausses détections (*False Alarm*) ou les détections manquées (*Miss*) traditionnellement utilisées pour évaluer les systèmes de vérification du locuteur entièrement automatiques. Les auteurs évaluent un système de vérification du locuteur assisté en visualisant le nombre d'acceptations correctes ainsi que le nombre de rejets corrects sur des comparaisons cible (deux enregistrements produits par un même locuteur) et des comparaisons imposteur (deux enregistrements produits par des locuteurs distincts). Ils évaluent également les performances de manière globale grâce à un taux de réussite. Ce taux exprime simplement

le pourcentage de réponses correctes en faisant abstraction du type de comparaison cible et imposteur.

3.10.3 Systèmes assistés dans la littérature

Dans cette partie, nous présentons des systèmes de vérification du locuteur assistés.

Dans Martin et Greenberg [2010], un article décrivant la conférence NIST 2010 sur la vérification de locuteur, une session particulière a été dédiée à la vérification du locuteur assistée par l'humain. Les humains impliqués étaient soit des experts soit des personnes ne travaillant pas dans le domaine audio. Les auteurs notent que, contre toute attente, les systèmes qui ont été proposés à cette session ne surpassent pas les performances des systèmes entièrement automatiques.

Dans Shum *et al.* [2014], les auteurs proposent une méthode de vérification du locuteur assistée utilisant un apprentissage actif. Plus précisément, cet apprentissage interroge un utilisateur avec des questions semblables à "*Est-ce que le locuteur J et I sont une seule et même personne ?*" afin d'obtenir un graphe entre classes de locuteur. À partir de ce graphe, on extrait une mesure ou une fonction de score qui sert alors de système de vérification du locuteur. Par ce procédé, les auteurs arrivent à créer un système de vérification du locuteur obtenant des résultats équivalents à des systèmes états de l'art avec seulement quelques requêtes posées à l'utilisateur. Cependant, pour ce système, les auteurs supposent que l'humain ne fait pas d'erreurs.

3.11 Autres domaines assistés

Dans la littérature, d'autres domaines de recherche hors du contexte de la parole ont étudié des systèmes assistés par l'humain. Dans cette partie, nous présentons quelques-uns de ces domaines.

3.11.1 Transcription d'images de texte assistée

Actuellement, les systèmes à l'état de l'art de reconnaissance de texte écrit à la main et de détection de ligne de texte dans des images sont toujours loin d'obtenir une performance comparable à celle d'un humain [Bluche *et al.*, 2015; Puigcerver, 2017]. Afin d'obtenir une bonne transcription, il est généralement demandé à un humain de corriger les erreurs des systèmes. Des systèmes d'assistance à la transcription d'images de texte (*Computer Assisted Transcription of Text Images - CATTI*) [Toselli *et al.*, 2007, 2010], prenant en

compte des interventions humaines pour améliorer la transcription, semblent alors être une solution adéquate pour réduire l'effort et le temps d'interaction humain total.

Dans Toselli *et al.* [2011b], les auteurs proposent une solution utilisant un graphe de mots qui reprend les idées proposées pour le domaine de la transcription de la parole assistée dans Toselli *et al.* [2011a] (voir 3.9.3). Cette méthode montre une réduction du nombre de mots à corriger et donc du nombre d'interactions homme-machine final. Dans ce même article, les auteurs exposent également une autre solution intéressante qui s'appuie sur une action de pointage (*Pointer Action - PA*). Une action de pointage correspond à n'importe quel dispositif de pointage tel que la souris ou un écran tactile. En utilisant simplement une action de pointage, les auteurs prouvent qu'un utilisateur fournit des informations utiles au système de transcription d'images de texte assisté. En effet, un utilisateur valide indirectement les mots transcrits jusqu'à la position du curseur et signale ainsi au système que le mot suivant est erroné. Le système peut alors prendre les devants et directement proposer une nouvelle suite de mots évitant ainsi des corrections humaines explicites. Si le premier mot de la nouvelle suite ne correspond toujours pas, alors l'humain en donne la correction. Cette autre solution appliquée avec la méthode à base de graphe de mots offre de bien meilleures performances.

3.11.2 Recherche d'image par le contenu assistée

La recherche d'image par le contenu, aussi connu sous le nom de *Content-Based Image Retrieval* (*CBIR*) [Goëau, 2009; Shete *et al.*, 2012], est devenue un domaine de recherche actif depuis les années 90 [Eakins et Graham, 1999; Kato, 1992]. L'objectif de ce domaine est de trouver des images pertinentes à partir d'une requête qui est bien souvent représentée par une image exemple du type d'image que l'utilisateur recherche. La recherche d'image par le contenu est toujours loin d'être résolue [Pasumarthi et Malleswari, 2016; Rani et Jindal, 2018] et est donc un bon candidat pour une version assistée par l'humain.

Dans Toselli *et al.* [2011c], les auteurs proposent une nouvelle méthode pour améliorer la pertinence des images requêtées. À partir des images suggérées par le système en fonction de la requête utilisateur, l'humain indique aux systèmes celles qui sont pertinentes. Ce feedback permet au système de recherche d'image de fournir de meilleures suggestions d'images à chaque intervention humaine. L'amélioration s'appuie sur l'utilisation d'un modèle probabiliste déduit à partir du feedback et des scores de similarité entre les images. La méthode proposée par ce papier montre que la pertinence des images s'améliore rapidement après quelques interventions humaines.

3.12 Bilan

Dans ce chapitre, nous avons exposé les besoins des systèmes assistés par l'humain. Nous avons également vu comment un humain est intégré dans un système assisté et par quels moyens il pouvait en améliorer les performances.

Pour ce qui est du cœur de la thèse, la SRL assistée, nous avons évoqué son principe ainsi que son absence dans la littérature. Afin de faire émerger d'éventuelles pistes pour la SRL assistée, nous avons présenté d'autres domaines qui ont déjà étudié des systèmes assistés par l'humain. En nous appuyant sur ce qui a été vu dans ce chapitre et dans le chapitre précédent, nous présentons dans la suite de ce manuscrit notre contribution à la SRL assistée.

Chapitre 4

Protocole d'évaluation des interactions humaines pour la SRL

Dans ce chapitre, nous commençons par nous intéresser à l'estimation du coût d'intervention humain dans le cadre de la SRL. Ensuite, nous étudions l'automatisation des corrections humaines. Enfin, nous nous intéressons à la correction humaine de la sortie d'un système automatique.

4.1 Estimation du coût d'intervention humain

Dans cette partie, nous présentons une mesure évaluant les interactions humaines pour la correction d'un système automatique. Nous déterminons les actions humaines permettant la correction d'une SRL avant de définir le coût d'une action. Ensuite, nous présentons un protocole d'expérimentation des coûts des actions. Enfin, nous terminons cette partie par une mesure du coût de chaque action humaine.

4.1.1 L'intérêt de définir une nouvelle mesure

La mesure classique d'évaluation de la qualité d'une SRL est le DER [NIST, 2003], présenté en 2.8.2. Cette mesure n'est pas pertinente lorsque l'on veut estimer le coût d'une correction faite par un humain. Elle n'est donc pas adaptée pour évaluer un système de SRL intégrant des interactions homme-machine. À notre connaissance, aucune mesure n'intègre cette estimation qui permet de comparer et d'optimiser des systèmes assistés.

Comme nous l'avons vu dans le chapitre précédent, quelques articles [Barrachina *et al.*, 2009; Budnik *et al.*, 2014; Laurent *et al.*, 2011; Rodríguez *et al.*, 2007; Toselli *et al.*, 2011b] proposent des systèmes assistés par l'humain. Cependant dans ces articles, les auteurs

supposent que chaque type de correction possède un coût identique. Or, chaque correction nécessite des actions spécifiques qui requièrent différents efforts. Par exemple, il est plus facile de choisir un label dans une liste de locuteurs potentiels que de créer un nouveau label qui nécessite de vérifier la présence d'un nouveau locuteur parmi les locuteurs déjà identifiés avant de saisir son label. Ainsi, les conclusions de ces articles sur l'efficacité de leurs systèmes assistés sont biaisées.

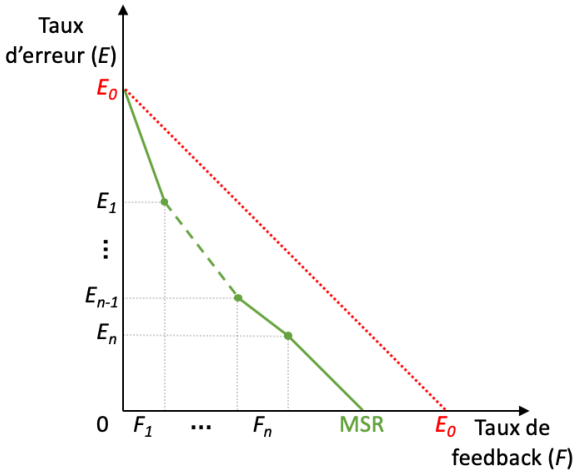


FIGURE 4.1 Illustration de l'estimation du MSR. E_0 le taux d'erreur initial du système, F_n la quantité d'information fournie par le feedback n et E_n le taux d'erreur après F_n. $MSR = E_n + \sum_1^n F_n$. Figure extraite de Geoffrois [2016]

Pour évaluer un système assisté, il est également possible d'évaluer la capacité du système à apprendre à partir des interventions humaines. Dans Geoffrois [2016], l'auteur présente une mesure pour évaluer cette capacité, la mesure *Minimal Supervision Rate* (*MSR*). Dans cet article, l'auteur propose de considérer les erreurs résiduelles d'un système assisté après le dernier feedback comme un ensemble de corrections restantes pour obtenir une sortie de système correcte. La mesure MSR reflète ces erreurs résiduelles. Une illustration de cette mesure est donnée dans la figure 4.1. Cette mesure intègre toutes les corrections d'erreurs dans un facteur de mérite et fournit une mesure. La mesure MSR dépend de la tâche étudiée et n'a pas de formule propre. La figure 4.1 illustre la mesure MSR pour évaluer un système automatique en utilisant un taux d'erreur. Sur cette figure, la courbe rouge correspond au taux d'erreur pour un système automatique. La courbe verte correspond au MSR et représente le taux d'erreur pour une version assistée du système. De manière générale, cette mesure peut être vue comme l'effort minimal requis d'un humain pour corriger toutes les erreurs du système par interactions successives avec ce système.

Cette mesure ne s'intéresse pas à l'investissement physique et mental du correcteur. Ainsi, un système assisté peut présenter un très bon score MSR pour quelques interventions

humaines alors que l'effort et le temps d'investissement humain pour les produire sont considérables. L'INA souhaite réduire l'effort et le temps d'investissement humain pour produire des documentations d'archives plus rapidement et à moindre coût. La mesure MSR, bien qu'intéressante, n'est pas adaptée à notre contexte.

4.1.2 HCIQ : Mesure pour le coût d'intervention humain

L'évaluation des interactions humaines avec un système de SRL assisté nécessite d'être reproductible et objective [Geoffrois, 2016]. Nous proposons une nouvelle mesure inspirée de la mesure Keystroke Saving Rate (KSR) utilisée dans la prédiction de mots pour des personnes ayant des difficultés de communication (voir 3.8.2). Nous l'appelons Human-Computer Interaction Quantity ($HCIQ$). Nous avons introduite cette mesure dans Broux *et al.* [2018b]. Le HCIQ peut être calculé aussi bien pour des systèmes assistés que pour des systèmes sans assistance où seul un humain corrige la SRL. De plus, tout comme le DER, le HCIQ peut être calculé pour chaque enregistrement ou pour une collection d'enregistrements audio/vidéo. Elle est définie par la formule :

$$HCIQ = \sum_{i=1}^{K} w_i n_i, \tag{4.1}$$

où i correspond à une action spécifique dans l'interface, w_i son coût associé ou son poids relatif, n_i le nombre d'occurrences de cette action et K le nombre total d'actions répertoriées. Moins le HCIQ est grand, moins l'humain doit fournir d'effort pour annoter (ou corriger) un document.

La mesure HCIQ, sous cette forme, ne permet pas de comparer des corpus différents. Afin de résoudre ce problème, nous proposons la formule suivante :

$$HCIQ_n = \frac{HCIQ}{d}, \tag{4.2}$$

où $HCIQ$ est normalisé par la durée du corpus d sur lequel le $HCIQ$ est évalué. Cette normalisation permet de s'affranchir de la spécificité du corpus (nombre de locuteurs, présence de parole spontanée, etc.). La mesure HCIQ_n est un ratio du nombre de corrections à faire pour une unité de temps.

Le HCIQ se rapproche de la mesure proposée dans Guillaumin *et al.* [2009] où les auteurs proposent une mesure reflétant l'effort nécessaire pour la correction d'images de visage mal labellisées. Dans cet article, l'humain a deux boutons à sa disposition : un bouton pour assigner un seul label à toutes les images dans une classe et un autre pour assigner un label à une image. À partir de ces boutons, la stratégie de correction la plus efficace est de

labelliser en premier la classe avec le nom de la personne la plus fréquente et ensuite de corriger les erreurs. Pour estimer le coût d'effort humain, les auteurs proposent la formule suivante :

$$Coût_{Humain} = \sum_{c=1}^{C} 1 + (N_c - \max_i(\{n_{c,i}\})), \tag{4.3}$$

où C représente le nombre de classes à vérifier, N_c le nombre d'images de la classe c et $n_{c,i}$ le nombre d'images de visage de la personne i dans la classe c. Comparée à la mesure HCIQ, cette mesure est moins précise. En effet, elle ne prend pas en considération l'effort lié à chaque interaction humaine. Par exemple, assigner un label à une image nécessite au préalable de sélectionner l'image mal labellisée. Cet effort n'est pas représenté dans la mesure proposée par l'article.

4.1.3 Choix des actions de corrections

Pour déterminer les actions mises à la disposition de l'humain pour la correction, nous nous appuyons sur un logiciel de référence dans le domaine de la transcription de la parole : *Transcriber* [Barras *et al.*, 2001]. Ce logiciel, autorisant le clavier et la souris comme moyens d'interaction, permet de découper un signal audio en segments (généralement des groupes de souffle). Chaque segment est obligatoirement lié à un label représentant un locuteur. Ce label est enrichi par des informations telles que le sexe du locuteur ou la langue dans laquelle il s'exprime. Des illustrations de *Transcriber* sont disponibles en annexe, dans la section C.1.

Dans *Transcriber*, les actions possibles modifiant une segmentation sont les suivantes : "*Créer une frontière*", "*Supprimer une frontière*" et "*Déplacer une frontière*". L'action "*Créer une frontière*" permet d'ajouter une frontière en coupant en deux un segment, l'action "*Supprimer une frontière*" fusionne deux segments consécutifs en gardant le label locuteur du segment apparu en premier dans le signal audio et l'action "*Déplacer une frontière*" déplace la frontière d'un segment mal positionnée. Quant aux actions modifiant le regroupement, *Transcriber* propose les actions "*Créer un label locuteur*" et "*Changer de label locuteur*". La première action permet de créer un nouveau label puis de l'associer au segment. La seconde action, quant à elle, permet de changer le label associé à un segment.

En plus des actions de segmentation et de regroupement, *Transcriber* propose une action "*Mettre à jour un label locuteur*" qui modifie les informations liées à un label ainsi que le label lui-même. Plus précisément, cette action permet de modifier les informations entrées lors de la création du label telles que le sexe du locuteur ou l'écriture du label.

Elle est à différencier de l'action "*Changer de label locuteur*" qui, elle, ne fait que changer l'association d'un label à un segment.

Dans ce logiciel, chaque partie du signal doit appartenir à un segment. Ainsi, aucune action permettant d'insérer ou d'enlever un segment n'est prévue par *Transcriber*. Ce dernier propose également des labels spécifiques pour annoter les zones sans paroles ou non transcrites.

Les actions proposées par *Transcriber* sont suffisantes pour corriger une SRL. Dans le reste du manuscrit, nous étudions et utilisons donc les actions suivantes :

- "*Créer un label locuteur*" ;
- "*Changer le label locuteur*" ;
- "*Créer une frontière*" ;
- "*Déplacer une frontière*" ;
- "*Supprimer une frontière*".

Comme nous nous intéressons uniquement aux actions de segmentation et de regroupement en locuteurs, l'action "*Mettre à jour un label locuteur*" est ignorée.

4.1.4 Définition d'un coût d'une action

Quantifier précisément l'effort d'une action est une tâche complexe. On suppose que le temps de réalisation d'une action reflète l'effort associé. Donc, plus une action demande un effort, qu'il soit physique ou mental, plus elle nécessite du temps à sa réalisation. Dans la suite du manuscrit, nous avons donc décidé de représenter le coût d'une action par sa durée de réalisation.

4.1.5 Protocole d'expérimentation des coûts des actions

L'expérimentation précise des actions peut être divisée en trois grandes étapes : l'estimation de chaque interaction de *Transcriber*, l'identification des actions et l'estimation des temps associés.

4.1.5.1 Estimation de chaque interaction

L'historique des interactions clavier et souris dans *Transcriber* permet de déterminer indirectement la suite des actions et ainsi de calculer précisément les temps de chacune d'elles. Pour construire cet historique, il a fallu modifier le code de *Transcriber*. Une interaction dans l'historique, c'est-à-dire un clic de souris ou une frappe de touche clavier, comprend trois informations : le temps d'apparition de l'interaction, le nom du module concerné par l'interaction et un commentaire (figure 4.2). Le module identifie un élément

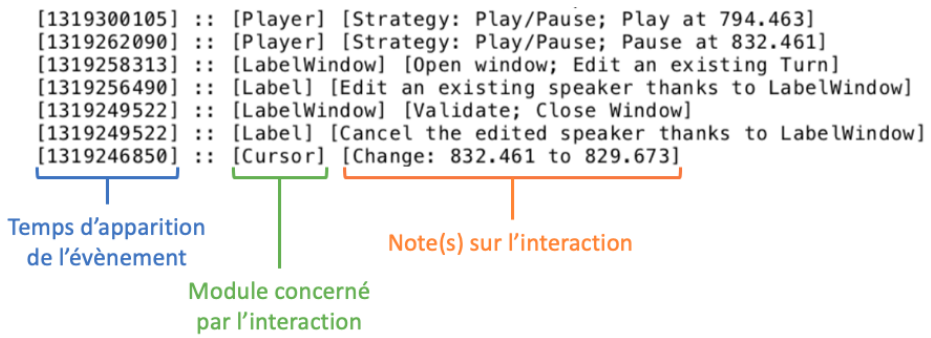


FIGURE 4.2 Exemple de traces dans un fichier log

de l'interface utilisateur tandis que le commentaire donne des indications précises sur l'évènement en cours.

4.1.5.2 Identification des actions à partir des interactions

La conversion des interactions en actions dans *Transcriber* est difficile à obtenir automatiquement. En effet, certaines actions peuvent être réalisées par des interactions identiques, mais effectuées dans un ordre différent [1]. En outre, l'annotateur peut faire des erreurs ou peut suspendre la session d'annotation pour une durée plus ou moins longue. Nous ne souhaitons pas que ces deux cas soient associées à des actions. Pour les éliminer et identifier correctement les actions, la vidéo de la séance d'annotation (un enregistrement de l'écran de l'utilisateur) est manuellement segmentée en une ou plusieurs actions grâce à l'historique. Chaque segment créé correspond donc précisément à la durée mesurée d'une action. Pour segmenter cette vidéo de capture de la session d'annotation faite avec *Transcriber*, nous avons utilisé le logiciel *ELAN* [Wittenburg *et al.*, 2006], un outil dédié à la création d'annotations complexes sur des ressources vidéo et audio. Des illustrations du logiciel *ELAN* sont disponibles en annexe, dans la section C.2.

Lors de cette annotation manuelle, nous imposons de segmenter également les temps où l'humain écoute pour la première fois le fichier audio. Il est à préciser que ces temps d'écoute sont à différencier d'une éventuelle pré-étape où l'humain écoute entièrement le fichier audio. Les moments d'écoute entre deux corrections ne doivent pas être comptabilisés comme des corrections.

L'annotation de la vidéo de la séance d'annotation au moyen de l'historique et du logiciel *ELAN* est une tâche aisée, mais fastidieuse.

1. Dans *Transcriber*, il existe plusieurs suites d'évènements aboutissant à une même correction.

4.1.5.3 Estimation des coûts associés aux actions identifiées

Les durées des actions sont calculées à partir du temps de la première interaction de l'action et du temps de la dernière interaction de l'action. Pour chaque action, nous estimons sa durée en prenant la moyenne de ces temps.

4.1.5.4 Variabilité des coûts

Selon Arora *et al.* [2009], trois paramètres jouent un rôle important dans l'estimation du coût d'annotation : l'annotateur, l'interface et le document annoté.

L'interface

L'estimation du temps pris par une action varie en fonction de l'interface utilisée. Plus celle-ci est ergonomique, plus un utilisateur annote rapidement et plus les temps d'annotation diminuent. Le logiciel *Transcriber* a l'avantage d'offrir une bonne ergonomie d'après les annotateurs consultés. Dans *Transcriber*, l'humain segmente généralement le signal en plaçant les frontières sur les changements de locuteur, les silences ou les inspirations. Dans notre cadre applicatif, seuls les changements de locuteur sont utiles et seront donc annotés.

L'annotateur

L'annotateur joue également un rôle dans l'estimation du coût d'annotation. Une personne réalisant souvent des annotations travaillera plus rapidement qu'une personne n'en ayant jamais fait. De même, des personnes utilisant régulièrement un logiciel d'annotation corrigeront plus rapidement que des personnes qui viennent de le découvrir. En outre, des personnes en bonne santé, que ce soit psychique ou physique, corrigeront plus rapidement qu'une personne fatiguée ou malade. Ainsi les temps d'annotation varient en fonction de l'expérience du correcteur, aussi bien dans le domaine de l'annotation que dans la connaissance du logiciel d'annotation utilisé, mais également en fonction de sa santé. Il existe deux types de stratégie possibles pour gérer la variabilité liée à l'annotateur : soit prendre la moyenne des temps d'annotateurs ayant la même expérience, soit prendre celle de plusieurs annotateurs ayant diverses expériences. Nous supposons que les annotateurs n'ont aucun problème de santé pour le travail demandé. La première stratégie permet d'obtenir des temps spécifiques pouvant être utiles en fonction de l'objectif choisi. La seconde offre des temps globaux recouvrant ainsi différents profils d'annotateur avec des niveaux d'expérience variables. Ces temps sont malheureusement contraints par les profils d'annotateur que nous avons à disposition.

Le document annoté

Concernant l'annotation des enregistrements audio, deux points sont à retenir : la qualité du signal et le degré de spontanéité. Plus la qualité du signal est basse, plus il est difficile d'annoter. De même, plus le degré de spontanéité est élevé, plus il est compliqué de déterminer qui parle à quel moment et donc plus il est difficile d'annoter. La qualité du signal et le degré de spontanéité ont donc des conséquences sur le temps d'annotation humain [Bazillon *et al.*, 2008]. Par conséquent, deux stratégies peuvent être envisageables pour gérer la variabilité liée au document annoté : soit annoter des enregistrements variés, soit annoter des enregistrements spécifiques. La première stratégie permet d'avoir des temps d'annotation moyennés. La seconde stratégie, qui se concentre plutôt sur un type d'enregistrement sera plus spécifique à une application choisie. Nous adoptons la seconde stratégie. Les enregistrements à annoter sont de bonne qualité audio avec un degré de spontanéité faible. Ce choix a été fait afin que les temps d'annotation soient plus faciles à mesurer et car ce type d'enregistrement correspond à la majorité des émissions intéressant l'INA.

4.1.6 Mesure du coût des actions

4.1.6.1 Corpus

Pour mesurer la durée des actions, nous avons utilisé des enregistrements télévisuels de la campagne d'évaluation de 2013 du défi ANR-REPERE (REPERE 1), présenté dans la partie 2.8.1.

Le corpus a été traité par le système de SRL de référence sans la dernière étape (regroupement i-vecteur/ILP) afin d'avoir un plus grand nombre d'erreurs de regroupement en locuteurs. Nous avons corrigé uniquement certaines zones des enregistrements. Ces zones respectent les choix pris en 4.1.5.4, soit des zones où l'audio est de bonne qualité et le degré de spontanéité est faible.

Certaines erreurs sont bien moins présentes que d'autres dans la sortie du système de SRL. Afin d'avoir suffisamment de valeurs pour nos statistiques, certaines erreurs ont dû être simulées. Plus précisément, nous avons simulé une quinzaine d'actions "*Créer un label locuteur*" et une dizaine d'actions "*Changer de label locuteur*".

4.1.6.2 Évaluation

Les durées des actions de *Transcriber* sont données dans le tableau 4.1. Elles ont été obtenues à partir de 20 minutes des données de REPERE 1. Ces temps proviennent d'un

Action	Occurrence	Moyenne (s)	écart-type (s)
Créer un label locuteur	28	12,7	6,0
Changer de label locuteur	32	7,6	3,8
Créer une frontière	38	12,0	7,6
Déplacer une frontière	42	14,6	8,2
Supprimer une frontière	46	5,1	2,3

TABLE 4.1 Durée des actions - 20 minutes de données de REPERE 1

seul annotateur. Pour 20 minutes de données, la correction des erreurs au moyen du logiciel *Transcriber* a nécessité 49 minutes. La segmentation manuelle de la vidéo de la séance d'annotation avec l'aide de l'historique, quant à elle, a nécessité 4h26. Comme on peut le remarquer, la mesure des coûts est une tâche nécessitant un temps assez important.

La correction d'une erreur est construite sur trois phases :

- détecter la présence d'une erreur ;
- localiser l'erreur ;
- corriger l'erreur.

Chaque durée d'action dans le tableau 4.1 représente la somme des temps de ces trois phases.

L'action la plus coûteuse en temps est l'action "*Déplacer une frontière*" avec une durée moyenne de 14,6 secondes. Cette action requiert de regarder ainsi que d'écouter le signal pour trouver le changement de locuteur et de déplacer correctement la frontière existante à cet endroit. Généralement, il est nécessaire d'écouter plusieurs fois le signal pour détecter le point de rupture. Après l'action "*Déplacer une frontière*", les actions les plus chronophages sont "*Créer un label locuteur*" et "*Créer une frontière*" avec une moyenne d'environ 12 à 13 secondes. La première action nécessite d'entrer un label pour le locuteur (et éventuellement d'autres méta-données du locuteur), tandis que la seconde action requiert d'écouter et regarder le signal pour détecter le changement de locuteur. Tout comme le déplacement de frontière, il est généralement nécessaire d'écouter à plusieurs reprises le signal pour trouver le changement de locuteur. Comparée au déplacement d'une frontière, une création d'une frontière requiert moins d'effort. L'action nommée "*Changer de label locuteur*" a une moyenne de 7,6 secondes. Grâce à une fenêtre contextuelle, elle consiste à sélectionner le label de locuteur correct dans une liste déroulante. Chercher un label dans une liste déroulante prend moins d'effort que la création de frontières. L'action la plus rapide est "*Supprimer une frontière*". Elle nécessite d'arrêter l'écoute quand une fausse frontière est détectée et de la supprimer par une combinaison de touches claviers.

Dans la suite de ce manuscrit, nous utilisons ces temps pour le calcul du HCIQ. Cependant, le HCIQ avec ces temps est adapté à des corpus avec un faible degré de

spontanéité. Les corpus que nous utilisons pour nos expériences (ESTER 1, ESTER 2, ETAPE et REPERE) ont des degrés de spontanéité variables qui pourraient biaiser en partie nos résultats en minimisant les coûts de correction.

4.1.7 Difficultés à déterminer le jeu d'actions

Plusieurs pistes ont été explorées avant d'adopter la mesure précise des durées.

La première piste explorée ne s'appuyait sur aucun logiciel d'annotation particulier, mais sur une estimation grossière des durées à partir de l'expérience et du ressenti des annotateurs. L'avantage de ce choix est d'avoir un ensemble d'actions unitaires et indépendantes des applications. La durée de chaque action a été choisie empiriquement en fixant un coût exprimé en nombre d'unités élémentaires. En nous inspirant des travaux en transcription de la parole assistée décrite dans Laurent *et al.* [2011], nous avons choisi de prendre "*l'appui sur une touche*" et "*le clic sur la souris*" comme unités élémentaires. Plus une action demande d'investissement, plus elle a un nombre d'unités élémentaires important. Les valeurs obtenues ont été jugées trop approximatives lors de nos évaluations préliminaires. En effet, certaines actions n'ont pas pu être quantifiées précisément par rapport à une unité élémentaire. Par exemple en prenant comme unité élémentaire "*l'appui sur une touche*", il est alors très difficile de quantifier des interactions telles que "*Rechercher un évènement dans un signal audio*".

La seconde piste explorée a été de modifier *Transcriber* pour mesurer directement les temps de chaque action. Cette méthode a l'avantage de calculer plus précisément les durées des actions que la méthode utilisant les unités élémentaires. Cependant après avoir modifié le code *Transcriber*, l'analyse des historiques a montré certaines limites : il est difficile de savoir ce qui est réellement compté à partir des temps puisqu'une action englobe plusieurs interactions homme-machine. Pour une action donnée, par exemple, il est possible que l'humain ait répété inutilement des interactions, commis de mauvaises interactions, voire pris une pause. Ces éléments sont alors pris en compte dans le temps de l'action. Afin de remédier à ces problèmes, il est donc nécessaire d'expliquer au préalable le fonctionnement et les limites des mesures à l'annotateur.

Le tableau 4.2 et le tableau 4.3 présentent respectivement les résultats obtenus pour des actions indépendantes de tout logiciel utilisant des unités élémentaires à partir du calcul direct des durées et à partir de l'historique. L'explication détaillée des résultats du tableau 4.2 est donnée dans l'annexe B. Le tableau 4.3 a été obtenu en fonction des choix pris en 4.1.5.4, soit à partir de zones audio de bonne qualité audio avec un degré de spontanéité faible. Ces temps proviennent de deux annotateurs, un linguiste spécialiste de l'annotation et un informaticien novice en annotation, corrigeant chacun environ 10 minutes du corpus

REPERE. Le traitement appliqué sur ce corpus avant d'effectuer la correction est le même que celui introduit dans la partie précédente.

Action	Coût en unité élémentaire
Créer un label locuteur	12
Changer de label locuteur	6
Créer une frontière	36
Supprimer une frontière	2
Créer un segment	Non applicable
Supprimer un segment	Non applicable

TABLE 4.2 Estimation des actions indépendantes de tout logiciel en unité élémentaire à partir de l'expérience avec le logiciel *Transcriber*

Action	Moyenne (s)
Créer un label locuteur	13,8
Changer de label locuteur	5,6
Créer une frontière	22,3
Supprimer une frontière	2,0
Créer un segment	Non applicable
Supprimer un segment	Non applicable

TABLE 4.3 Estimation des actions au moyen du logiciel *Transcriber* traçant directement les temps de chaque action

En considérant qu'une unité élémentaire dure une seconde, on peut constater que les coûts des corrections de label des tableaux 4.2 et 4.3 sont très proches des coûts du tableau 4.1 de la partie précédente. En effet, pour les tableaux 4.1, 4.2 et 4.3, l'action "*Créer un label locuteur*" coûte respectivement 12,7 s, 12 unités élémentaires et 13,8 s. L'action "*Changer le label locuteur*", quant à elle, coûte respectivement 7,6 s, 6 unités élémentaires et 5,6 s. Concernant les corrections de frontières, les coûts sont très différents pour les trois tableaux, à l'exception des tableaux 4.2 et 4.3 pour l'action "*Supprimer une frontière*". En effet, pour les tableaux 4.1, 4.2 et 4.3, l'action "*Créer une frontière*" coûte respectivement 12,0 s, 36 unités élémentaires et 22,3 s. L'action "*Supprimer une frontière*", quant à elle, coûte respectivement 5,1 s, 2 unités élémentaires et 2,0 s. On remarque également que les tableaux 4.2 et 4.3 considèrent que "*Créer une frontière*" est l'action la plus coûteuse avec un coût deux à trois fois supérieur à celui de la seconde action la plus coûteuse, l'action "*Créer un label locuteur*". Dans le tableau 4.1, ces deux actions ont un coût quasi identique. Ainsi, les résultats du tableau 4.1 confirment que les valeurs choisies empiriquement et reportées dans les tableaux 4.2 et 4.3 ont été mal estimées.

4.2 Automatisation des corrections humaines

Dans cette partie, nous présentons un automate simulant les corrections humaines. Cette partie décrit l'intérêt d'avoir un automate, les limitations que nous nous imposons dans sa conception ainsi que le jeu d'actions utilisé. Elle détaille également la structure et le fonctionnement de l'automate.

4.2.1 L'apport d'un automate à disposition

Afin de déterminer comment aider un annotateur humain, il est nécessaire d'étudier plusieurs cas de corrections humaines sur divers exemples. Cependant, une correction humaine prend du temps [Bazillon *et al.*, 2008] et nécessite d'avoir des correcteurs humains à sa disposition. Nous n'avons malheureusement pas suffisamment de temps, ni des correcteurs humains disponibles à notre guise. L'alternative retenue est de créer un automate simulant les corrections humaines et plus précisément celles qui concernent la segmentation et le regroupement en locuteurs. Avec un automate, l'obtention d'une correction est rapide.

Un automate peut également servir de base pour développer des systèmes assistés effectuant des adaptations automatiques sur une SRL en fonction des corrections humaines.

4.2.2 Limitations de l'automate

Afin d'avoir un automate simple et robuste, plusieurs règles ont été définies. Il a été décidé qu'un automate simule un annotateur qui :

1. ne fait aucune erreur ;
2. corrige l'émission du début à la fin dans l'ordre temporel afin de valider l'annotation automatique faite a posteriori ;
3. ne corrige que le tour de parole courant, qui vient d'être écouté.

La première règle permet d'éviter la modélisation aléatoire et complexe des erreurs pouvant être commises par un humain. De plus, cette simplification permet d'avoir un document sans erreurs à la fin du processus de correction et ainsi un DER à 0%. La seconde règle, quant à elle, suppose que corriger à partir du début jusqu'à la fin aide l'annotateur à mieux comprendre et à améliorer la correction. Cette règle et la dernière sont des conditions expérimentales choisies pour faciliter la création de l'automate et peuvent être remises en cause.

4.2.3 L'intérêt d'une série de corrections humaines déterministe

Afin de faciliter la création d'un automate simulant un annotateur, les séries d'actions humaines seront déterministes. Nous voulons qu'aucune action ne puisse être substituée par un ensemble d'actions fournissant la même correction.

Une des actions de *Transcriber* ne remplit pas ce critère. En effet, l'action "*Déplacer une frontière*" peut être remplacée par les deux actions suivantes : "*Supprimer une frontière*" et "*Créer une frontière*". En s'appuyant sur le tableau 4.1, on constate que l'addition des temps des actions "*Créer une frontière*" et "*Supprimer une frontière*" est plus chronophage de 2,5 secondes que l'action "*Déplacer une frontière*". Cette différence de temps peut principalement s'expliquer par un nombre d'interactions du clavier et de la souris moins important pour l'action "*Déplacer une frontière*", mais également parce que cette dernière, contrairement à "*Créer une frontière*", ne requiert pas de choisir un label locuteur. En effet, lorsqu'on crée une frontière, il faut définir quel locuteur a pris la parole au moyen d'une fenêtre contextuelle.

Cependant, lorsqu'une frontière de l'hypothèse est mal positionnée à plus ou moins 2 secondes autour de la frontière de référence, le label du locuteur après la frontière est incorrect dans 73% des cas. Par conséquent, lorsqu'on utilise l'action "*Déplacer une frontière*", il faut ensuite utiliser l'action "*Créer un label locuteur*" ou l'action "*Changer le label locuteur*". Ces combinaisons d'actions sont plus chronophages que les actions "*Créer une frontière*" et "*Supprimer une frontière*" réunies, qui aboutissent au même résultat. En moyenne, il est donc plus rentable de supprimer puis créer une frontière.

Nous avons gardé deux actions pour modifier les frontières des segments et deux actions pour modifier les labels affectant le regroupement en locuteurs. En combinant ces actions, nous pouvons décrire toutes les corrections d'une manière unique. Les quatre actions sélectionnées utilisées pour l'automate sont :

- "*Créer une frontière*" ;
- "*Supprimer une frontière*" ;
- "*Créer un label locuteur*" ;
- "*Changer le label locuteur*".

4.2.4 Structure de l'automate

La simulation de l'annotateur repose sur deux types d'information pour déterminer si une correction est requise au temps t :

1. La correspondance entre le segment de référence (vérité terrain) et le segment de l'hypothèse ;

2. La correspondance entre les labels de locuteur de la référence et l'hypothèse qui minimise le DER.

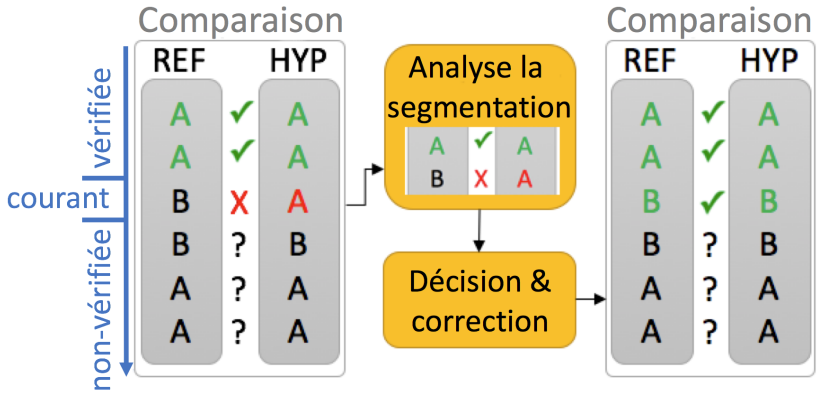


FIGURE 4.3 Illustration d'un annotateur simulé

En cas de différence au niveau de la segmentation ou du label au temps t entre la référence et l'hypothèse, une correction est nécessaire. L'annotateur simulé corrige en premier les erreurs de segmentation, puis les erreurs de regroupement en locuteurs (figure 4.3).

Après chaque correction, le système peut lancer un système de SRL sur la partie non vérifiée (les segments avec un temps de début $> t$) en prenant en considération les segments déjà vérifiés (les segments avec un temps de fin $\leq t$). Dans le cas d'un système initial sans erreurs de segmentation, la correction du regroupement en locuteurs est facile à mettre en place [Broux *et al.*, 2016], car les segments de l'hypothèse sont identiques aux segments de la référence.

La correction de la segmentation est plus difficile et l'annotateur simulé a besoin de prendre en considération la précision des frontières de la référence. Une tolérance de plus ou moins 250 ms est généralement appliquée aux frontières des segments de référence pour le calcul du DER. Nous appliquons la même tolérance pour éviter les nombreuses micro-corrections, généralement inutiles. Ainsi, avant de procéder à l'évaluation de la correspondance des segments et des labels, toute frontière de l'hypothèse appartenant à une zone de tolérance est déplacée à coût nul afin d'être alignée avec la frontière de la référence. Dans le cas où un segment de référence serait recouvert par la tolérance, celui-ci n'est pas évalué par l'automate. L'automate n'évalue pas non plus les segments de l'hypothèse recouverts par la tolérance aux frontières de la référence. Le tableau 4.4 illustre le nombre de corrections pour le corpus REPERE à la sortie du système de référence avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible HCIQ en fonction de différentes tolérances aux frontières (δ optimal). Lorsqu'une tolérance de 0 ms est appliquée,

on constate que les différentes corrections augmentent de manière non négligeable par rapport à une tolérance de 250 ms.

Tolérance (ms)	0	250
Seuil δ optimal	10	30
Créer un label locuteur	326	295
Changer le label locuteur	755	415
Créer une frontière	1282	814
Supprimer une frontière	709	342
Toutes actions confondues	3072	1866

TABLE 4.4 Nombre d'occurrences des différentes actions pour le corpus REPERE à la sortie du système de référence avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible HCIQ en fonction de la tolérance aux frontières appliquée

Pour obtenir ces résultats, les vérités terrain sont pré-traitées. Chaque zone de parole superposée est représentée par un segment avec un label regroupant tous les noms des intervenants. Ainsi deux zones de parole superposée composées du même ensemble d'intervenants partagent un label commun. Nous avons également fusionné les segments émis par un même locuteur et espacés de moins de 2 secondes dans la référence et l'hypothèse. Ce traitement a été appliqué afin de correspondre aux attentes de l'INA qui est intéressé par un découpage en tours de paroles de locuteurs plutôt qu'une succession de segments de parole énoncés par un même locuteur et espacés de quelques millisecondes. Enfin, nous avons seulement évalué les zones annotées dans le fichier UEM.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
# de locuteurs	467	396	235	326	1424
# de changements de locuteur	1954	1881	3369	1287	8491
# de segments	1991	1911	3398	1318	8618
Moyenne de la durée des segments (s)	17,85	13,50	7,37	8,05	12,91

TABLE 4.5 Quelques statistiques sur la vérité terrain de divers corpus

Le tableau 4.5 offre des informations après pré-traitement sur les vérités terrain des corpus. Pour chaque corpus dans ce tableau, on constate que le nombre de locuteurs [2] est plus élevé que celui du tableau 2.1. Cela s'explique par la manière dont on a considéré la parole superposée et par l'ajout d'un label locuteur pour représenter l'absence de segment dans chaque enregistrement.

Dans la suite de ce manuscrit, nous nommons "système oracle" cet annotateur simulé lorsqu'aucune adaptation automatique n'est effectuée au fur et à mesure des corrections.

2. Nombre de locuteurs du tableau 2.1 : 342, 250, 148, 158 et 898 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

4.3 Correction humaine de la sortie d'un système automatique

Dans cette partie, nous nous intéressons aux corrections de la sortie d'un système automatique de SRL faites par un humain. Tout d'abord, nous expliquons l'intérêt d'évaluer la correction d'un système automatique. Ensuite, nous évaluons la correction humaine sur des données n'ayant subi aucun traitement de SRL. Enfin, nous comparons cette correction avec la correction humaine à diverses étapes du système automatique présenté en 2.9.

4.3.1 L'intérêt d'évaluer la correction d'un système automatique

Avant de vouloir créer des systèmes de SRL assistés, il est crucial de savoir si un système de SRL automatique permet de réduire le temps de correction d'un humain. En effet, si un humain met plus de temps à corriger la sortie d'un système de SRL automatique que d'effectuer entièrement seul la SRL, alors développer des systèmes assistés perd tout intérêt.

À partir du HCIQ, du jeu d'actions et du système oracle que nous avons défini, nous sommes en mesure d'évaluer si un système de SRL automatique permet de réduire le temps de traitement.

4.3.2 Correction sans aide d'un système automatique

Nous évaluons dans cette partie le HCIQ sur des corpus n'ayant subi aucun traitement de SRL. Chaque enregistrement de chaque corpus est représenté par son fichier UEM et est donné en entrée du système oracle.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	1107	947	1762	645	4461
Créer une frontière	1678	1685	2765	1175	7303
Supprimer une frontière	0	0	0	0	0
Toutes actions confondues	3167	2992	4682	2115	12956

TABLE 4.6 Nombre d'occurrences des différentes actions du système oracle pour divers corpus n'ayant subi aucun traitement de SRL

Les tableaux 4.6 et 4.7 illustrent respectivement le nombre d'occurrences et la durée estimée des différentes actions du système oracle pour divers corpus. Le tableau 4.8, quant à lui, propose des informations complémentaires sur les mêmes zones évaluées.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	140,2	119,9	223,2	81,7	565,0
Créer une frontière	335,6	337,0	553,0	235,0	1460,6
Supprimer une frontière	0	0	0	0	0
Toutes actions confondues (HCIQ)	556,7	533,1	809,0	379,1	2277,9

TABLE 4.7 Durée estimée en minutes des différentes actions du système oracle pour divers corpus n'ayant subi aucun traitement de SRL. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
DER	69,17	61,71	68,01	65,90	66,55
$DER_{false\,alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\,speaker}$	0,00	0,00	0,00	0,00	0,00
$DER_{confusion}$	69,17	61,71	68,01	65,90	66,55
$HCIQ_n$ sans système automatique	0,94	1,24	1,93	2,14	1,41

TABLE 4.8 Mesures appliquées sur divers corpus n'ayant subi aucun traitement de SRL

Dans les tableaux 4.6 et 4.7, on remarque que le nombre d'actions "*Supprimer une frontière*" ainsi que la durée associée est nulle, quel que soit le corpus. Ceci s'explique par le fait qu'il n'y a pas eu de traitement de SRL. Aucune frontière n'a été créée dans les hypothèses et donc aucune frontière n'est à vérifier.

Dans le tableau 4.8, les fortes valeurs du DER s'expliquent par l'utilisation d'un seul segment pour chaque enregistrement. À partir du $HCIQ_n$ présent dans ce même tableau, on peut constater que le corpus REPERE est un des corpus qui requiert le plus de corrections pour une unité de temps puisqu'il nécessite en moyenne 2,14 minutes de corrections humaines qui s'ajoutent au temps d'écoute pour 1 minute de signal audio. En outre, il montre que les corpus ETAPE et REPERE, ayant un plus haut degré de spontanéité que les corpus ESTER, obtiennent des scores $HCIQ_n$ élevés. En s'appuyant sur le $HCIQ_n$, on remarque également que les corpus ESTER 1 et ESTER 2 nécessitent des temps de correction proches des durées des corpus tandis que les corpus ETAPE et REPERE demandent deux fois plus de temps de correction.

Tous les résultats présentés dans cette partie servent de point de référence afin de savoir si le développement de systèmes de SRL assistés présente un intérêt.

4.3.3 Correction sur la sortie d'un système automatique

Dans cette partie, nous évaluons la correction humaine du système automatique présenté en 2.9. Plus précisément, nous évaluons la correction sur différentes étapes qui le composent.

Pour tous les résultats présentés dans cette partie, les zones de parole superposée et les vérités terrain ont subi les mêmes traitements introduits en 4.2.4.

4.3.3.1 Étape segmentation GLR/BIC

La première étape que nous évaluons est la troisième étape du système de référence, la segmentation GLR/BIC (voir 2.5). À ce stade, un enregistrement représente une succession de segments non regroupés par locuteur. Ainsi, pour un enregistrement donné, aucun segment ne partage son label avec un autre segment.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	1114	1151	2289	667	5221
Créer une frontière	1031	1076	2285	870	5262
Supprimer une frontière	978	869	1100	469	3416
Toutes actions confondues	3505	3456	5829	2301	15091

TABLE 4.9 Nombre d'occurrences des différentes actions du système oracle pour divers corpus à la sortie de la $3^{ème}$ étape du système de référence

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	141,1	145,8	289,9	84,49	661,29
Créer une frontière	206,2	215,2	457,0	174,0	1052,4
Supprimer une frontière	83,1	73,9	93,5	39,9	290,4
Toutes actions confondues (HCIQ)	511,3	511,1	873,2	360,8	2256,4

TABLE 4.10 Durée estimée en minutes des différentes actions du système oracle pour divers corpus à la sortie de la $3^{ème}$ étape du système de référence. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Dans les tableaux 4.9 et 4.10 par rapport aux tableaux 4.6 et 4.7, on constate que pour tous les corpus, le nombre d'actions *"Créer une frontière"* et la durée associée diminuent. En revanche lors de la correction sans aide d'un système automatique (partie 4.3.2), il n'y avait pas de suppression de frontières, donc inévitablement, le nombre et la durée associés aux actions *"Supprimer une frontière"* augmentent, quel que soit le corpus. Mais globalement, la durée associée à ces deux actions relatives à la correction de la segmentation a diminué pour chaque corpus. Pour le corpus ESTER 1 par exemple, 289,3 minutes contre 335,6 minutes pour le tableau 4.7, soit un gain de temps de 14% environ pour la partie correspondant à la segmentation. Concernant l'action *"Changer le label locuteur"*, on remarque que le nombre ainsi que la durée augmentent très légèrement pour les corpus ESTER 1 et REPERE 1

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
DER	53,59	54,22	71,53	51,03	57,69
$DER_{false\ alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\ speaker}$	0,59	0,50	0,91	1,57	0,75
$DER_{confusion}$	53,00	53,73	70,62	49,46	56,93
# de locuteurs	1281	1171	1274	626	4352
# de changements de locuteur	1635	1484	1593	782	5494
# de segments	1672	1514	1622	813	5621
Moyenne de la durée des segments (s)	21,25	17,04	15,45	13,06	17,73
$HCIQ_n$ sans système automatique	0,94	1,24	1,93	2,14	1,41
$HCIQ_n$ seg. GLR/BIC (étape 3)	0,86	1,19	2,09	2,04	1,39
↳ gain étape 3/sans sys. auto. (%)	8,5	4,0	-8,2	4,7	1,4

TABLE 4.11 Mesures appliquées sur divers corpus à la sortie de la $3^{ème}$ étape du système de référence

tandis qu'ils augmentent significativement pour les corpus ESTER 2 et ETAPE. Le nombre d'occurrences de l'action "*Créer un label locuteur*" du tableau 4.9 ne change pas. La raison est que le système automatique utilise des labels de locuteurs abstraits (par exemple, "*locuteur1*") alors que l'humain travaille avec des labels de locuteurs concrets correspondant le plus souvent au nom réel du locuteur (par exemple, "*Arnold Schwarzenegger*"). Ainsi, l'humain doit toujours fournir au système les labels corrects.

Dans le tableau 4.11 par rapport au tableau 4.8, on constate que les DER de tous les corpus restent toujours élevés. Concernant le $HCIQ_n$, ils ont tous baissé à l'exclusion du corpus ETAPE qui a augmenté. On constate également que l'ordre des corpus en fonction du $HCIQ_n$ n'est plus le même. En effet, c'est maintenant ETAPE qui demande le plus de correction par unité de temps.

Dans le tableau 4.11 par rapport au tableau 4.5 donnant des informations sur les vérités terrain, on constate pour tous les corpus que le nombre de locuteurs [3] a grandement augmenté. En effet, le plus souvent, la segmentation augmente le nombre de locuteurs car il n'y a aucun regroupement entre segments non consécutifs. On constate aussi que le nombre de changements de locuteur [4] et le nombre de segments [5] ont, quant à eux, nettement diminué. En outre, la moyenne de la durée des segments [6] a augmenté d'environ

3. Nombre de locuteurs du tableau 4.5 : 467, 396, 235, 326 et 1424 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

4. Nombre de changements de locuteur du tableau 4.5 : 1954, 1881, 3369, 1287 et 8491 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

5. Nombre de segments du tableau 4.5 : 1991, 1911, 3398, 1318 et 8618 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

6. Moyenne de la durée des segments du tableau 4.5 : 17,85, 13,50, 7,37, 8,05 et 12,91 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

5 secondes quel que soit le corpus. Cela signifie sans doute que la segmentation n'est pas assez efficace pour détecter tous les points de rupture.

4.3.3.2 Étape resegmentation par l'algorithme de Viterbi

Dans cette partie, nous évaluons la sortie de la cinquième étape du système de référence, la resegmentation par l'algorithme de Viterbi (voir 2.7.1). La SRL à cette étape est composée de plusieurs ensembles de segments associés à différents labels. Dans cette partie, au lieu d'évaluer la sortie uniquement pour un seuil λ de 3 pour le regroupement mono-gaussien/BIC, soit le seuil prédéfini pour le système de référence (voir 2.9), nous évaluons la sortie pour divers seuils λ.

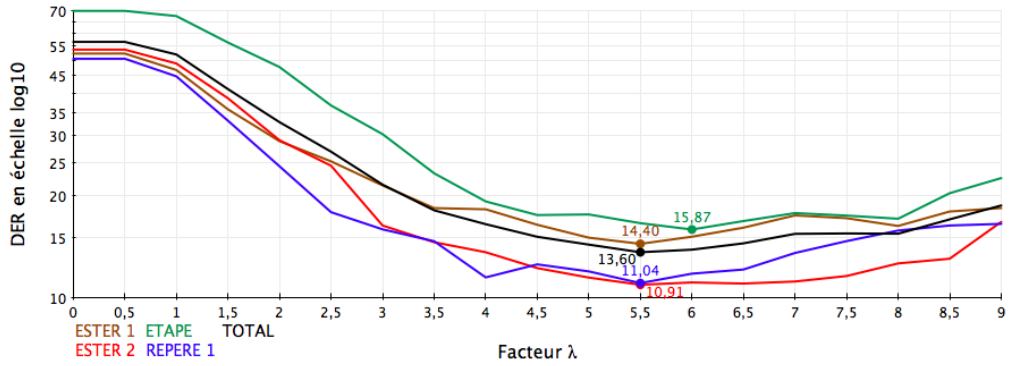


FIGURE 4.4 DER à la sortie de la cinquième étape du système de référence en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

La figure 4.4 donne la valeur du DER à la sortie de la cinquième étape du système de référence pour divers corpus en fonction de différents seuils λ. Concernant la forme des courbes, on note qu'elles décroissent jusqu'à un seuil λ entre 4 et 5. Elles restent ensuite plus ou moins stationnaires avant de remonter à partir d'un seuil entre 7 et 8.

La figure 4.5 donne le HCIQ_n à la sortie de la cinquième étape du système de référence en fonction de différents seuils λ pour divers corpus. Les meilleurs scores HCIQ_n dans la figure 4.5 se situent respectivement à un seuil λ de 5, 7, 5 et 4 pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1. Pour tous ces corpus rassemblés, le meilleur seuil se situe à un seuil λ de 5. Les tableaux 4.12, 4.13 et 4.14 apportent des informations liées à ces seuils.

Comparés aux résultats obtenus après la $3^{ème}$ étape du système de référence, tous les résultats sont plus faibles à quelques exceptions près. La moyenne de la durée des segments a augmenté quel que soit le corpus, le nombre d'actions *"Créer une frontière"* du corpus

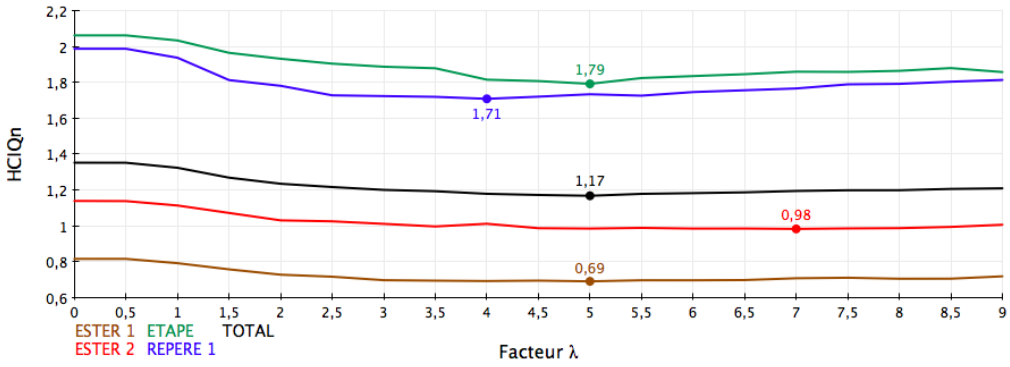


FIGURE 4.5 $HCIQ_n$ à la sortie de la cinquième étape du système de référence en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	5,0	7,0	5,0	4,0	5,0
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	800	897	1677	463	3866
Créer une frontière	942	1002	2314	840	5098
Supprimer une frontière	453	383	481	156	1493
Toutes actions confondues	2577	2642	4627	1754	11649

TABLE 4.12 Nombre d'occurrences des différentes actions du système oracle pour divers corpus à la sortie de la $5^{ème}$ étape du système de référence avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	5,0	7,0	5,0	4,0	5,0
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	101,3	113,6	212,4	58,6	489,7
Créer une frontière	188,4	200,4	462,8	168,0	1019,6
Supprimer une frontière	38,5	32,5	40,9	13,3	126,9
Toutes actions confondues (HCIQ)	409,1	422,8	748,9	302,3	1888,5

TABLE 4.13 Durée estimée en minutes des différentes actions du système oracle pour divers corpus à la sortie de la $5^{ème}$ étape du système de référence avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

ETAPE a légèrement augmenté et le nombre d'actions *"Créer un label locuteur"* reste constant.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	5,0	7,0	5,0	4,0	5,0
DER	15,02	11,16	17,58	11,47	14,33
$DER_{false\,alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\,speaker}$	0,33	0,19	0,39	0,85	0,36
$DER_{confusion}$	14,69	10,97	17,19	10,62	13,96
# de locuteurs	294	218	147	201	882
# de changements de locuteur	1200	1067	949	499	3739
# de segments	1237	1097	978	530	3866
Moyenne de la durée des segments (s)	28,72	23,52	25,62	20,03	25,09
$HCIQ_n$ sans système automatique	0,94	1,24	1,93	2,14	1,41
$HCIQ_n$ seg. GLR/BIC (étape 3)	0,86	1,19	2,09	2,04	1,39
↳ gain étape 3/sans sys. auto. (%)	8,5	4,0	-8,2	4,7	1,4
$HCIQ_n$ reseg. par Viterbi (étape 5)	0,69	0,98	1,79	1,71	1,17
↳ gain étape 5/sans sys. auto. (%)	26,6	21,0	7,2	20,1	17,0

TABLE 4.14 Mesures appliquées sur divers corpus à la sortie de la $5^{ème}$ étape du système de référence avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

En se référant aux tableaux 4.6 et 4.7, on remarque que le nombre d'actions *"Changer le label locuteur"* et *"Créer une frontière"* et leurs durées associées diminuent pour tous les corpus.

Si on compare avec le tableau 4.8, les DER sont environ trois à six fois plus faibles. Les $HCIQ_n$, quant à eux, ont tous diminué. Toutefois, comme pour les valeurs obtenues après la $3^{ème}$ étape du système de référence, on remarque que l'ordre des corpus requérant le plus de corrections pour une unité de temps n'est plus le même. En effet, c'est le corpus ETAPE qui a le $HCIQ_n$ le plus élevé.

Par rapport au tableau 4.5, on remarque que le nombre de locuteurs, le nombre de changements de locuteur et le nombre de segments sont plus faibles. On constate également que la moyenne de la durée des segments a augmenté de plus de 10 secondes, quel que soit le corpus.

4.3.3.3 Étape regroupement i-vecteur/ILP

Nous évaluons dans cette partie la sortie de la dernière étape du système de référence, le regroupement i-vecteur/ILP (voir 2.6.2). Pour le seuil λ du regroupement mono-gaussien/BIC, nous avons repris celui défini pour le système de référence en 2.9, soit un seuil λ de 3.

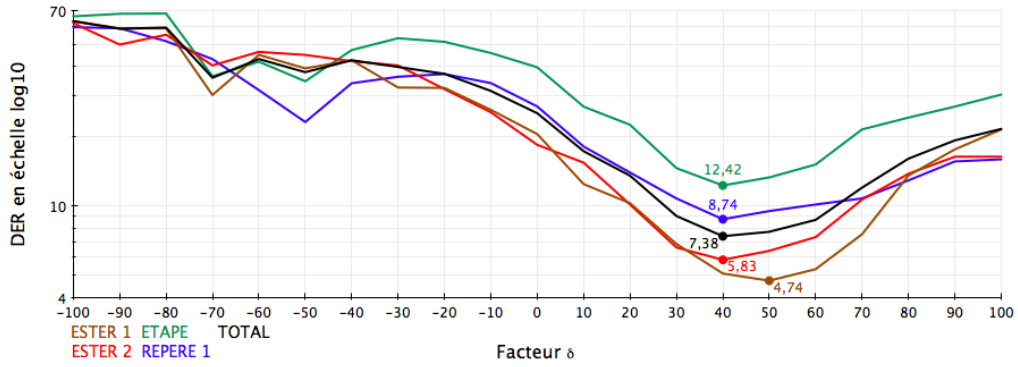


FIGURE 4.6 DER à la sortie de la dernière étape du système de référence en fonction de divers seuils δ du regroupement i-vecteur/ILP pour divers corpus

La figure 4.6 représente le DER à la sortie de la dernière étape du système de référence pour divers corpus en fonction de différents seuils δ. Dans cette figure, on note que toutes les courbes décroissent jusqu'à un seuil δ de 40 ou 50 avant de croître.

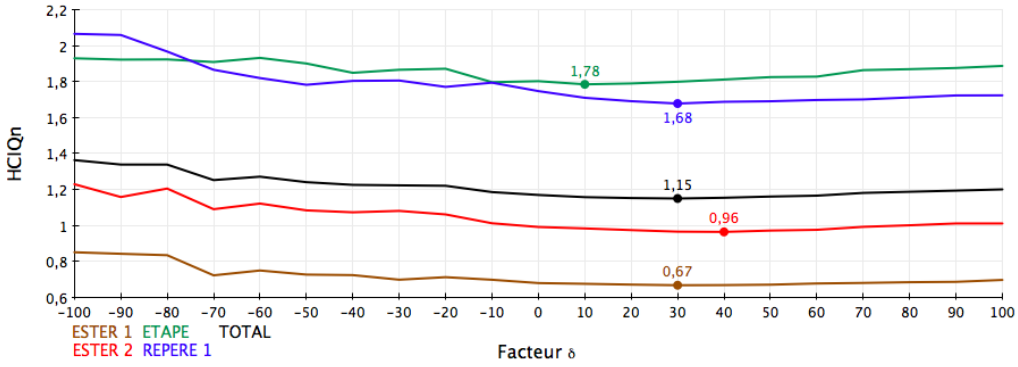


FIGURE 4.7 HCIQ_n à la sortie de la dernière étape du système de référence en fonction de divers seuils δ du regroupement i-vecteur/ILP pour divers corpus

La figure 4.7 représente le HCIQ_n à la sortie de la dernière étape du système de référence en fonction de différents seuils δ pour divers corpus. Les meilleurs scores HCIQ_n dans cette figure se situent respectivement à un seuil δ de 30, 40, 10 et 30 pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1. Pour tous ces corpus rassemblés, le meilleur seuil se situe à un seuil δ de 30. Les tableaux 4.15, 4.16 et 4.17 apportent des informations liées à ces seuils.

Par rapport aux résultats obtenus après la $5^{ème}$ étape du système de référence, le nombre d'occurrences de l'action *"Changer le label locuteur"* est plus faible à l'exception du corpus ETAPE, le nombre d'occurrences associé à l'action *"Créer une frontière"* a diminué sauf pour le corpus REPERE 1 et le nombre d'occurrences de l'action *"Supprimer une frontière"*

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil δ optimal	30	40	10	30	30
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	768	861	1711	415	3771
Créer une frontière	902	979	2298	842	5022
Supprimer une frontière	435	396	434	160	1477
Toutes actions confondues	2487	2596	4598	1712	11462

TABLE 4.15 Nombre d'occurrences des différentes actions du système oracle pour divers corpus à la sortie de la dernière étape du système de référence avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible HCIQ_n

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil δ optimal	30	40	10	30	30
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	97,3	109,1	216,7	52,6	477,7
Créer une frontière	180,4	195,8	459,6	168,4	1004,4
Supprimer une frontière	37,0	33,7	36,9	13,6	125,5
Toutes actions confondues (HCIQ)	395,5	414,7	746,0	297,0	1859,9

TABLE 4.16 Durée estimée en minutes des différentes actions du système oracle pour divers corpus à la sortie de la dernière étape du système de référence avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible HCIQ_n. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

a légèrement diminué sauf pour les corpus ESTER 2 et REPERE 1. Concernant le HCIQ_n, il a diminué quel que soit le corpus. Le DER, quant à lui, a diminué pour tous les corpus sauf pour le corpus ETAPE.

Comparées aux tableaux 4.6 et 4.7, les actions *"Changer le label locuteur"* et *"Créer une frontière"* ont diminué.

Par rapport au tableau 4.5 donnant des informations sur les vérités terrain, on remarque que le nombre de locuteurs, le nombre de changements de locuteur, le nombre de segments et la moyenne de la durée des segments sont très proches des résultats obtenus après la $5^{ème}$ étape du système de référence.

4.3.3.4 Discussion

De manière générale, avec ou sans aide d'un système automatique, les actions de segmentation représentent environ 56% de toutes actions confondues (tableau 4.18). De même, les erreurs de segmentation représentent environ 61% du temps de correction total (tableau 4.19). Nous pouvons également remarquer que l'action "*Créer une frontière*", qui

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil δ optimal	30	40	10	30	30
DER	6,83	5,83	26,85	10,75	9,37
$DER_{false\ alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\ speaker}$	0,30	0,20	0,40	0,88	0,75
$DER_{confusion}$	6,53	5,63	26,46	9,88	8,62
# de locuteurs	310	290	138	189	942
# de changements de locuteur	1222	1104	913	504	3793
# de segments	1259	1134	942	535	3920
Moyenne de la durée des segments (s)	28,22	22,75	26,60	19,84	24,74
$HCIQ_n$ sans système automatique	0,94	1,24	1,93	2,14	1,41
$HCIQ_n$ seg. GLR/BIC (étape 3)	0,86	1,19	2,09	2,04	1,39
↳ gain étape 3/sans sys. auto. (%)	8,5	4,0	-8,2	4,7	1,4
$HCIQ_n$ reseg. par Viterbi (étape 5)	0,69	0,98	1,79	1,71	1,17
↳ gain étape 5/sans sys. auto. (%)	26,6	21,0	7,2	20,1	17,0
$HCIQ_n$ reg. i-vecteur/ILP (dernière étape)	0,67	0,96	1,78	1,68	1,15
↳ gain dernière étape/sans sys. auto. (%)	28,7	22,6	7,8	21,5	18,4

TABLE 4.17 Mesures appliquées sur divers corpus à la sortie de la dernière étape du système de référence avec le seuil δ du regroupement i-vecteur/ILP donnant le plus faible $HCIQ_n$

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Sans système automatique	52,98	56,31	59,06	55,55	56,37
Seg. GLR/BIC (étape 3)	57,32	56,28	58,07	58,19	57,50
Reseg. par Viterbi (étape 5)	54,13	52,42	60,41	56,78	56,58
Reg. i-vecteur/ILP (dernière étape)	53,76	52,97	59,42	58,53	56,70
Moyenne	54,55	54,49	59,24	57,26	56,79

TABLE 4.18 Pourcentage du nombre d'occurrences des actions de segmentation par rapport au nombre d'occurrences de toutes actions confondues des tableaux 4.6, 4.9, 4.12 et 4.15

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Sans système automatique	60,28	63,21	68,35	61,99	64,12
Seg. GLR/BIC (étape 3)	56,58	56,56	63,04	59,28	59,51
Reseg. par Viterbi (étape 5)	55,46	55,08	67,26	59,97	60,71
Reg. i-vecteur/ILP (dernière étape)	54,97	55,34	66,55	61,28	60,75
Moyenne	56,82	57,55	66,30	60,63	61,27

TABLE 4.19 Pourcentage de la durée des actions de segmentation par rapport à la durée de toutes actions confondues des tableaux 4.7, 4.10, 4.13 et 4.16

est coûteuse en temps, est la plus utilisée puisqu'elle correspond à peu près à la moitié des corrections globales (tableau 4.20).

Si l'on examine les $HCIQ_n$, on remarque qu'annoter entièrement manuellement des données nécessite plus de temps humain que de corriger ces mêmes données issues d'un système automatique de SRL. Ainsi, étudier et développer une version assistée d'un système

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Sans système automatique	52,98	56,31	59,06	55,55	56,37
Seg. GLR/BIC (étape 3)	29,41	31,13	39,20	37,81	34,87
Reseg. par Viterbi (étape 5)	36,55	37,93	50,01	47,89	43,76
Reg. i-vecteur/ILP (dernière étape)	36,27	37,71	49,98	49,18	43,81
Moyenne	38,80	40,77	49,56	47,61	44,70

TABLE 4.20 Pourcentage du nombre d'occurrences de l'action *"Créer une frontière"* par rapport au nombre d'occurrences de toutes actions confondues des tableaux 4.6, 4.9, 4.12 et 4.15

automatique de SRL peut présenter un intérêt afin de réduire encore plus le temps de correction humain. On remarque également qu'optimiser le DER (les seuils λ et δ) d'un système permet d'avoir une entrée d'un système de correction qui réduit l'HCIQ$_n$ pour les systèmes de SRL étudiés.

Pour le système de SRL assisté par un humain que nous présentons dans le chapitre suivant, nous utilisons la sortie de la $5^{ème}$ étape comme entrée, soit la sortie de l'étape de resegmentation par l'algorithme de Viterbi. Ce choix a été fait, car le HCIQ$_n$ ne baisse que très légèrement lors de la dernière étape par rapport à la précédente et surtout, la sortie de la $5^{ème}$ étape est plus cohérente avec les méthodes assistées que nous proposons dans la suite. La figure 4.5, donnant le HCIQ$_n$ à la sortie de la $5^{ème}$ étape en fonction de différents seuils λ pour divers corpus, sert alors de point de départ pour les méthodes assistées que nous allons expérimenter. En effet, les scores de cette figure serviront de références afin de déterminer si une méthode assistée réduit ou non l'effort et le temps d'interaction humain.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	5,0	7,0	5,0	4,0	5,0
DER avant correction	15,02	11,16	17,58	11,47	14,33
DER après correction	2,36	2,63	8,75	5,62	4,24
DER avant/DER après (%)	15,71	23,57	49,77	49,0	29,59

TABLE 4.21 DER avant et après la correction des erreurs de regroupements en locuteurs de la sortie de la $5^{ème}$ avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible HCIQ$_n$

Le tableau 4.21 donne le DER après la correction des erreurs de regroupements en locuteurs de la sortie de la $5^{ème}$ du système de référence. Dans ce tableau, nous remarquons que le DER chute radicalement après la correction des erreurs de regroupements en locuteurs. Le DER après correction représente les erreurs dues à une mauvaise segmentation. Les ratios du tableau 4.21 montrent que les erreurs de segmentation contribuent au maximum

à environ 33% du DER (correspond à la moyenne de la ligne DER avant sur DER après). Comparativement, les erreurs de segmentation correspondent à environ 61% du HCIQ (tableau 4.19). La correction des erreurs de segmentation fait croître le HCIQ tandis qu'elle affecte de façon négligeable le DER.

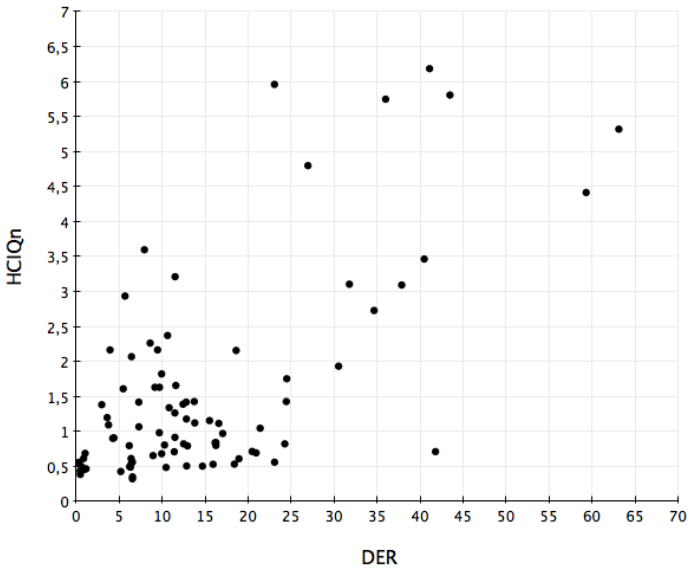


FIGURE 4.8 DER par rapport au $HCIQ_n$ pour chaque enregistrement des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 à la sortie de la $5^{ème}$ du système de référence avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$ pour l'ensemble des enregistrements

La figure 4.8 montre le DER par rapport au $HCIQ_n$ pour chaque enregistrement des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 à la sortie de la $5^{ème}$ du système de référence. Cette figure montre que le $HCIQ_n$ et le DER sont corrélés. Cette corrélation, de 0,66, est significative car elle dépasse largement le seuil critique de 0,21 pour un total de 87 fichiers. Plus le DER d'un fichier à corriger est faible, plus son $HCIQ_n$ est faible. Cependant, la corrélation est plus ou moins forte selon les actions. En effet, la corrélation entre le DER et le $HCIQ_n$ des actions respectives "*Créer un label locuteur*", "*Changer le label locuteur*", "*Créer une frontière*" et "*Supprimer une frontière*" est de 0,33, 0,70, 0,64 et 0,41.

En outre, la figure 4.8 montre aussi que la majorité des enregistrements évalués ont un DER compris entre 0 et 25% et nécessitent un temps de correction entre 0,5 et 1,5 de la durée de l'enregistrement.

La figure 4.9 donne la F-Mesure de la détection des frontières de changement de locuteur par rapport au $HCIQ_n$ pour chaque enregistrement des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 à la sortie de la $5^{ème}$ du système de référence. La F-Mesure a été calculée à partir de la précision et du rappel de la détection des frontières avec une

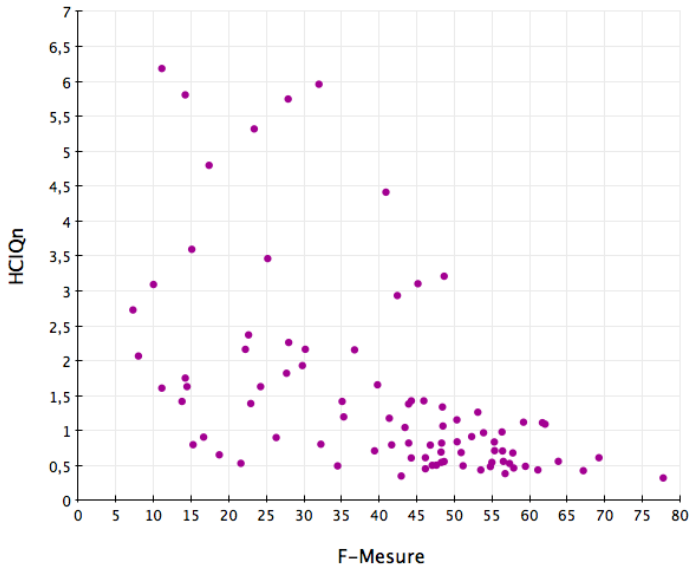


FIGURE 4.9 F-Mesure des frontières avec une tolérance de 250 ms par rapport au $HCIQ_n$ pour chaque enregistrement des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$ pour l'ensemble des enregistrements

tolérance aux frontières de 250 ms. À partir de cette figure, on constate que la majorité des enregistrements ont une F-Mesure comprise entre 40 et 65%. Seulement la moitié des frontières sont correctement détectées pour la majorité des enregistrements. Cette figure confirme le nombre d'occurrences élevé des actions de segmentation des précédents tableaux.

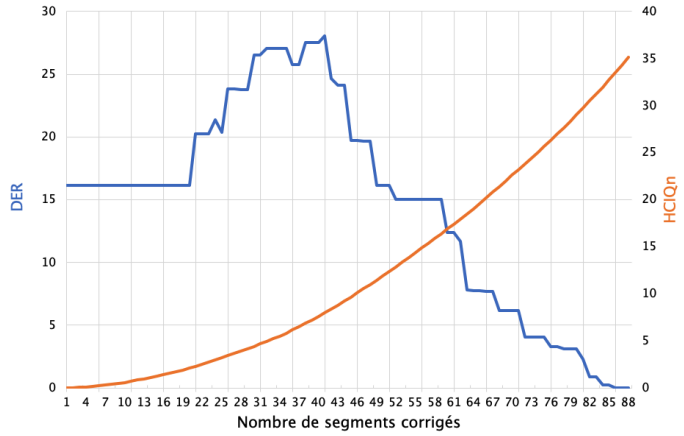


FIGURE 4.10 DER et $HCIQ_n$ en fonction du nombre de segments corrigés par rapport à la référence pour l'enregistrement 20071220_1900_1920_inter du corpus ESTER 2 avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$ pour l'ensemble des enregistrements

La figure 4.10 donne le DER et le $HCIQ_n$ en fonction du nombre de segments corrigés par rapport à la référence de l'enregistrement 20071220_1900_1920_inter du corpus ESTER 2. Cette figure met en évidence que le DER ne diminue pas nécessairement et régulièrement au fur et à mesure des corrections humaines comme nous pourrions le penser de prime abord.

4.4 Bilan

Ce chapitre a permis d'introduire une nouvelle mesure, le HCIQ, ainsi qu'un jeu d'actions pour corriger une SRL. Il a également proposé et appliqué un protocole pour évaluer les coûts de chaque action. Le jeu d'actions, le HCIQ et le coût des actions permettent ensemble d'évaluer le temps passé d'un correcteur humain pour une SRL.

En outre, ce chapitre a introduit un automate pour simuler les corrections de SRL d'un annotateur humain. Cet automate peut servir de base pour développer un système assisté. L'automate décrit dans ce chapitre est disponible dans l'outil *S4D* [Broux *et al.*, 2018a], présenté en détail en annexe D.

Enfin, ce chapitre a montré que corriger des données issues d'un système automatique de SRL est moins chronophage que d'annoter ces mêmes données entièrement manuellement. Il a également montré que le temps de correction était principalement dû au nombre d'occurrences de l'action *"Créer une frontière"* nécessaire à la correction des erreurs de segmentation et de son coût associé.

Dans la suite de ce manuscrit, nous allons proposer et étudier des méthodes de SRL assistés par l'humain.

Chapitre 5

Réduction du coût d'intervention humain par des systèmes assistés

Dans ce chapitre, nous présentons et évaluons trois systèmes de SRL assistés afin de tenter de réduire le coût d'intervention humain nécessaire à la correction de la sortie d'un système de SRL automatique. Nous commençons par décrire le système de base sur lequel s'appuie chaque système assisté décrit dans ce chapitre. Ensuite, nous présentons un système avec une méthode assistée dans le cadre où il n'y a pas d'erreur de segmentation. Pour finir, nous étudions un système avec diverses méthodes assistées lorsqu'il y a des erreurs de segmentation et des erreurs de regroupement en locuteurs.

5.1 Système de base utilisé

Dans cette section, nous décrivons l'architecture du système de SRL assisté par un humain que nous proposons et utilisons dans ce chapitre.

5.1.1 Description du système assisté par un humain

La figure 5.1 présente l'architecture du système de SRL assisté par un humain que nous proposons. Il est composé de deux parties principales. La première consiste à appliquer un système de SRL automatique sur un flux audio. Une segmentation initiale du flux est alors obtenue. La seconde consiste à demander à un humain de corriger la sortie de la première partie. Chaque correction humaine est à son tour prise en considération par un système qui tente d'améliorer la SRL en réaffectant des zones du signal audio non vérifiées à des locuteurs différents. Cet ajustement a pour objectif de réduire l'effort et le temps de correction restant. Dans ce système assisté, l'annotateur peut effectuer des corrections

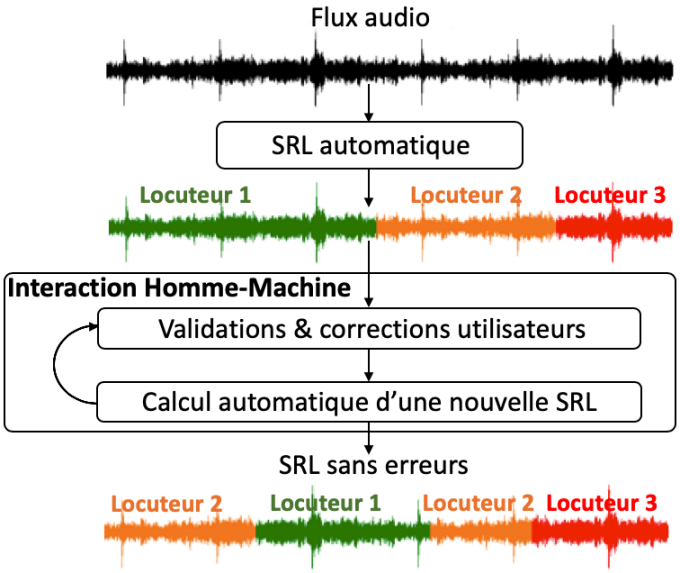


FIGURE 5.1 Architecture du système de SRL assisté par l'humain

sur le regroupement en locuteurs ou sur la segmentation. À la fin du processus, le taux d'erreurs de SRL, à savoir le DER, devrait avoir diminué et même être nul si toutes les erreurs de segmentation et de classification sont corrigées.

La première étape peut être très lente. Par conséquent, elle n'est généralement pas effectuée sur l'instant, mais à l'avance. Comme nous l'avons précisé en 4.3.3.4, cette première étape correspond à la sortie de la $5^{ème}$ du système de référence.

Les conditions expérimentales, quant à elles, sont identiques à celles du chapitre précédent, soit : seules les zones annotées de l'UEM sont évaluées, les silences intra-locuteur de moins de 2 secondes sont supprimés et chaque zone de parole superposée est représentée par un segment avec un label regroupant tous les noms des intervenants.

5.1.2 Conditions expérimentales

Pour toutes les expériences de ce chapitre, nous avons choisi d'utiliser le contexte suivant : un annotateur humain corrige une SRL au moyen de l'application d'annotation de référence *Transcriber*. Nous supposons également que les segments de parole sont présentés avec la transcription de la parole afin d'aider l'annotateur. Dans le cadre de ce contexte, nous reprenons les règles définies en 4.2.2, soit un automate simulant un humain qui :

1. ne fait aucune erreur ;
2. corrige l'émission du début à la fin en suivant l'ordre temporel afin de valider l'annotation automatique faite a posteriori ;

3. ne corrige que le tour de parole venant d'être écouté.

Comme nous l'avons décrit dans la section 4.2.1, faire appel à un humain pour chaque SRL à corriger présente de forts inconvénients. Ainsi, pour se rapprocher du contexte choisi, avons-nous décidé d'utiliser l'automate ainsi que le jeu d'actions déterministes s'appuyant sur *Transcriber*, tous deux présentés dans la partie 4.2. Les temps de ce jeu d'actions, quant à eux, sont extraits à partir du protocole proposé dans la partie 4.1. La première règle permet d'éviter la modélisation aléatoire et complexe des erreurs pouvant être réalisées par un humain. Elle permet également d'avoir à la fin de la correction un document sans erreurs et par conséquent un DER à 0%.

5.2 Regroupement en locuteurs assisté

Au lieu d'étudier simultanément l'étape de segmentation assistée et l'étape de regroupement assistée, nous décidons d'étudier en premier l'étape de regroupement en locuteurs assisté. Ainsi dans le cadre de cette partie l'étape de segmentation n'est pas traitée.

Dans cette partie, nous présentons des modifications que nous apportons au système de SRL d'entrée afin de n'avoir que des erreurs de regroupement en locuteurs à corriger. Ensuite, nous présentons une méthode de regroupement assistée.

5.2.1 Modifications du système de SRL d'entrée

Pour le système de SRL d'entrée, la principale modification du système de SRL d'entrée se situe à l'étape de segmentation. En effet, puisque nous nous concentrons uniquement sur l'étape de regroupement en locuteurs, nous utiliserons une segmentation manuelle parfaite, à savoir la vérité terrain. Cela a pour objectif d'obtenir des segments réellement prononcés par un seul locuteur, permettant ainsi de simplifier l'étude des méthodes de regroupement assistée en se plaçant dans des conditions idéales. Dans cette même optique, nous retirons l'étape de resegmentation par l'algorithme de Viterbi du système de SRL d'entrée. En effet, cela impliquerait une réévaluation des frontières et donc une probabilité de ne pas obtenir des segments purs.

Une fois que toutes les étapes sélectionnées du système de référence ont été appliquées sur le document audio, il ne reste plus qu'à faire appel à l'automate pour vérifier chaque segment.

La figure 5.2 montre le DER du système de SRL d'entrée pour divers corpus en fonction de différents seuils λ correspondant au seuil de l'étape du regroupement mono-gaussien/BIC

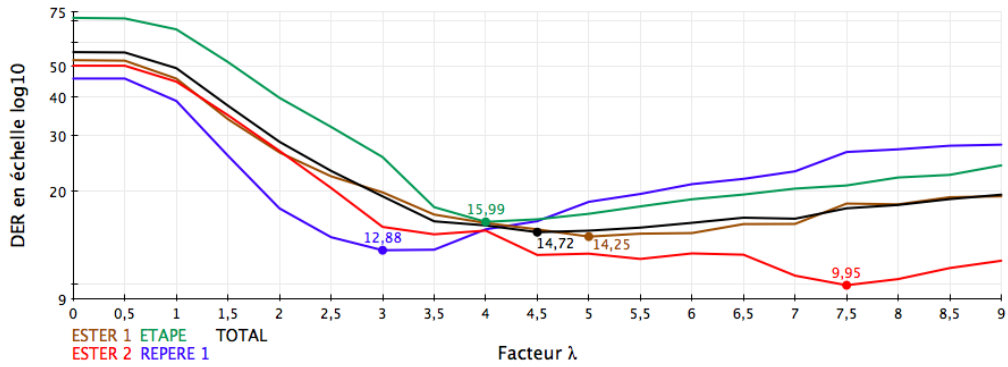


FIGURE 5.2 DER du système de SRL d'entrée sans erreur de segmentation en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

(voir 2.6.1.1). Concernant la forme des courbes, on note qu'elles décroissent jusqu'à une valeur λ entre 3 et 4 avant de remonter, à l'exception de celle du corpus ESTER 2.

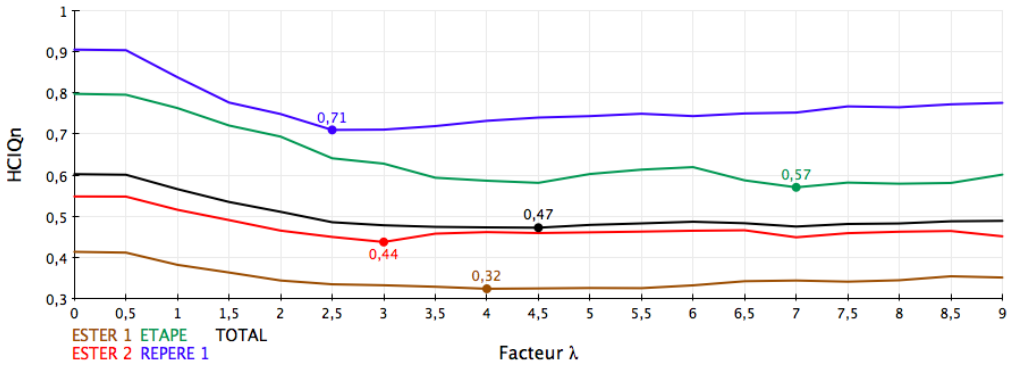


FIGURE 5.3 $HCIQ_n$ du système de SRL d'entrée sans erreur de segmentation en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

La figure 5.3 montre le $HCIQ_n$ du système de SRL d'entrée pour divers corpus en fonction de différents seuils λ. Dans cette figure, les meilleurs scores des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 se situent respectivement à un seuil λ de 4, 3, 7 et 2,5. Pour tous ses corpus regroupés, le meilleur score se situe à un seuil λ de 4,5. Les tableaux 5.1, 5.2 et 5.3 proposent des informations liées à ces seuils. Le tableau 5.1 présente le nombre d'occurrences de chaque action de regroupement en locuteurs, le tableau 5.2 les durées consacrées pour chaque type d'action et le tableau 5.3 des mesures appliquées sur la sortie du système de SRL d'entrée. Cette figure et ces tableaux servent de point de référence pour nos expériences. En effet, il faut que les méthodes qu'on propose fournissent des $HCIQ_n$ inférieurs aux meilleurs scores $HCIQ_n$ présents dans ces résultats.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	3,0	7,0	2,5	4,5
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	874	883	1623	499	4035
Toutes actions confondues	1256	1243	1778	794	5227

TABLE 5.1 Nombre d'occurrences des différentes actions du système oracle pour divers corpus à la sortie du système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	3,0	7,0	2,5	4,5
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	110,7	111,8	205,6	63,2	511,1
Toutes actions confondues (HCIQ)	191,6	188,0	238,4	125,6	763,4

TABLE 5.2 Durée estimée en minutes des différentes actions du système oracle pour divers corpus à la sortie du système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	3,0	7,0	2,5	4,5
$DER_{confusion}$	15,75	15,31	20,31	14,19	14,72
# de locuteurs	355	361	102	227	913
$HCIQ_n$ système oracle	0,32	0,44	0,57	0,71	0,47

TABLE 5.3 Mesures appliquées sur divers corpus à la sortie du système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

Si l'on compare ces résultats aux tableaux 4.6, 4.7 et 4.8 où chaque enregistrement de chaque corpus est représenté entièrement par son fichier UEM, on constate que les $HCIQ_n$ des corpus [1] ont été divisés environ par trois. C'est tout-à-fait normal puisque nous avons enlevé la difficulté liée à la recherche d'une segmentation parfaite. On constate également que le nombre d'occurrences [2] de l'action *"Changer de label locuteur"* et la durée associée [3] sont inférieurs.

1. $HCIQ_n$ du tableau 4.8 : 0,94, 1,24, 1,93, 2,14 et 1,41 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

2. Nombre d'occurrences de l'action *"Changer de label locuteur"* du tableau 4.6 : 1107, 947, 1762, 645 et 4461 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

3. Durée estimée en minutes de l'action *"Changer de label locuteur"* du tableau 4.7 : 140,2, 119,9, 223,2, 81,7 et 565,0 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

Quant au tableau 5.3, on remarque que les DER sont plus ou moins proches de ceux du tableau 4.14 [4] où les enregistrements correspondent à la sortie de la $5^{ème}$ étape du système de référence. On remarque également que le nombre de locuteurs de chaque corpus a diminué par rapport au tableau 4.5 [5] donnant des informations sur les vérités terrain des corpus.

5.2.2 Aide à la correction des labels par adaptation incrémentale

Dans cette partie, nous présentons une méthode assistée par l'humain réévaluant les labels des segments de parole restant à vérifier. Cette méthode a été présentée au workshop international *Multimodal Media Data Analytics* de 2016 [6] [Broux *et al.*, 2016].

5.2.2.1 Principe de la méthode

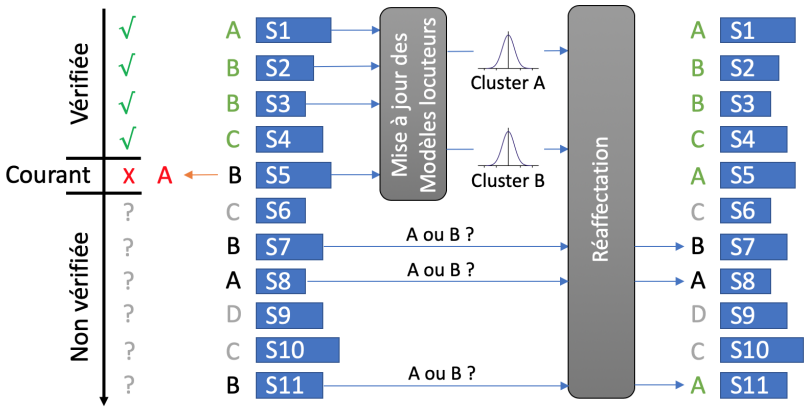


FIGURE 5.4 Exemple d'une entrée utilisateur et d'une réaffectation

Les actions de l'utilisateur consistent à valider ou à corriger les labels des locuteurs estimés par le système de SRL. La stratégie de la méthode assistée par l'humain que nous proposons consiste à associer chaque correction de label, impliquant toujours deux locuteurs C_i et C_j, au calcul de nouveaux modèles de locuteurs, entraînés sur les segments de parole vérifiés. Les nouveaux modèles sont rapides à estimer puisque les modèles de locuteurs utilisés sont des modèles mono-gaussiens (voir 2.4.1). À l'issue d'une correction, les nouveaux modèles de locuteurs sont supposés être plus précis que les modèles déduits durant l'initialisation, c'est-à-dire les modèles du système de SRL entièrement automatique.

4. DER du tableau 4.14 : 15,02, 11,16, 17,58, 11,47 et 14,33 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

5. Nombre de locuteurs du tableau 4.5 : 467, 396, 235, 326 et 1424 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

6. Lien de la conférence : https ://mklab.iti.gr/mmda2016/

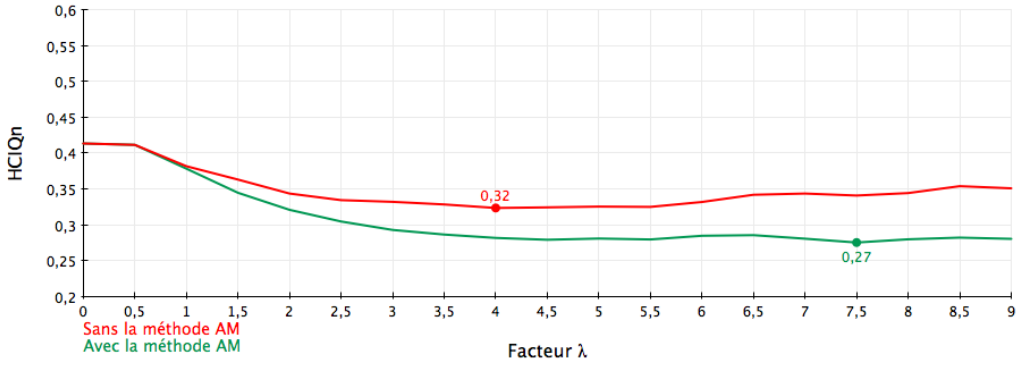
Ces modèles de locuteurs simples sont ensuite utilisés pour réestimer la distance ΔBIC avec les segments de parole restants liés à la dernière correction de label de locuteurs (les segments attribués à C_i et C_j seulement). Cette méthode correspond à un apprentissage incrémental (voir 3.6.1).

Une illustration de ces interactions est fournie par la figure 5.4. Dans cet exemple, quatre locuteurs (A, B, C et D) ont été détectés par le système de SRL entièrement automatique. L'utilisateur a validé manuellement les quatre premiers segments de parole ($S_1 \ldots S_4$) avant de reporter une modification de label locuteur pour le segment S_5, étiqueté comme locuteur B au lieu de locuteur A. La méthode assistée consiste à créer des modèles pour les locuteurs qui ont été confondus (A et B). Ces modèles sont utilisés pour réestimer les labels des segments restants étiquetés avec A ou B (les segments S_7, S_8 et S_{11} dans l'exemple) et peuvent conduire à une modification des labels de locuteurs (le segment S_{11}). Les segments de parole restants étiquetés avec d'autres labels (C et D) ne sont pas réévalués. La SRL modifiée est mise à jour avant que l'annotateur passe au segment suivant S_6. Le processus se réitère tant qu'il reste des segments à vérifier.

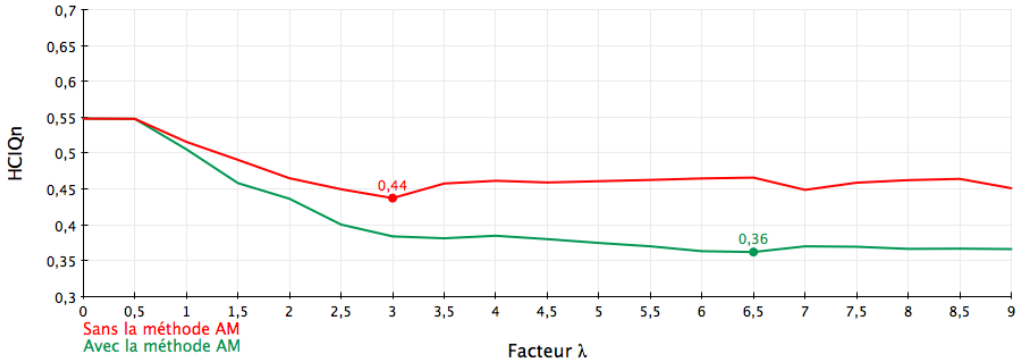
Cette méthode réévalue les segments non vérifiés qu'ils correspondent ou non à de la parole superposée. Sans une vérification humaine, les méthodes pour la détection de la parole superposée ne sont pas suffisamment fiables. Cependant, lorsque l'humain valide ou corrige un segment correspondant à de la parole superposée, on suppose que l'humain l'indique au système assisté. Cette méthode n'effectue aucune réaffectation après une correction ou une validation des segments de parole superposée.

Dans la suite de ce manuscrit, nous faisons référence à cette méthode avec le nom AM pour adaptation des modèles.

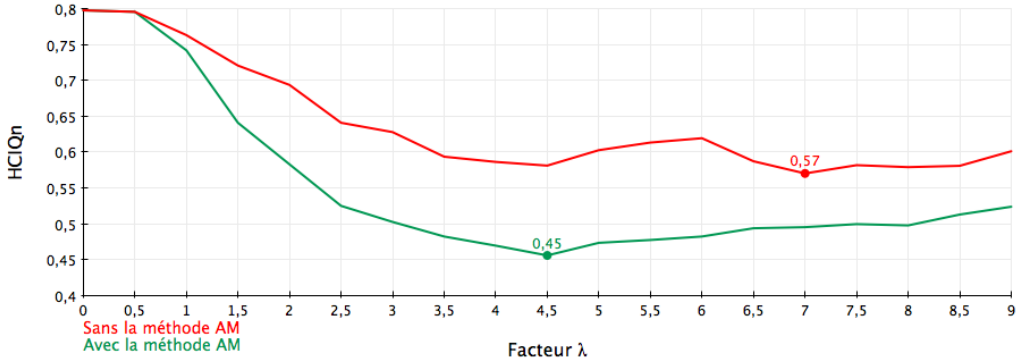
5.2.2.2 Évaluation de la méthode



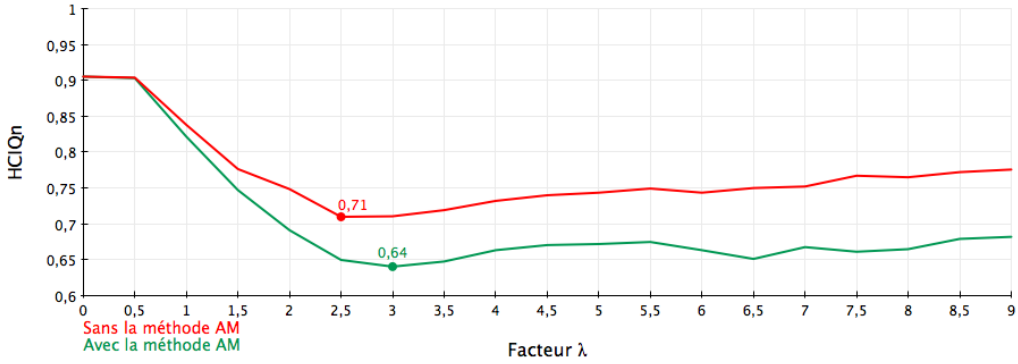
(a) ESTER 1



(b) ESTER 2



(c) ETAPE



(d) REPERE 1

FIGURE 5.5 $HClQ_n$ de la méthode AM en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

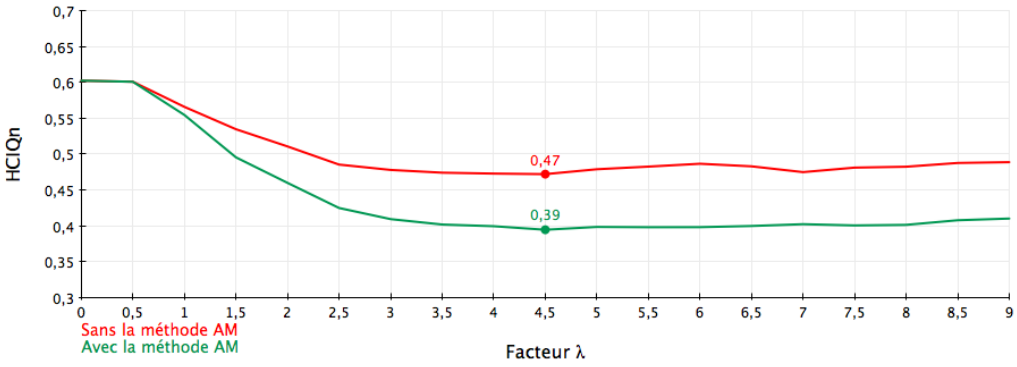


FIGURE 5.6 $HCIQ_n$ de la méthode AM en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

L'objectif est d'obtenir une SRL sans erreurs avec le moins d'action possible. Pour atteindre ce but, nous comparons le $HCIQ_n$ obtenu avec ou sans la méthode AM (c'est-à-dire avec ou sans une réestimation des labels de locuteurs non vérifiés) en utilisant divers seuils λ pour l'étape du regroupement mono-gaussien/BIC du système de SRL entièrement automatique.

La figure 5.5 donne le $HCIQ_n$ de la méthode AM pour divers corpus. La figure 5.6, quant à elle, donne le $HCIQ_n$ de la méthode AM pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés. Avec la méthode AM, les meilleurs scores des corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 se situent respectivement à un seuil λ de 7,5, 6,5, 4,5 et 3. Pour tous les corpus regroupés, le meilleur score se situe à un seuil λ de 4,5. Les tableaux 5.4, 5.5 et 5.6 donnent des informations liées à ces seuils. Le tableau 5.4 montre le nombre d'occurrences de chaque action, le tableau 5.5 les durées consacrées à chaque type d'action et le tableau 5.6 des mesures complémentaires.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	7,5	6,5	4,5	3,0	4,5
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	649	628	1246	402	3046
Toutes actions confondues	1031	988	1401	697	4238

TABLE 5.4 Nombre d'occurrences des différentes actions du système avec la méthode AM pour divers corpus à la sortie du système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

De manière générale en examinant les courbes des figures 5.5 et 5.6, que le système prenne ou non en considération les corrections humaines, le $HCIQ_n$ décroît jusqu'à un seuil λ d'environ 3 avant de rester plus ou moins constant. Pour chaque corpus, la courbe du

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	7,5	6,5	4,5	3,0	4,5
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	82,2	79,5	157,8	50,9	385,8
Toutes actions confondues (HCIQ)	163,1	155,7	190,6	113,4	638,1

TABLE 5.5 Durée estimée en minutes des différentes actions du système avec la méthode AM pour divers corpus à la sortie d'un système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	7,5	6,5	4,5	3,0	4,5
$DER_{confusion}$	18,18	12,47	16,19	12,88	14,72
# de locuteurs	242	238	131	195	913
$HCIQ_n$ système oracle	0,32	0,44	0,57	0,71	0,47
$HCIQ_n$ AM	0,27	0,36	0,45	0,64	0,39

TABLE 5.6 Mesures appliquées sur divers corpus à la sortie du système de SRL automatique sans erreur de segmentation avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$ pour le système avec la méthode AM

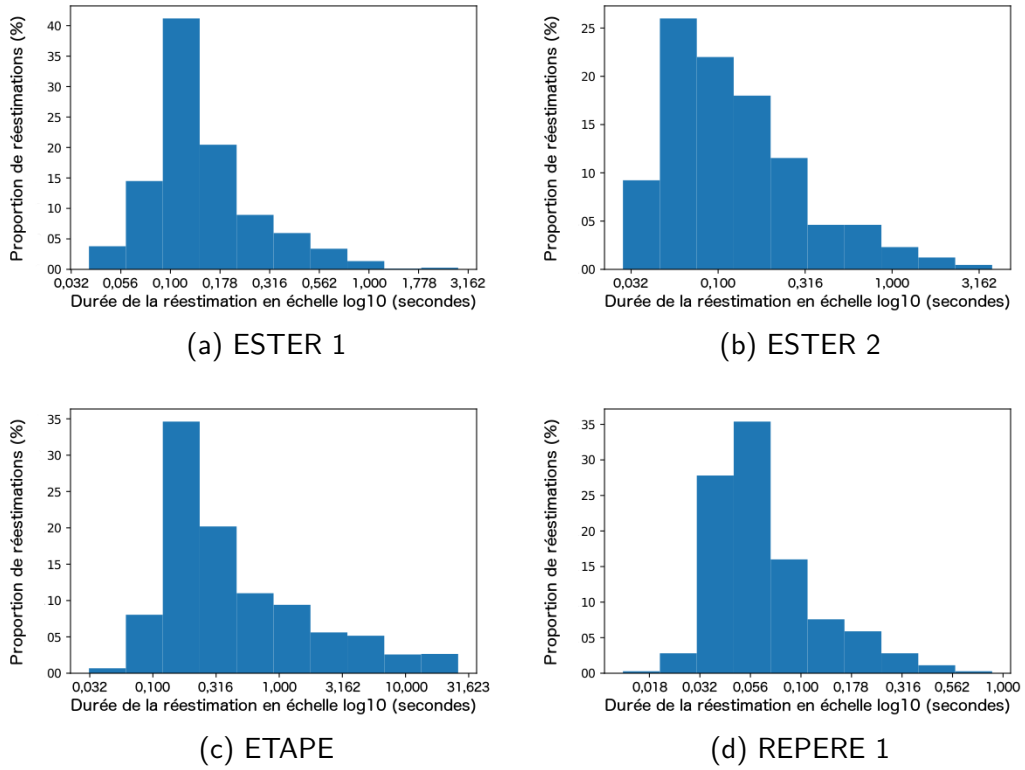
système avec AM offre de bien meilleurs résultats quel que soit le seuil λ que celle du système ne prenant pas en considération les corrections humaines, hormis aux seuils 0 et 0,5 où les valeurs sont identiques pour les deux systèmes. Cependant, on constate que les meilleures valeurs en $HCIQ_n$ ne se situent pas à un même seuil selon le corpus.

Si l'on compare les tableaux 5.3 et 5.6, on constate que tous les DER ont diminué à l'exception du corpus ESTER 1 et du corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1. On constate également que le nombre de locuteurs est plus faible à l'exception du corpus ETAPE et du corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1.

Dans les tableaux 5.4 et 5.5, on constate une diminution du nombre d'occurrences de l'action "*Changer le label locuteur*" pour tous les corpus.

À partir du tableau 5.6, on remarque que la méthode AM permet de réduire le $HCIQ_n$ d'environ 0,08. Plus précisément en s'appuyant sur les tableaux 5.2 et 5.5, elle permet d'économiser 28,5 min, 32,3 min, 47,8 min et 12,2 min pour respectivement les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1[7]. Pour le corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1, le temps économisé est de 125,3 min.

7. HCIQ du tableau 5.2 : 191,6, 188,0, 238,4, 125,6 et 763,4 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.



(a) ESTER 1 (b) ESTER 2

(c) ETAPE (d) REPERE 1

FIGURE 5.7 Temps de réestimation après chaque correction pour divers corpus

Après chaque correction humaine de regroupement en locuteurs, les segments non vérifiés dont les labels sont impliqués sont réestimés. Ce processus est généralement rapide puisque le temps de réestimation dure moins d'une demi-seconde dans la majorité des cas (figure 5.7). Par conséquent, cette méthode peut être appliquée en temps réel sans aucun impact sur l'interface utilisateur. Remarquons cependant qu'il existe des cas extrêmement rares où le temps de réestimation dure plusieurs secondes comme on peut le voir avec le corpus ETAPE. Dans ce corpus, ces cas proviennent uniquement d'un seul fichier. Ce fichier d'une durée de 25 minutes présente énormément de paroles superposées. En raison de la manière dont on a considéré la parole superposée dans ce fichier, de nombreux segments ont une durée inférieure à 2 secondes. À cause de la quantité importante de parole superposée et de segments inférieurs à 2 secondes, le système de SRL automatique d'entrée a regroupé par erreur la majorité des segments en un seul locuteur. Par conséquent, après chaque correction humaine en lien avec ce locuteur, la méthode AM doit traiter un nombre de segments assez conséquent donnant ainsi un temps de réestimation élevé.

5.2.2.3 Discussion

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Gain de AM en $HCIQ_n$ (%)	15,62	18,18	21,05	9,86	17,02

TABLE 5.7 Gain relatif de la méthode AM par rapport au système oracle

La méthode AM que nous proposons prend en considération les corrections d'un annotateur en modifiant l'affectation des labels des segments non vérifiés. Dans le cas d'une segmentation parfaite, cette méthode assistée permet d'obtenir une réduction notable du temps passé à la correction, soit une réduction d'environ 16,3% (correspond à la moyenne des valeurs du tableau 5.7). Non seulement cette méthode est efficace, mais les corrections sont également effectuées rapidement. Ainsi, elle peut être intégrée dans une application sans impacter la réactivité de l'interface et sans augmenter le travail d'un annotateur à condition que la segmentation soit parfaite.

5.3 Segmentation et regroupement en locuteurs assistés

Dans cette partie, nous étudions la segmentation assistée et le regroupement en locuteurs assistés. Nous commençons par expliquer l'intérêt d'une méthode de regroupement en locuteurs assistée pour une méthode de segmentation assistée. Ensuite, nous présentons une méthode de segmentation assistée et deux méthodes de regroupements en locuteurs assistées. Enfin, nous évaluons ces méthodes séparément puis ensemble.

5.3.1 Regroupement en locuteurs assisté : une aide à la segmentation assistée

Réestimer ou améliorer l'étape de segmentation sans réévaluer l'étape de regroupement en locuteurs est une tâche difficile. Effectivement, même si l'on peut s'appuyer sur des paramètres extraits du signal audio comme l'énergie, l'amélioration de la segmentation s'appuie essentiellement sur les modèles de locuteurs trouvés. Ainsi, en vue d'aider une méthode de segmentation assistée, nous l'utilisons conjointement avec une méthode de regroupement en locuteurs assistée.

5.3.2 Aide à la correction par resegmentation incrémentale

Dans cette partie, nous présentons une méthode assistée par l'humain qui consiste à réévaluer les zones de SRL non vérifiées.

5.3.2.1 Principe de la méthode

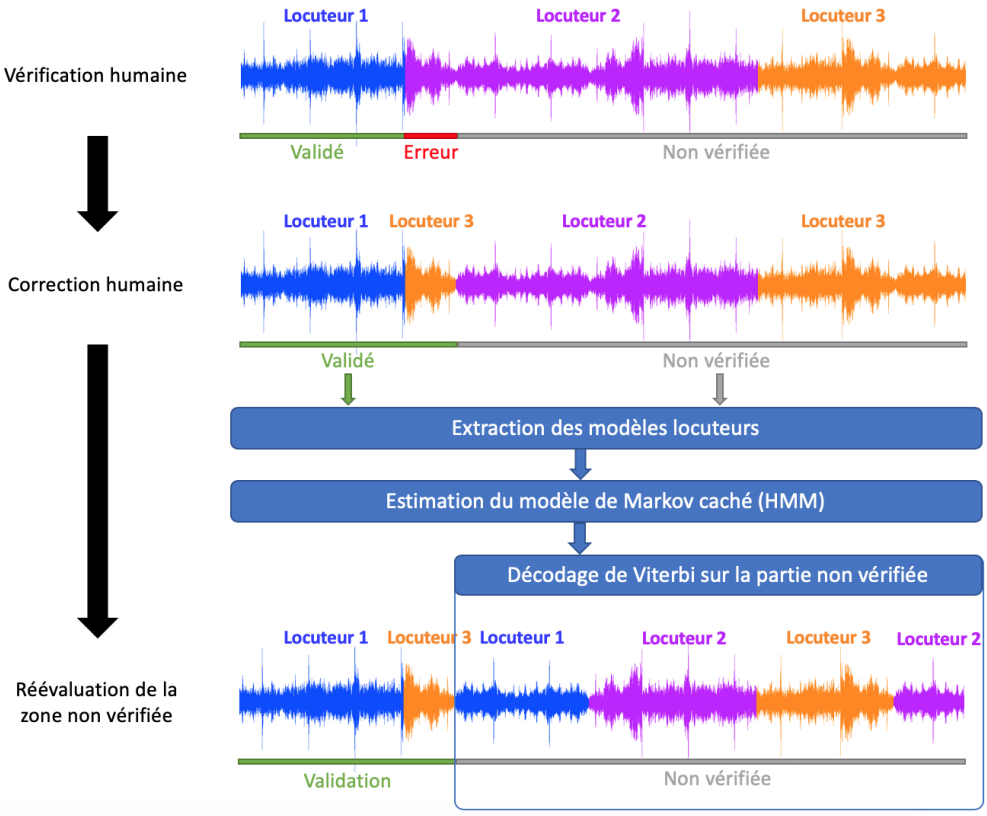


FIGURE 5.8 Exemple d'une intervention humaine et d'une réévaluation de la segmentation

Pour la réévaluation de la segmentation, nous nous appuyons principalement sur le décodage de Viterbi (voir 2.7.1). Nous avons décidé de relancer une resegmentation dès qu'un tour de parole a été entièrement corrigé. Ce choix s'appuie sur l'hypothèse qu'un tour de parole entièrement corrigé permet d'avoir des modèles de locuteurs plus précis et ainsi avoir un meilleur modèle de Markov caché permettant alors de mieux déterminer la probabilité d'émission de la séquence d'observations. Seule la partie audio non vérifiée par l'humain est affectée par cette réévaluation. La partie déjà vérifiée reste, quant à elle, inchangée. La figure 5.8 illustre ce procédé. Cette méthode correspond à un apprentissage incrémental (voir 3.6.1). Dans la suite de ce manuscrit, nous nommons ce procédé la méthode RV pour "Resegmentation par Viterbi".

Puisque les segments de parole superposée présentent une difficulté pour les méthodes de SRL, la méthode RV se comporte de la même manière que la méthode AM concernant ces segments (voir 5.2.2.1). Ainsi, les segments de parole superposée ayant été vérifiés ne sont pas réévalués par la méthode RV.

5.3.2.2 Méthodes de regroupements en locuteurs utilisées : AM et HAC-C

Comme nous l'avons expliqué en 5.3.1, la réévaluation de la segmentation s'appuie principalement sur les modèles de locuteurs détectés. Par conséquent, une méthode de regroupement en locuteurs assistée peut être utilisée en complément de la réévaluation de la segmentation assistée afin d'aider à améliorer davantage la précision des modèles, mais également à mieux estimer leur nombre.

Nous avons décidé d'évaluer deux méthodes de regroupements assistées : la méthode AM présentée dans la partie 5.2.2 et une version modifiée de la méthode de classification ascendante hiérarchique contrainte proposée dans Bredin et Poignant [2013].

Proche des méthodes décrites dans Basu *et al.* [2004] et Davidson et Ravi [2005], la méthode de Bredin et Poignant [2013] consiste à effectuer une classification ascendante hiérarchique classique, avec pour initialisation un tour de parole par classe, mais en respectant certaines conditions exprimées sous la forme de paires de liens impossibles (*cannot-link*). Cette contrainte spécifie que deux classes particulières ne doivent pas être regroupées sous une même classe lors du processus de la classification ascendante hiérarchique. Lorsqu'un regroupement est réalisé sur une classe liée à une contrainte, la contrainte se propage alors aux nouveaux éléments associés. La classification ascendante hiérarchique contrainte consiste donc à regrouper à chaque itération les deux classes les plus proches pour lesquelles il n'y a pas de conflit.

Notre version modifiée consiste à utiliser également un autre type de contrainte, la paire de lien obligatoire (*must-link*) [Basu *et al.*, 2004; Davidson et Ravi, 2005]. Cette autre contrainte spécifie, quant à elle, que deux classes doivent être regroupées sous une même classe lors du processus de la classification ascendante hiérarchique.

Notre version modifiée se déroule en plusieurs étapes. Tout d'abord, nous extrayons nos contraintes à partir des segments vérifiés par l'humain. Plus précisément, les segments vérifiés non regroupés ensemble forment des paires de liens impossibles et ceux regroupés ensemble forment des paires de liens obligatoires. Ensuite, nous représentons chaque segment du signal audio par une classe. Les classes des segments ayant des contraintes de paires de lien obligatoire sont par la suite regroupées. Pour finir, nous effectuons la classification ascendante hiérarchique contrainte sur ce jeu de classes. La classification va effectuer des

regroupements tout en respectant les contraintes de paires de lien impossible. Une fois le regroupement terminé, nous mettons à jour la partie non vérifiée par l'annotateur.

Dans la suite de ce manuscrit, nous faisons référence à cette méthode modifiée avec le nom HAC-C pour "Classification Ascendante Hiérarchique avec Contraintes".

En raison de la difficulté que représentent les segments de parole superposée pour toutes les méthodes de SRL, la méthode HAC-C se comporte de la même manière que la méthode AM et la méthode RV concernant ces segments (voir 5.2.2.1). Par conséquent, la méthode HAC-C n'est pas mise en œuvre pour les segments de parole superposée vérifiés.

Précisons que les méthodes AM et HAC-C sont appliquées après chaque correction de regroupement en locuteurs avant la méthode RV.

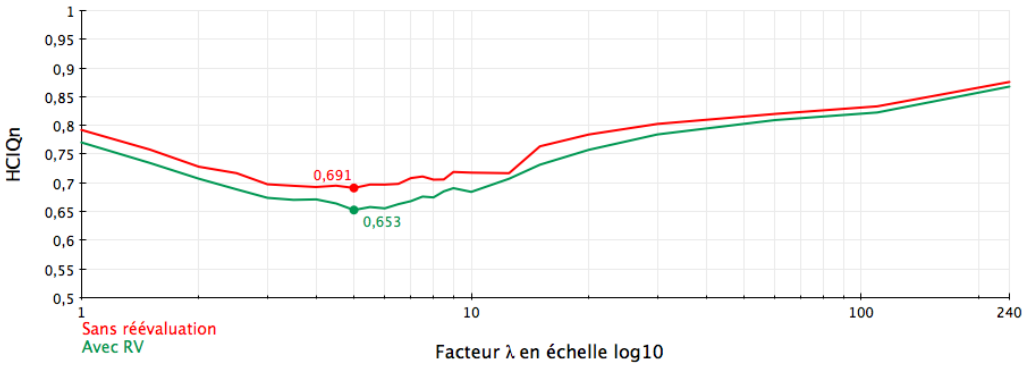
5.3.2.3 Évaluation de la méthode RV

La figure 5.9 donne le HCIQ_n de la méthode RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus. La figure 5.10, quant à elle, fournit le HCIQ_n pour le corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1.

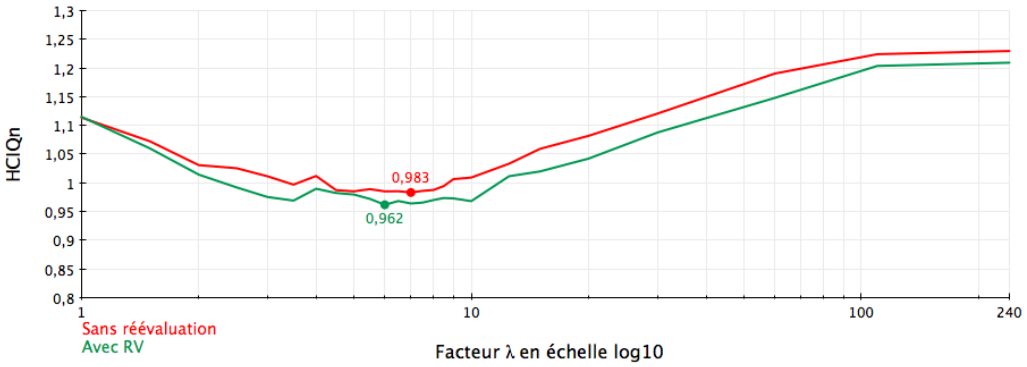
Sur ces figures et pour celles que nous allons présenter dans les parties suivantes, nous évaluons les méthodes jusqu'à un seuil λ de 240, seuil pour lequel chaque enregistrement de chaque corpus est représenté par un seul label de locuteur. Notons que tous les enregistrements du corpus ETAPE aboutissent à un seul locuteur dès le seuil de 110.

De manière globale, on remarque qu'il n'y a aucune amélioration significative avec la méthode RV. Les courbes sont très proches de la courbe "*Sans réévaluation*", à l'exception du corpus REPERE. En outre, les meilleurs HCIQ_n des courbes des méthodes assistées se trouvent à des seuils λ identiques ou très proches de la courbe "*Sans réévaluation*".

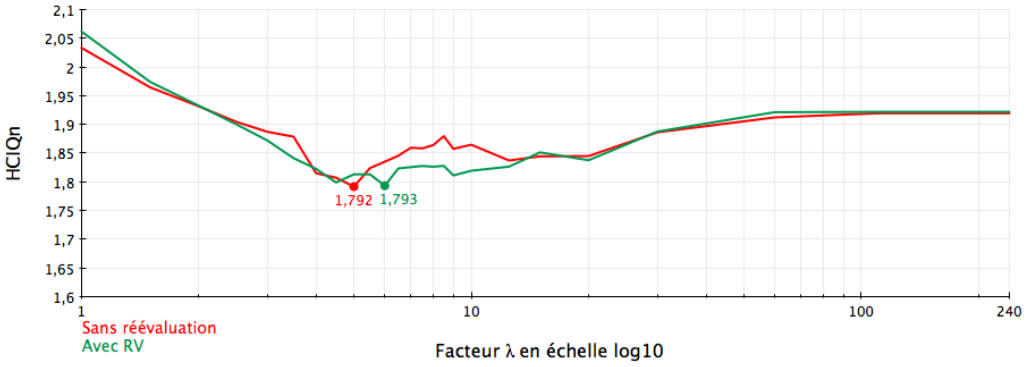
Pour les corpus ESTER et pour le corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1, on constate une très légère amélioration par rapport à un système de SRL non assisté. En effet, le gain relatif est environ de 5% pour ESTER 1, de 2% pour ESTER 2 et de 1% pour le corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE 1. Concernant le corpus ETAPE, le gain relatif est quasi nul. Pour le corpus REPERE, la méthode assistée présente une dégradation des performances. La perte relative est d'environ 3%.



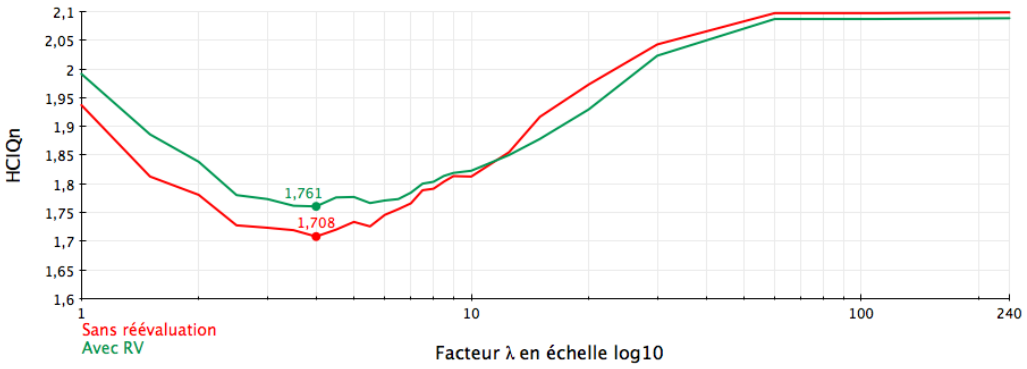
(a) ESTER 1



(b) ESTER 2



(c) ETAPE



(d) REPERE 1

FIGURE 5.9 $HCIQ_n$ de la méthode RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

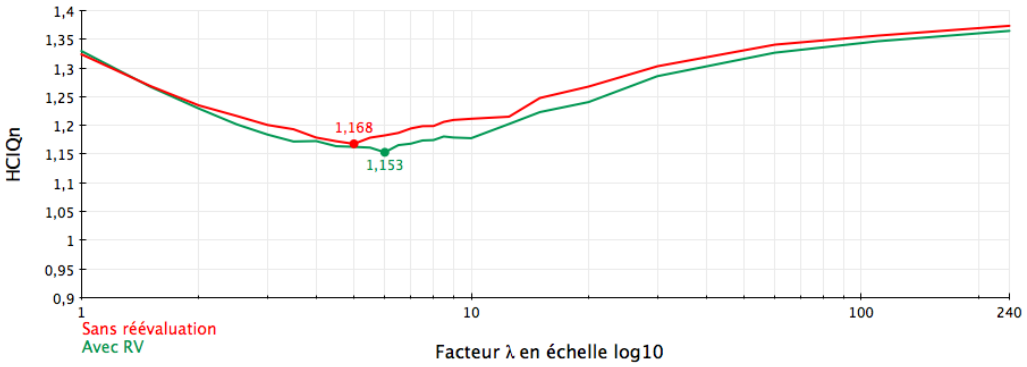


FIGURE 5.10 $HCIQ_n$ de la méthode RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

5.3.2.4 Évaluation des méthodes AM et HAC-C

La figure 5.11 illustre pour divers corpus les mesures de $HCIQ_n$ des méthodes AM et HAC-C en fonction de différents seuils λ correspondant au seuil de l'étape du regroupement mono-gaussien/BIC. La figure 5.12, quant à elle, correspond au $HCIQ_n$ pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés.

Concernant la méthode AM, on constate qu'elle améliore les performances pour tous les corpus sauf REPERE 1. On remarque également que les seuils λ optimaux pour le système avec la méthode AM sont à des seuils proches ou identiques des seuils optimaux pour le système non assisté. De plus, on constate que la courbe de la méthode AM se comporte de la même manière que la courbe "*Sans réévaluation*".

Concernant la méthode HAC-C, on remarque qu'elle améliore les performances pour tous les corpus. On remarque aussi que ses seuils λ optimaux sont proches, voire identiques, aux seuils optimaux du système non assisté. À l'exception du corpus ETAPE, la courbe de la méthode HAC-C se comporte dans l'ensemble comme la courbe "*Sans réévaluation*".

De manière générale, on constate que le système avec HAC-C est plus performant que la méthode AM. Ainsi dans la suite de cette partie, nous décidons d'étudier en détail uniquement la méthode assistée HAC-C. Les seuils λ optimaux de la méthode HAC-C se situent respectivement à 4,0, 5,0, 4,0 et 4,0 pour ESTER 1, ESTER 2, ETAPE et REPERE 1. Pour tous ces corpus regroupés, le meilleur seuil se situe à 4,0. Les tableaux 5.8, 5.9 et 5.10 donnent des informations sur la méthode HAC-C pour ces seuils. Plus précisément, le tableau 5.8 illustre le nombre d'occurrences de chaque action humaine, le tableau 5.9 les durées consacrées pour chaque type d'action et le tableau 5.10 des informations complémentaires.

Si l'on compare ces informations à celles des tableaux 4.12 et 4.13 relatifs au système non assisté, on remarque que pour les corpus ESTER et ETAPE, il y a une légère augmentation du nombre d'actions "*Supprimer une frontière*" [8] et, sauf pour ESTER 1, une légère augmentation du nombre d'actions "*Créer une frontière*" [9]. En revanche, on remarque une forte diminution du nombre d'actions "*Changer le label locuteur*" [10]. Concernant le corpus REPERE, on constate que seul le nombre d'actions "*Changer le label locuteur*" a changé. Plus précisément, il a diminué d'une cinquantaine d'occurrences.

En comparant les $HCIQ_n$ du système oracle du tableau 4.14 avec ceux de la méthode HAC-C du tableau 5.10, on remarque que la méthode HAC-C permet de réduire le $HCIQ_n$ d'environ 0,06. Plus précisément en s'appuyant sur les tableaux 4.13 et 5.9, elle permet d'économiser 20,0 min, 27,2 min, 41,2 min et 7,6 min pour respectivement les corpus ESTER 1, ESTER 2, ETAPE et REPERE [11]. Pour le corpus regroupant ESTER 1, ESTER 2, ETAPE et REPERE d'une durée de 1617 minutes (26h57), le temps économisé est de 93,9 min (1h33).

Par rapport aux tableaux 4.5 donnant des informations sur la vérité terrain, on constate que les corpus donnés en entrée sont très sous-segmentés. En effet, le nombre de changements de locuteur [12] et le nombre de segments [13] ont été divisés environ par 2, voire par 3.

8. Nombre d'occurrences de l'action *"Supprimer une frontière"* du tableau 4.12 : 453, 383, 481, 156 et 1493 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

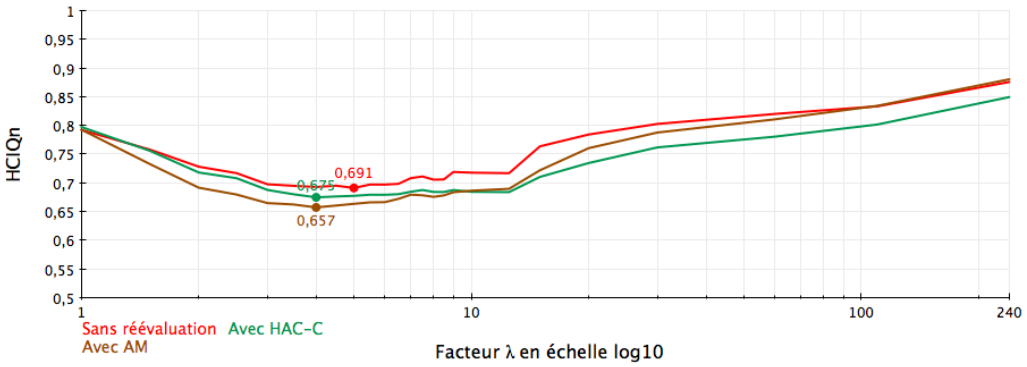
9. Nombre d'occurrences de l'action *"Créer une frontière"* du tableau 4.12 : 942, 1002, 2314, 840 et 5098 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

10. Nombre d'occurrences de l'action *"Changer le label locuteur"* du tableau 4.12 : 800, 897, 1677, 463 et 3866 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

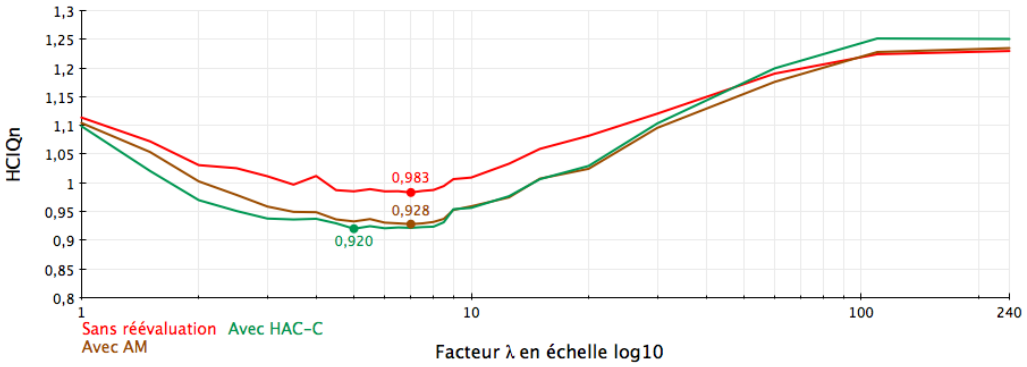
11. HCIQ du tableau 4.13 : 409,1, 422,8, 748,9, 302,3 et 1888,5 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

12. Nombre de changements de locuteur du tableau 4.5 : 1954, 1881, 3369, 1287 et 8491 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

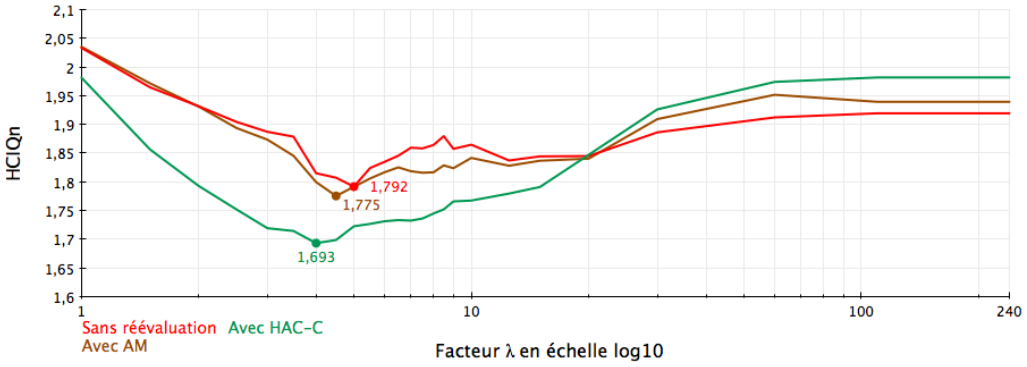
13. Nombre de segments du tableau 4.5 : 1991, 1911, 3398, 1318 et 8618 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.



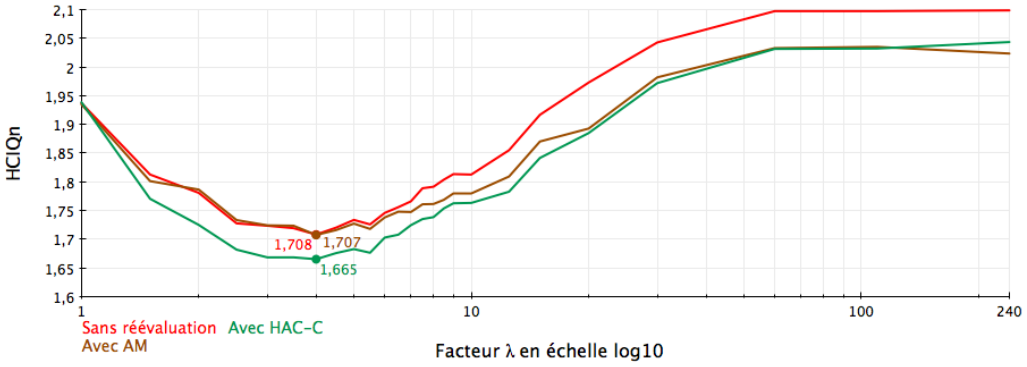
(a) ESTER 1



(b) ESTER 2



(c) ETAPE



(d) REPERE 1

FIGURE 5.11 $HCIQ_n$ de la méthode AM et de la méthode HAC-C en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

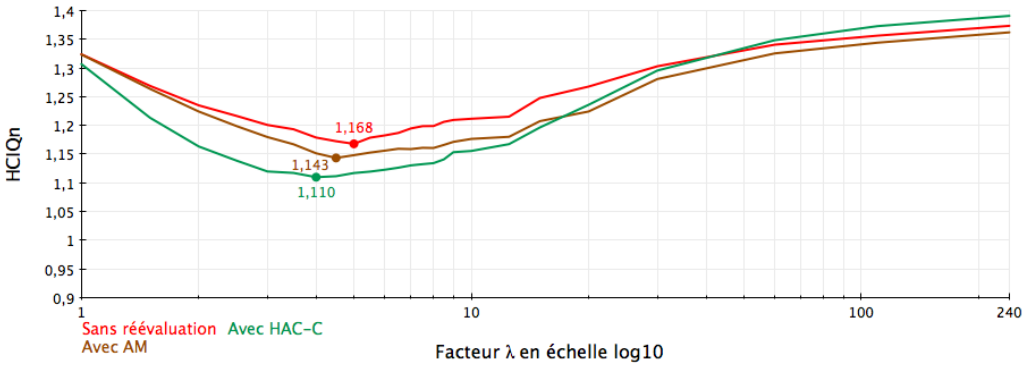


FIGURE 5.12 $HCIQ_n$ de la méthode AM et de la méthode HAC-C en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	5,0	4,0	4,0	4,0
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	679	698	1389	403	3203
Créer une frontière	907	983	2268	840	5003
Supprimer une frontière	480	405	534	156	1600
Toutes actions confondues	2448	2446	4346	1694	10998

TABLE 5.8 Nombre d'occurrences des différentes actions du système avec la méthode HAC-C pour divers corpus à la sortie du système de SRL automatique avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	5,0	4,0	4,0	4,0
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	86,0	88,4	175,9	51,0	405,7
Créer une frontière	181,4	196,6	453,6	168,0	1000,6
Supprimer une frontière	40,8	34,4	45,4	13,3	136,0
Toutes actions confondues (HCIQ)	389,1	395,6	707,7	294,7	1794,6

TABLE 5.9 Durée estimée en minutes des différentes actions du système avec la méthode HAC-C pour divers corpus à la sortie du système de SRL automatique avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	5,0	4,0	4,0	4,0
DER	18,20	11,46	19,20	11,47	16,46
$DER_{false\,alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\,speaker}$	0,35	0,18	0,44	0,85	0,38
$DER_{confusion}$	17,86	11,28	18,76	10,62	16,07
# de locuteurs	335	258	176	201	1017
# de changements de locuteur	1264	1109	1044	499	3936
# de segments	1301	1139	1073	530	4063
Moyenne de la durée des segments (s)	27,31	22,65	23,35	20,03	23,87
$HCIQ_n$ système oracle	0,691	0,983	1,792	1,708	1,168
$HCIQ_n$ RV	0,653	0,962	1,793	1,761	1,153
$HCIQ_n$ AM	0,657	0,928	1,775	1,707	1,143
$HCIQ_n$ HAC-C	0,675	0,920	1,693	1,665	1,110

TABLE 5.10 Mesures appliquées sur divers corpus à la sortie du système de SRL automatique avec le seuil λ donnant le plus faible $HCIQ_n$ pour le système avec la méthode HAC-C

5.3.2.5 Évaluation de la méthode AM + RV et de la méthode HAC-C + RV

La figure 5.13 donne pour divers corpus les valeurs du $HCIQ_n$ de la méthode AM + RV et de la méthode HAC-C + RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC. La figure 5.14, quant à elle, illustre le $HCIQ_n$ pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés.

On remarque que la méthode AM + RV améliore les performances de tous les corpus à l'exception de REPERE 1. En effet, elle offre un meilleur score $HCIQ_n$ que le système non assisté. On constate aussi que les seuils λ optimaux pour les systèmes avec AM + RV sont assez élevés et sont très variables d'un corpus à un autre. En outre, ces seuils ne sont pas toujours proches des seuils optimaux pour le système non assisté.

En ce qui concerne la méthode HAC-C + RV, on constate qu'elle améliore les performances pour tous les corpus sauf pour REPERE 1 où elle offre un $HCIQ_n$ égal au système non assisté. On remarque également que ses seuils λ optimaux sont proches de ceux du système non assisté à l'exception du corpus ETAPE. Pour le corpus ETAPE, le seuil optimal est de 110. Pour ce seuil, chaque enregistrement du corpus ne contient qu'un seul label de locuteur. Ainsi la méthode HAC-C + RV est la plus efficace pour ETAPE lorsqu'aucun système de SRL n'a été appliqué sur les données d'entrée.

Dans l'ensemble, on remarque que la méthode AM + RV et la méthode HAC-C + RV fournissent des résultats proches. Puisque la méthode HAC-C + RV est la méthode ayant

généralement des seuils λ optimaux plutôt stables d'un corpus à un autre, qu'elle n'offre aucune dégradation notable de performance pour tous les corpus et que nous avons choisi d'étudier plus en détail HAC-C dans la partie précédente, nous décidons de n'étudier que la méthode HAC-C + RV dans cette partie. Les meilleurs seuils λ de la méthode HAC-C + RV se situent à 4,0, 8,5, 110,0, 3,0 pour respectivement ESTER 1, ESTER 2, ETAPE et REPERE 1. Pour l'ensemble des quatre corpus, le meilleur seuil se situe à 5,0. Les tableaux 5.11, 5.12 et 5.13 fournissent des informations sur la méthode HAC + RV pour ces seuils.

Si l'on compare ces informations à celles des tableaux 4.12 et 4.13 concernant le système non assisté, on constate qu'il y a une forte diminution du nombre d'actions "*Changer le label locuteur*" [14] et "*Supprimer une frontière*" [15]. Pour REPERE 1, ces diminutions sont moins importantes. Concernant l'action "*Créer une frontière*" [16], on remarque une légère augmentation pour tous les corpus.

En comparant les valeurs des HCIQ_n du système oracle du tableau 4.14 avec ceux de la méthode HAC-C + RV du tableau 5.13, on constate que la méthode HAC-C + RV permet de diminuer le HCIQ_n d'environ 0,09. Plus précisément, en s'appuyant sur les tableaux 4.13 et 5.12, le temps économisé pour les corpus ESTER 1, ESTER 2 et ETAPE [17] est respectivement de 42,0 min, 54,2 min, 69,4 min et on remarque une perte de temps quasi insignifiante pour le corpus REPERE (0,1 min). Pour le corpus regroupant tous ces corpus, le temps économisé est de 135,7 min.

Par rapport au tableau 4.5 fournissant des informations sur la vérité terrain, on remarque que les données d'entrées des corpus sont très sous-segmentées au vu du nombre de changements de locuteur [18] et du nombre de segments [19] du tableau 5.13 qui sont au moins deux fois moins importants.

14. Nombre d'occurrences de l'action *"Changer le label locuteur"* du tableau 4.12 : 800, 897, 1677, 463 et 3866 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

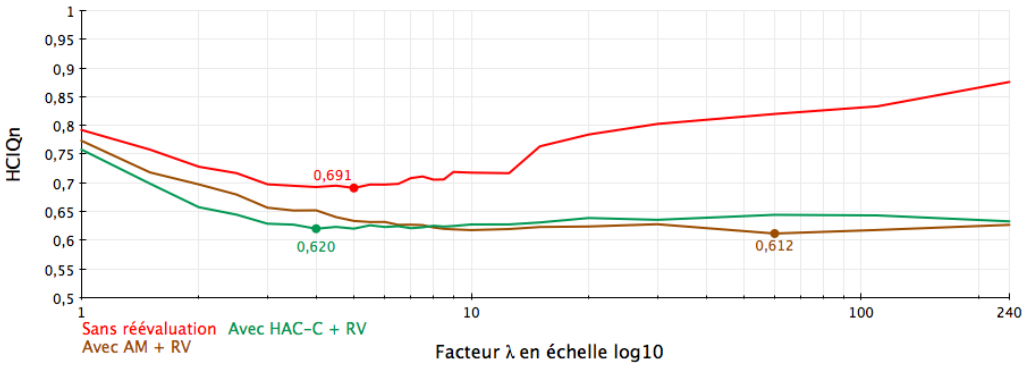
15. Nombre d'occurrences de l'action *"Supprimer une frontière"* du tableau 4.12 : 453, 383, 481, 156 et 1493 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

16. Nombre d'occurrences de l'action *"Créer une frontière"* du tableau 4.12 : 942, 1002, 2314, 840 et 5098 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

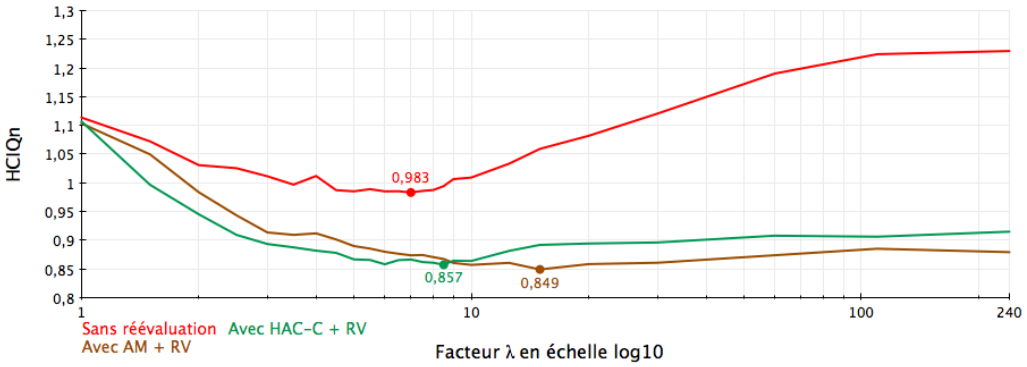
17. HCIQ du tableau 4.13 : 409,1, 422,8, 748,9, 302,3 et 1888,5 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

18. Nombre de changements de locuteur du tableau 4.5 : 1954, 1881, 3369, 1287 et 8491 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.

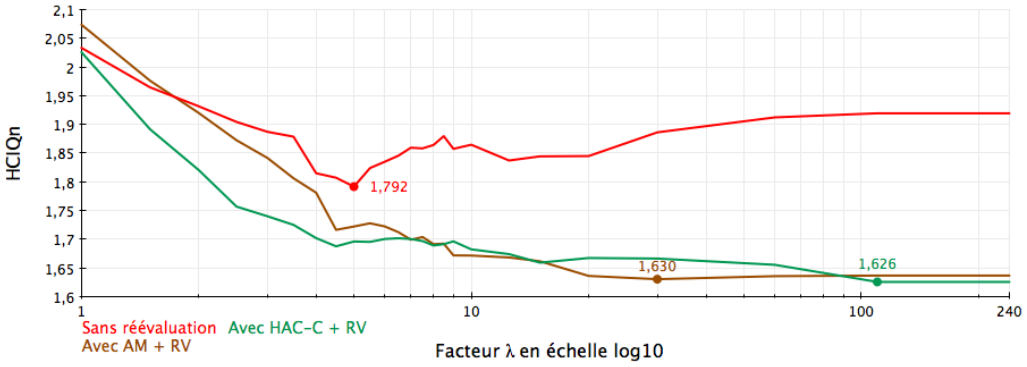
19. Nombre de segments du tableau 4.5 : 1991, 1911, 3398, 1318 et 8618 pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et TOTAL.



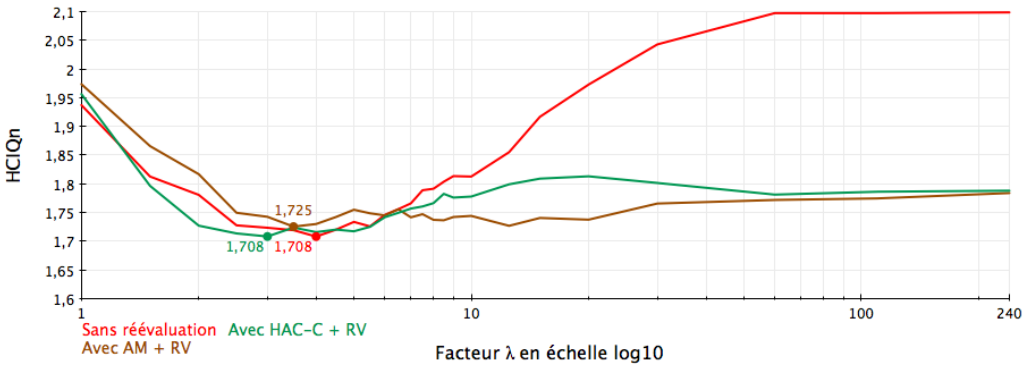
(a) ESTER 1



(b) ESTER 2



(c) ETAPE



(d) REPERE 1

FIGURE 5.13 $HClQ_n$ de la méthode AM + RV et de la méthode HAC-C + RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour divers corpus

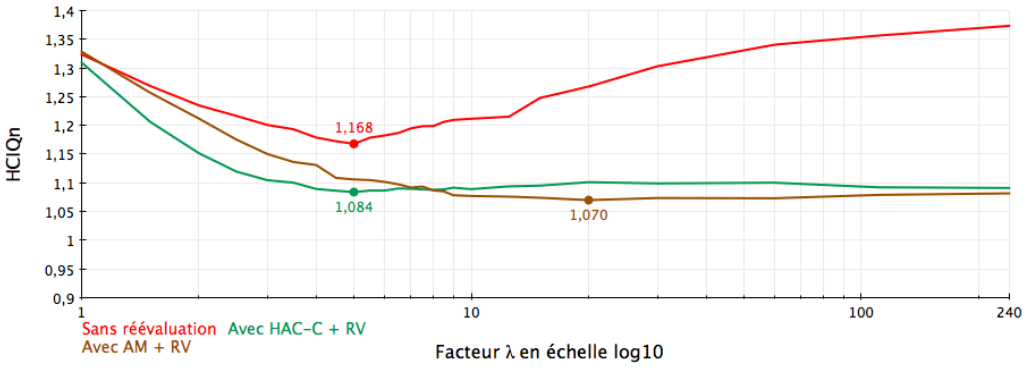


FIGURE 5.14 $HCIQ_n$ de la méthode AM + RV et de la méthode HAC-C + RV en fonction de divers seuils λ du regroupement mono-gaussien/BIC pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	8,5	110,0	3,0	5,0
Créer un label locuteur	382	360	155	295	1192
Changer le label locuteur	572	539	1073	380	2815
Créer une frontière	981	1068	2427	922	5391
Supprimer une frontière	207	124	299	87	773
Toutes actions confondues	2142	2091	3954	1684	10171

TABLE 5.11 Nombre d'occurrences des différentes actions du système avec la méthode HAC-C + RV pour divers corpus à la sortie du système de SRL automatique avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	8,5	110,0	3,0	5,0
Créer un label locuteur	80,9	76,2	32,8	62,4	252,3
Changer le label locuteur	72,4	68,3	135,9	48,1	356,6
Créer une frontière	196,2	213,6	485,4	184,4	1078,2
Supprimer une frontière	17,6	10,5	25,4	7,4	65,7
Toutes actions confondues (HCIQ)	367,1	368,6	679,5	302,4	1752,8

TABLE 5.12 Durée estimée en minutes des différentes actions du système avec la méthode HAC-C + RV pour divers corpus à la sortie du système de SRL automatique avec le seuil λ du regroupement mono-gaussien/BIC donnant le plus faible $HCIQ_n$. Durée estimée (durée moyenne de l'action $\times$ nombre d'occurrences)

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Seuil λ optimal	4,0	8,5	110,0	3,0	5,0
DER	18,20	13,02	68,08	15,89	14,33
$DER_{false\,alarm}$	0,00	0,00	0,00	0,00	0,00
$DER_{missed\,speaker}$	0,35	0,17	0,24	0,80	0,36
$DER_{confusion}$	17,86	12,85	67,84	10,23	13,97
# de locuteurs	335	197	25	234	882
# de changements de locuteur	1264	1005	100	542	3739
# de segments	1301	1035	129	573	3866
Moyenne de la durée des segments (s)	27,31	24,93	194,23	18,52	25,09
$HCIQ_n$ système oracle	0,691	0,983	1,792	1,708	1,168
$HCIQ_n$ RV	0,653	0,962	1,793	1,761	1,153
$HCIQ_n$ AM	0,657	0,928	1,775	1,707	1,143
$HCIQ_n$ HAC-C	0,675	0,920	1,693	1,665	1,110
$HCIQ_n$ AM + RV	0,612	0,849	1,630	1,725	1,070
$HCIQ_n$ HAC-C + RV	0,620	0,857	1,626	1,708	1,084

TABLE 5.13 Mesures appliquées sur divers corpus à la sortie du système de SRL automatique avec le seuil λ donnant le plus faible $HCIQ_n$ pour le système avec la méthode HAC-C + RV

5.3.2.6 Discussion

Le tableau 5.14 donne le nombre de segments de différentes durées dans les vérités terrain de divers corpus. Puisqu'on applique une tolérance de 250 ms au bord des frontières des segments, les segments des vérités terrain faisant moins de deux fois la tolérance ne sont pas corrigés par l'automate. On constate donc que la proportion de segments dont la durée n'excède pas deux secondes réellement pris en compte par l'automate est de 21,36%, 32,06%, 40,78%, 39,36% et 33,79% et la proportion de segments de parole superposée est de 5,49%, 9,45%, 0,00%, 19,87% et 6,65% pour respectivement ESTER 1, ESTER 2, ETAPE, REPERE 1 et l'ensemble des 4 corpus.

Le tableau 5.15 donne la moyenne de la durée des zones évaluées par enregistrement pour divers corpus. Comme on peut le voir, REPERE 1 est composé d'enregistrements de très courtes durées par rapport aux autres corpus.

Au vu du nombre important de segments de parole superposée et de la faible durée des enregistrements associés au corpus REPERE 1, on peut supposer que la méthode RV n'apporte aucune amélioration en $HCIQ_n$ en raison d'une difficulté à obtenir de bons modèles de locuteur pour ce corpus.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
# de segments $\leq$ 2 s	675	811	1901	671	4058
↳ dont parole superposée	212	194	245	221	872
# de segments $\leq$ 2 s et $>$ 0,5 s	358	519	1031	420	2328
↳ dont parole superposée	70	114	0	161	345
# de segments $>$ 2 s	1318	1100	1497	647	4562
↳ dont parole superposée	22	39	1	51	113
# de segments $>$ 0,5 s	1676	1619	2528	1067	6890
↳ dont parole superposée	92	153	1	212	458
Ratio # $\leq$ 2 s et $>$ 0,5 s sur # $>$ 0,5 s	21,36	32,06	40,78	39,36	33,79
Ratio # parole superposée $>$ 0,5 s sur # $>$ 0,5 s	5,49	9,45	0,00	19,87	6,65

TABLE 5.14 Nombre de segments de différentes tailles dans les vérités terrain de divers corpus

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Durée par enregistrement (min)	32	15	24	6	23

TABLE 5.15 Moyenne de la durée des zones évaluées par enregistrement pour divers corpus

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
Gain de RV en $HCIQ_n$ (%)	5,50	2,14	-0,06	-3,10	1,28
Gain de AM en $HCIQ_n$ (%)	4,92	5,59	0,95	0,06	2,14
Gain de HAC-C en $HCIQ_n$ (%)	2,31	6,41	5,52	2,52	4,97
Gain de AM + RV en $HCIQ_n$ (%)	11,43	13,63	9,04	-0,99	8,39
Gain de HAC-C + RV en $HCIQ_n$ (%)	10,27	12,82	9,26	0	7,19

TABLE 5.16 Gain relatif de différentes méthodes assistées par rapport au système oracle

À partir des résultats présentés dans les précédentes parties, on peut tirer plusieurs conclusions. Premièrement, la méthode RV sans méthode de regroupement en locuteurs assistée ne présente pas d'améliorations satisfaisantes notables. Cependant, avec une méthode de regroupement en locuteurs assistées, que ce soit AM ou HAC-C, elle présente un réel intérêt. En effet, elle économise un temps de correction humain d'environ 9,9% avec HAC-C et d'environ 10,6% avec AM (pourcentages calculés à partir des corpus ESTER 1, ESTER 2, ETAPE et TOTAL du tableau 5.16) à condition que les enregistrements soient assez longs et présentent peu de segments de parole superposée. Si cette condition n'est pas respectée, l'algorithme de Viterbi sur lequel repose la méthode RV fournit alors des modèles GMM peu efficaces et ne produit aucune réduction de temps de correction. La

méthode RV couplée avec une méthode de regroupement en locuteurs assistée permet de mieux réétiqueter les trames et de supprimer des frontières incorrectes.

Deuxièmement, au vu des formes des courbes, appliquer une correction avec la méthode RV couplée avec HAC-C ou AM sur la sortie d'une SRL automatique et non directement sur le flux audio n'apporte que quelques légères améliorations. En effet, à l'exception du corpus REPERE 1, les meilleures valeurs des HCIQ_n sont proches de celles où chaque enregistrement n'est composé que d'un seul segment (λ à 110 pour ETAPE, 240 pour les autres corpus).

Pour finir, les méthodes de regroupement en locuteurs assistées, à elles seules, réduisent le temps de correction humain, mais de manière moins importante qu'avec la méthode RV. Effectivement, la méthode HAC-C seule permet de réduire le temps de correction d'environ 4,3% et la méthode AM d'environ 2,7%. Néanmoins, ces méthodes offrent l'avantage d'offrir une amélioration sur le corpus REPERE 1.

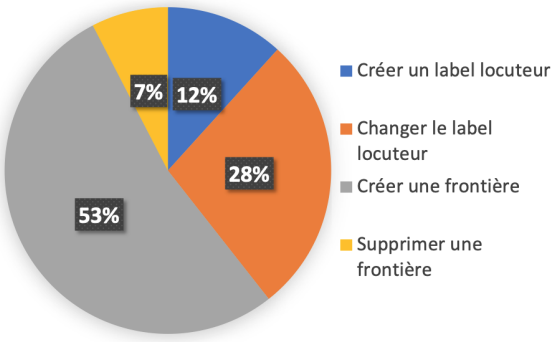


FIGURE 5.15 Proportion d'occurrences des différentes actions du système avec la méthode HAC-C + RV en fonction du seuil λ du regroupement mono-gaussien/BIC donnant le plus faible HCIQ_n pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

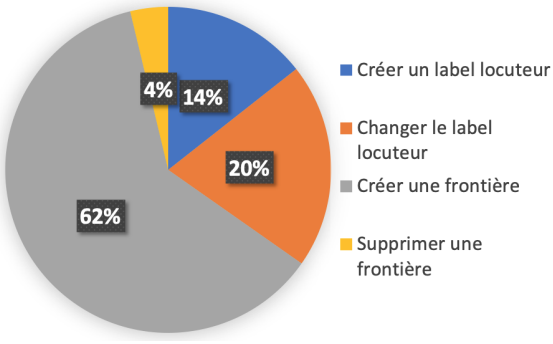


FIGURE 5.16 Durée estimée en minutes des différentes actions du système avec la méthode HAC-C + RV en fonction du seuil λ du regroupement mono-gaussien/BIC donnant le plus faible HCIQ_n pour les corpus ESTER 1, ESTER 2, ETAPE et REPERE 1 regroupés

Le coût d'une création de frontière est très élevé et les corrections liées à cette action sont fréquentes voire dominantes (figures 5.15 et 5.16). Nous devons ajouter une frontière lorsqu'un changement de locuteur a été manqué. Afin de faire décroître les erreurs de segmentation et notamment le nombre d'actions "*Créer une frontière*", plusieurs solutions ont été envisagées et essayées :

- garder les frontières issues du système d'entrée après une réévaluation avec la méthode RV ;
- couper les segments de plus de 6 secondes dans les zones où l'énergie du signal est la plus faible après chaque réévaluation avec la méthode RV ;
- ajouter au système d'entrée le regroupement i-vecteur/ILP ;
- remplacer le système d'entrée par un système à réseau de neurones.

Cependant, aucune de ces solutions visant à réduire le nombre d'actions "*Créer une frontière*" n'a surpassé les méthodes présentées. Un changement de locuteur manqué provient essentiellement de l'algorithme de Viterbi, dont les modèles GMM ne sont pas assez précis. Il serait alors intéressant d'appliquer un apprentissage pondéré de type MAP (voir 2.4.2.2) afin de favoriser les données déjà vérifiées par rapport aux données non vérifiées et ainsi espérer une réduction des corrections de segmentation. Une autre solution intéressante serait d'utiliser un système de transcription afin de positionner des frontières uniquement entre les mots. Cette solution pourrait faire baisser le nombre d'actions "*Créer une frontière*" et rendre encore plus avantageuses les corrections de regroupement en locuteurs. Cependant, un système de transcription n'est pas sans erreurs et pourrait malheureusement augmenter le nombre de frontières erronées.

Pour que la réévaluation avec la méthode RV présente un intérêt, il faut également que l'humain mette moins de temps à corriger une SRL. Puisqu'une resegmentation avec l'algorithme de Viterbi sur l'ensemble du signal audio peut être longue, deux stratégies ont été évaluées afin de réduire le temps de calcul de la méthode RV :

- rendre la main à l'utilisateur lorsque le temps de calcul est strictement supérieur au temps de lecture du prochain tour de parole de la SRL de référence ;
- ne réévaluer que les 10 premières secondes de la partie de la SRL non vérifiée.

Ces deux stratégies permettent d'effectuer une adaptation en temps réel avec une faible incidence sur le $HCIQ_n$ (perte d'environ 0,1 point).

5.4 Bilan

Dans ce chapitre, nous avons proposé des systèmes assistés réduisant le coût d'intervention humain total pour obtenir des SRL sans erreurs.

Lorsque la segmentation est parfaite, nous avons proposé la méthode AM. Cette méthode propose une diminution du temps corrections humain d'environ 16,3%.

Lorsque la segmentation n'est pas parfaite, nous avons proposé diverses méthodes : AM, HAC-C et RV. Les méthodes AM et HAC-C, qui réévaluent uniquement le regroupement en locuteurs, diminuent le temps de correction humain respectivement de 2,7% et 4,3%. La méthode RV, qui réévalue autant la segmentation que le regroupement en locuteurs, offre une diminution du temps de correction d'environ 10% lorsqu'elle est couplée à AM ou HAC-C et sur des enregistrements assez longs présentant peu de parole superposée.

Chapitre 6

Conclusions et perspectives

6.1 Conclusions

Dans le cadre de l'analyse automatique de documents et de l'aide à l'annotation de documents audiovisuels, ce manuscrit s'est articulé autour de la segmentation et du regroupement en locuteurs dans des documents audiovisuels, en interaction avec des annotateurs humains. Plus précisément, il a eu pour but de proposer des outils d'aide à la documentation à même d'interagir en temps réel avec un annotateur, pour améliorer la qualité des annotations pouvant être produites en un temps réduit.

Dans ce manuscrit, nous avons présenté une nouvelle mesure pour évaluer le coût de correction humain d'un système entièrement automatique ou assisté, la mesure HCIQ. Même si elle a été essentiellement présentée dans le contexte de la SRL, elle peut être appliquée à nombre d'autres domaines tels que la transcription ou la traduction. Nous avons aussi introduit un jeu d'actions humaines pour corriger une SRL ainsi qu'un protocole pour évaluer les coûts de chaque action. Comme pour la mesure HCIQ, le protocole peut être appliqué pour estimer les coûts d'actions appartenant à d'autres domaines. De plus, nous avons proposé un automate simulant les corrections de SRL d'un annotateur humain, qui est disponible dans l'outil *S4D* [Broux *et al.*, 2018a]. Nous avons montré qu'annoter entièrement des données est plus chronophage pour un humain que de corriger ces mêmes données issues d'un système automatique de SRL.

Nous avons proposé des systèmes assistés réduisant le coût d'intervention humain pour obtenir des SRL sans erreurs. Lorsque la segmentation est parfaite, la méthode proposée permet de diminuer le temps de corrections humain d'environ 16,3%. Lorsque la segmentation est imparfaite, nous avons proposé trois méthodes : AM, HAC-C et RV. Les méthodes AM et HAC-C réévaluant le regroupement en locuteurs permettent de réduire le temps de correction humain de 2,7% et 4,3% respectivement. La méthode RV réévaluant

la segmentation et le regroupement en locuteurs permet de diminuer le temps de correction de 10% au minimum lorsqu'elle est utilisée conjointement avec la méthode AM ou HAC-C et sur des enregistrements assez longs présentant peu de parole superposée.

Corpus	ESTER 1	ESTER 2	ETAPE	REPERE 1	TOTAL
$HCIQ_n$ sans système automatique	0,94	1,24	1,93	2,14	1,41
$HCIQ_n$ reg. i-vecteur/ILP (dernière étape)	0,67	0,96	1,78	1,68	1,15
$HCIQ_n$ reseg. par Viterbi (étape 5)	0,69	0,98	1,79	1,71	1,17
↳ avec AM + RV	0,61	0,85	1,63	1,72	1,07
↳ avec HAC-C + RV	0,62	0,86	1,63	1,71	1,08

TABLE 6.1 Récapitulatif des $HCIQ_n$ à différentes étapes du système automatique présenté en 2.9 avec ou sans méthodes assistées

Concernant l'INA qui emploie actuellement des documentalistes professionnels pour annoter entièrement manuellement les notices documentaires, des solutions peuvent être appliquées pour obtenir un meilleur rendement.Tout d'abord, appliquer toutes les étapes du système entièrement automatique introduit en 2.9 et corriger les erreurs restantes permet de réduire de manière non négligeable le temps de travail humain. En effet, le $HCIQ_n$, soit le ratio du nombre de corrections à faire pour une unité de temps, décroît en moyenne d'environ 20%. Enfin, il est possible d'appliquer sur les données de l'INA le système entièrement automatique introduit en 2.9 jusqu'à la $5^{ème}$ étape et de corriger les erreurs restantes en utilisant un système assisté reposant sur la méthode RV avec AM ou HAC-C. Cette dernière solution permet de réduire le temps de correction humain par minute audio en moyenne d'environ 10%. Le tableau 6.1 montre les $HCIQ_n$ de ces deux solutions.

6.2 Perspectives

Il serait intéressant d'étudier davantage les méthodes assistées introduites dans ce manuscrit. Pour les méthodes AM et HAC-C, il serait pertinent d'étudier si utilisées conjointement elles permettent d'améliorer la prise de décision sur les segments à regrouper. Pour la méthode RV, il faudrait étudier si un apprentissage pondéré de type MAP, favorisant les données déjà vérifiées aux données non vérifiées, permet de réduire davantage le temps de correction humain total. Pour toutes ces méthodes assistées, il faudrait étudier si des versions reposant sur le modèle x-vecteur (voir annexe A) ou d'autres modèles issus de réseaux de neurones offrent de meilleures performances.

Puisqu'on a relevé que la correction de la segmentation est problématique pour nos méthodes assistées, il serait intéressant de développer des systèmes de SRL optimisant la détection de frontière en dépit d'erreurs de regroupement en locuteurs. Les sorties de ces

systèmes pourront ensuite être transmises aux méthodes assistées afin qu'elles en tiennent compte.

Pour les expériences que nous avons menées, nous avons décidé qu'une émission est corrigée du début à la fin en suivant l'ordre temporel afin de valider l'annotation automatique faite a posteriori. Cette condition expérimentale oblige un humain à écouter tout le signal audio. Ainsi, il serait également intéressant d'étudier un cas d'apprentissage actif (voir 3.6.2) où un système de SRL assisté demande à un humain de corriger des zones qu'il juge particulièrement erronées afin de pouvoir améliorer ses performances. La tâche d'identification de ces zones peut se révéler aussi complexe que la tâche de SRL elle-même. On peut également se demander si la capacité d'identifier automatiquement une zone à risque n'offre pas, par la même occasion, la possibilité de la corriger automatiquement. La mesure MSR présentée en 4.1.1 appliquée avec la mesure HCIQ sera particulièrement utile pour évaluer un tel système.

Pour le LIUM, les apports de ce manuscrit servent de base de réflexion dans le projet *Chist-Era - Autonomous Lifelong Learning IntelligEnt Systems*[1] (ALLIES). Ce projet vise le développement de systèmes autonomes, capables de maintenir leurs performances dans le temps selon un scénario d'apprentissage avec ou sans intervention humaine.

1. Lien de la description du projet : https ://projets-lium.univ-lemans.fr/allies/

Bibliographie

Ali, A., Bell, P., Glass, J., Messaoui, Y., Mubarak, H., Renals, S. et Zhang, Y. : The mgb-2 challenge : Arabic multi-dialect broadcast media recognition. Dans *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*, pages 279–284. IEEE, 2016.

Ali, A., Vogel, S. et Renals, S. : Speech recognition challenge in the wild : Arabic mgb-3. Dans *Proceedings of the 2017 IEEE International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 316–322. IEEE, 2017.

Alquier, E., Carrive, J. et Lalande, S. : Une production documentaire au service de l'usage : Perspectives d'automatisation autour des outils de consultation et de production documentaire de l'Institut national de l'audiovisuel (INA). *Document Numerique*, 20 (2):115–136, 8 2018.

Amengual, J. C., Castaño, A., Castellanos, A., Jiménez, V. M., Llorens, D., Marzal, A., Prat, F., Vilar, J. M., Benedi, J. M., Casacuberta, F. et others : The EuTrans spoken language translation system. *Machine Translation*, 15(1-2):75–103, 2000.

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

Anguera, X. et Bonastre, J. F. : A novel speaker binary key derived from anchor models. Dans *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2118–2121. ISCA, 2010.

Anguera, X. et Bonastre, J. F. : Fast speaker diarization based on binary keys. Dans *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4428–4431. IEEE, 2011.

Arora, S., Nyberg, E. et Rosé, C. P. : Estimating annotation cost for active learning in a multi-annotator environment. Dans *Proceedings of the 2009 ACL International Workshop on Active Learning for Natural Language Processing*, pages 18–26. ACL, 2009.

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E. et others : Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.

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

Barras, C., Geoffrois, E., Wu, Z. et Liberman, M. : Transcriber : Development and use of a tool for assisting speech corpora production. *Speech Communication*, 33(1):5–22, 2001.

Barras, C., Meignier, S. et Gauvain, J. L. : Unsupervised online adaptation for speaker verification over the telephone. Dans *Proceedings of the 2004 ISCA Workshop on Speaker and Language Recognition (Odyssey)*, page 4. ISCA, 2004a.

Barras, C., Zhu, X., Meignier, S. et Gauvain, J. L. : Improving speaker diarization. Dans *Proceedings of the Fall 2004 NIST International Workshop on Rich Transcription (RT)*. NIST, 2004b.

Basu, S., Banerjee, A. et Mooney, R. J. : Active semi-supervision for pairwise constrained clustering. Dans *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, pages 333–344. SIAM, 2004.

Baum, L. E. et Petrie, T. : Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

Bazillon, T., Estève, Y. et Luzzati, D. : Manual vs assisted transcription of prepared and spontaneous speech. Dans *Proceedings of the 6th ELRA International Conference on Language Resources and Evaluation (LREC)*. ELRA, 2008.

Bell, P., Gales, M. J., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M. et others : The MGB challenge : Evaluating multi-genre broadcast media recognition. Dans *Proceedings of the 2015 IEEE International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 687–693. IEEE, 2015.

Bender, O., Hasan, S., Vilar, D., Zens, R. et Ney, H. : Comparison of generation strategies for interactive machine translation. Dans *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 33–40. EAMT, 2005.

Betser, M., Bimbot, F., Ben, M. et Gravier, G. : Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted gmms. Dans *Proceedings of the 8th ISCA International Conference on Spoken Language Processing (INTERSPEECH-ICSLP)*, pages 2329–2332. ISCA, 2004.

Bluche, T., Ney, H. et Kermorvant, C. : The LIMSI handwriting recognition system for the HTRtS 2014 contest. Dans *Proceedings of the 13th IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 86–90. IEEE, 2015.

Bonastre, J. F., Delacourt, P., Fredouille, C., Merlin, T. et Wellekens, C. : A speaker tracking system based on speaker turn detection for NIST evaluation. Dans *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 1177–1180. IEEE, 2000.

Bousquet, P. M., Larcher, A., Matrouf, D., Bonastre, J. F. et Plchot, O. : Variance-spectra based normalization for i-vector standard and probabilistic linear discriminant analysis. Dans *Proceedings of the 2012 ISCA International Workshop on Speaker and Language Recognition (Odyssey)*. ISCA, 2012.

Bousquet, P. M., Matrouf, D. et Bonastre, J. F. : Intersession compensation and scoring methods in the i-vectors space for speaker recognition. Dans *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2011.

Bredin, H. et Poignant, J. : Integer linear programming for speaker diarization and cross-modal identification in TV broadcast. Dans *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2013.

Broux, P. A., Desnous, F., Larcher, A., Petitrenaud, S., Carrive, J. et Meignier, S. : S4D : Speaker diarization toolkit in python. Dans *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2018a.

Broux, P. A., Doukhan, D., Petitrenaud, S., Meignier, S. et Carrive, J. : An active learning method for speaker identity annotation in audio recordings. Dans *Proceedings of the 1st EurAI International Workshop on Multimodal Media Data Analytics (MMDA)*. EurAI, 2016.

Broux, P. A., Doukhan, D., Petitrenaud, S., Meignier, S. et Carrive, J. : Computer-assisted speaker diarization : How to evaluate human corrections. Dans *Proceedings of the 11th ELRA International Conference on Language Resources and Evaluation (LREC)*. ELRA, 2018b.

Bruynooghe, M. : Classification ascendante hiérarchique des grands ensembles de données : Un algorithme rapide fondé sur la construction des voisinages réductibles. *Cahiers de l'analyse des données*, 3(1):7–33, 1978.

Budnik, M., Poignant, J., Besacier, L. et Quénot, G. : Automatic propagation of manual annotations for multimodal person identification in TV shows. Dans *Proceedings of the 12th IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–4. IEEE, 2014.

Calliope, L. et Fant, G. : *La parole et son traitement automatique*. Masson Paris, 1989.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M. et Zaidan, O. F. : Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. Dans *Proceedings of the Joint 5th ACL International Workshop on Statistical Machine Translation (WMT) and MetricsMATR*, pages 17–53. ACL, 2010.

Campbell, W. M., Sturim, D. E. et Reynolds, D. A. : Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, 2006.

Cardinal, P., Boulianne, G., Comeau, M. et Boisvert, M. : Real-time correction of closed-captions. Dans *Proceedings of the 45th ACL Annual Meeting on Interactive Poster and Demonstration Sessions*, pages 113–116. ACL, 2007.

Casacuberta, F., Civera, J., Cubel, E., Lagarda, A. L., Lapalme, G., Macklovitch, E. et Vidal, E. : Human interaction for high-quality machine translation. *Communications of the ACM*, 52(10):135–138, 2009.

Cettolo, M., Vescovi, M. et Rizzi, R. : Evaluation of BIC-based algorithms for audio segmentation. *Computer Speech & Language*, 19(2):147–170, 2005.

Chapelle, O., Scholkopf, B. et Zien, A. : Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Chen, S., Gopalakrishnan, P. et others : Speaker, environment and channel change detection and clustering via the bayesian information criterion. Dans *Proceedings of the 1998 DARPA International Workshop on Broadcast News Transcription and Understanding*, volume 8, pages 127–132. DARPA, 1998.

Civera, J., Cubel, E., Lagarda, A. L., Picó, D., González, J., Vidal, E., Casacuberta, F., Vilar, J. M. et Barrachina, S. : From machine translation to computer assisted translation using finite-state models. Dans *Proceedings of the 2004 ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2004a.

Civera, J., Vilar, J. M., Cubel, E., Lagarda, A. L., Barrachina, S., Casacuberta, F., Vidal, E., Picó, D. et González, J. : A syntactic pattern recognition approach to computer assisted translation. Dans *Proceedings of the Joint 2004 IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 207–215. IAPR, 2004b.

Cubel, E., Civera, J., Vilar, J. M., Lagarda, A. L., Casacuberta, F., Vidal, E., Picó, D., González, J. et Rodríguez, L. : Finite-state models for computer assisted translation. Dans *Proceedings of the 16th EurAI European Conference on Artificial Intelligence (ECAI)*, pages 586–590. EurAI, 2004.

Cubel, E., González, J., Lagarda, A., Casacuberta, F., Juan, A. et Vidal, E. : Adapting finite-state translation to the TransType2 project. Dans *Proceedings of the 8th International Workshop of the European Association for Machine Translation*. EAMT, 2003.

Dasgupta, S. : The two faces of active learning. Dans *Proceedings of the 12th International Conference on Discovery Science (DS)*, pages 35–55, 2009.

Davidson, I. et Ravi, S. S. : Agglomerative hierarchical clustering with constraints : Theoretical and empirical results. Dans *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 59–70, 2005.

Davis, S. B. et Mermelstein, P. : Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

Defays, D. : An efficient algorithm for a complete link method. *The Computer Journal*, 20 (4):364–366, 1977.

Dehak, N. : *Discriminative and generative approaches for long-and short-term speaker characteristics modeling : Application to speaker verification*. Thèse de doctorat, Université du Québec à Montréal, 2009.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P. et Ouellet, P. : Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788–798, 2011.

Delacourt, P. : *La segmentation et le regroupement par locuteurs pour l'indexation de documents audio*. Thèse de doctorat, École nationale supérieure des télécommunications de Paris, 2000.

Delacourt, P. et Wellekens, C. J. : DISTBIC : A speaker-based segmentation for audio data indexing. *Speech Communication*, 32(1-2):111–126, 2000.

Delgado, H., Anguera, X., Fredouille, C. et Serrano, J. : Fast single-and cross-show speaker diarization using binary key speaker modeling. *IEEE Transactions on Audio, Speech and Language Processing*, 23(12):2286–2297, 2015.

Dempster, A. P., Laird, N. M. et Rubin, D. B. : Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, pages 1–38, 1977.

Dix, A. : Human-computer interaction. Dans *Encyclopedia of database systems*, pages 1327–1331. Springer, 2009.

Dufour, R., Jousse, V., Estève, Y., Béchet, F. et Linarès, G. : Spontaneous speech characterization and detection in large audio database. Dans *Proceedings of the 13th International Conference on Speech and Computer (SPECOM)*, 2009.

Dupuy, G. : *Les collections volumineuses de documents audiovisuels : Segmentation et regroupement en locuteurs*. Thèse de doctorat, Université du Maine, 2015.

Dupuy, G., Meignier, S., Deléglise, P. et Esteve, Y. : Recent improvements on ILP-based clustering for broadcast news speaker diarization. Dans *Proceedings of the 2014 ISCA International Workshop on Speaker and Language Recognition (Odyssey)*. ISCA, 2014a.

Dupuy, G., Meignier, S. et Esteve, Y. : Is incremental cross-show speaker diarization efficient for processing large volumes of data ? Dans *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014b.

Dupuy, G., Rouvier, M., Meignier, S. et Esteve, Y. : I-vectors and ILP clustering adapted to cross-show speaker diarization. Dans *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2012.

Eakins, J. et Graham, M. E. : Content-based image retrieval. http://www.leeds.ac.uk/educol/documents/00001240.htm#_Toc442192734, 1999.

Foster, G., Isabelle, P. et Plamondon, P. : Target-text mediated interactive machine translation. *Machine Translation*, 12(1-2):175–194, 1997.

Foster, G., Langlais, P. et Lapalme, G. : User-friendly text prediction for translators. Dans *Proceedings of the 2002 ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 148–155. ACL, 2002.

Furui, S. : Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2):254–272, 1981.

Galibert, O. : Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. Dans *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1131–1134. ISCA, 2013.

Galibert, O. et Kahn, J. : The first official REPERE evaluation. Dans *Proceedings of the 1st ISCA International Workshop on Speech, Language and Audio in Multimedia (SLAM)*. ISCA, 2013.

Galliano, S., Gravier, G. et Chaubard, L. : The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. Dans *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2009.

Gauvain, J. L. et Lee, C. H. : Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

Geoffrois, E. : Evaluating interactive system adaptation. Dans *Proceedings of the 10th ELRA International Conference on Language Resources and Evaluation (LREC)*. ELRA, 2016.

Gish, H., Siu, M. H. et Rohlicek, R. : Segregation of speakers for speech recognition and speaker identification. Dans *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 873–876. IEEE, 1991.

Goëau, H. : *Structuration de collections d'images par apprentissage actif crédibiliste*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I, 2009.

Gravier, G., Adda, G., Paulson, N., Carré, M., Giraudel, A. et Galibert, O. : The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. Dans *Proceedings of the 8th ELRA International Conference on Language Resources and Evaluation (LREC)*. ELRA, 2012.

Greenberg, C. S., Martin, A. F., Barr, B. N. et Doddington, G. R. : Report on performance results in the NIST 2010 speaker recognition evaluation. Dans *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2011a.

Greenberg, C. S., Martin, A. F., Brandschain, L., Campbell, J. P., Cieri, C., Doddington, G. R. et Godfrey, J. J. : Human assisted speaker recognition in NIST SRE10. Dans *Proceedings of the 2010 ISCA International Workshop on Speaker and Language Recognition (Odyssey)*, page 32. ISCA, 2010.

Greenberg, C. S., Martin, A. F., Doddington, G. R. et Godfrey, J. J. : Including human expertise in speaker recognition systems : Report on a pilot evaluation. Dans *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5896–5899. IEEE, 2011b.

Guillaumin, M., Verbeek, J. et Schmid, C. : Is that you ? Metric learning approaches for face identification. Dans *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV)*, pages 498–505. IEEE, 2009.

Haton, J. P., Cerisara, C., Fohr, D., Laprie, Y. et Smaïli, K. : *Reconnaissance automatique de la parole : Du Signal à son Interprétation*. Dunod, 2006.

He, Q.-H., Yang, J.-C., Li, Y.-X., He, J., Zhang, X.-Y. et Li, W. : Combining gmm, jensen's inequality and bic for speaker indexing. *Electronics letters*, 46(9):654–655, 2010.

Hermansky, H. et Sharma, S. : Temporal patterns (TRAPS) in ASR of noisy speech. Dans *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 289–292. IEEE, 1999.

Hernandez-Sierra, G., Calvo, J. R., Bonastre, J. F. et Bousquet, P. M. : Session compensation using binary speech representation for speaker recognition. *Pattern Recognition Letters*, 49:17–23, 2014.

Himawan, I., Rahman, M. H., Sridharan, S. et Fookes, C. B. : QUT system description to the NIST SRE 2018 campaign. Dans *Proceedings of the 2018 NIST Speaker Recognition Evaluation (SRE)*. NIST, 2018.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et others : Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Huang, Z., García-Perera, L. P., Villalba, J., Povey, D. et Dehak, N. : Jhu diarization system description. Dans *Proceedings of the 4th International Conference on Speech and Language Technologies for Iberian Languages (IberSPEECH)*, pages 236–239. ISCA & INESC-ID & RTTH, 2018.

Jaksch, T., Ortner, R. et Auer, P. : Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Jarvis, R. A. : Special feature : An interactive minicomputer laboratory for graphics, image processing and pattern recognition. *IEEE Computer*, 7(10):49–60, 1974.

Jiang, Y., Lee, K. A., Tang, Z., Ma, B., Larcher, A. et Li, H. : PLDA modeling in i-vector and supervector space for speaker verification. Dans *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2012.

Kahn, J. : *Parole de locuteur : Performance et confiance en identification biométrique vocale*. Thèse de doctorat, Université d'Avignon, 2011.

Kakade, S. M., Shalev-Shwartz, S. et Tewari, A. : Efficient bandit algorithms for online multiclass prediction. Dans *Proceedings of the 25th ACM International Conference on Machine Learning (ICML)*, pages 440–447. ACM, 2008.

Kato, T. : Database architecture for content-based image retrieval. Dans *Proceedings of the 1992 SPIE International Symposium on Electronic Imaging : Science and Technology*, volume 1662, pages 112–124. SPIE, 1992.

Kenny, P. : Bayesian speaker verification with heavy-tailed priors. Dans *Proceedings of the 2010 ISCA International Workshop on Speaker and Language Recognition (Odyssey)*, page 14. ISCA, 2010.

Kenny, P., Boulianne, G., Ouellet, P. et Dumouchel, P. : Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1435–1447, 2007.

Kenny, P., Mihoubi, M. et Dumouchel, P. : New map estimators for speaker recognition. Dans *Proceedings of the 8th ISCA European Conference on Speech Communication and Technology (INTERSPEECH-EUROSPEECH)*. ISCA, 2003.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V. et Dumouchel, P. : A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 16(5):980–988, 2008.

Koshinaka, T., Nagatomo, K. et Shinoda, K. : Online speaker clustering using incremental learning of an ergodic hidden markov model. *IEICE Transactions on Information and Systems*, 95(10):2469–2478, 2012.

Kuhn, R., Nguyen, P., Junqua, J. C., Goldwasser, L., Niedzielski, N., Fincke, S., Field, K. et Contolini, M. : Eigenvoices for speaker adaptation. Dans *Proceedings of the 5th ISCA International Conference on Spoken Language Processing (ICSLP)*. ISCA, 1998.

Laird, N. : The EM algorithm. *Computational Statistics*, 9:509–520, 1993.

Langlais, P., Foster, G. et Lapalme, G. : Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4):267–294, 2000.

Langlais, P. et Lapalme, G. : TransType : Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17(2):77–98, 2002.

Larcher, A. : *Modèles acoustiques à structure temporelle renforcée pour la vérification du locuteur embarquée*. Thèse de doctorat, Université d'Avignon, 2009.

Larcher, A., Lee, K. A. et Meignier, S. : An extensible speaker identification sidekit in python. Dans *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5095–5099. IEEE, 2016.

Laurent, A. : *Auto-adaptation et reconnaissance automatique de la parole*. Thèse de doctorat, Université du Maine, 2010.

Laurent, A., Meignier, S., Merlin, T. et Deléglise, P. : Computer-assisted transcription of speech based on confusion network reordering. Dans *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4887. IEEE, 2011.

Le, V. B., Mella, O. et Fohr, D. : Speaker diarization using normalized cross likelihood ratio. Dans *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2007.

Le Lan, G. : *Analyse en locuteurs de collections de documents multimédia*. Thèse de doctorat, Université du Maine, 2017.

Macklovitch, E. : Transtype2 : The last word. Dans *Proceedings of the 5th ELRA International Conference on Language Resources and Evaluation (LREC)*, pages 167–172. ELRA, 2006.

Mangu, L., Brill, E. et Stolcke, A. : Finding consensus in speech recognition : Word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400, 2000.

Martin, A. F. et Greenberg, C. S. : The NIST 2010 speaker recognition evaluation. Dans *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2010.

Matějka, P., Glembek, O., Castaldo, F., Alam, M. J., Plchot, O., Kenny, P., Burget, L. et Černocky, J. : Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. Dans *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4828–4831. IEEE, 2011.

Matejka, P., Schwarz, P. et others : 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, 2007.

Matrouf, D., Bonastre, J. F. et Mezaache, S. E. : Factor analysis multi-session training constraint in session compensation for speaker verification. Dans *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2008.

Matrouf, D., Scheffer, N., Fauve, B. et Bonastre, J. F. : A straightforward and efficient implementation of the factor analysis model for speaker verification. Dans *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2007.

McCowan, I. A., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P. et Bourlard, H. : On the use of information retrieval measures for speech recognition evaluation. Rapport technique, IDIAP, 2004.

McLaren, M., Ferrer, L., Castan, D. et Lawson, A. : The 2016 speakers in the wild speaker recognition evaluation. Dans *Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 823–827. ISCA, 2016a.

McLaren, M., Ferrer, L. et Lawson, A. : Exploring the role of phonetic bottleneck features for speaker and language recognition. Dans *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5575–5579. IEEE, 2016b.

McLaren, M., Lei, Y. et Ferrer, L. : Advances in deep neural network approaches to speaker recognition. Dans *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4814–4818. IEEE, 2015.

Meignier, S., Bonastre, J. F. et Igounet, S. : E-HMM approach for learning and adapting sound models for speaker indexing. Dans *Proceedings of the 2001 ISCA Workshop on Speaker and Language Recognition (Odyssey)*. ISCA, 2001.

Mermelstein, P. : Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, 116:374–388, 1976.

Mohamed, A.-r., Dahl, G. E., Hinton, G. et others : Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):14–22, 2012.

Moraru, D. : *Segmentation en locuteurs de documents audio et audiovisuels : application à la recherche d'information multimédia*. Thèse de doctorat, Institut polytechnique de Grenoble, 2004.

Munkres, J. : Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

Nanjo, H., Akita, Y. et Kawahara, T. : Computer assisted speech transcription system for efficient speech archive. Dans *Proceedings of the 9th WESPAC Western Pacific Conference on Acoustics (WESPAC)*. WESPAC, 2006.

Nepveu, L., Lapalme, G., Langlais, P. et Foster, G. : Adaptive language and translation models for interactive machine translation. Dans *Proceedings of the 2004 ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2004.

Ney, H., Ortmanns, S. et Lindam, I. : Extensions to the word graph method for large vocabulary continuous speech recognition. Dans *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1791–1794. IEEE, 1997.

NIST : RT-2002 evaluation plan, April 2002.

NIST : The rich transcription spring 2003 (RT-03S) evaluation plan, February 2003.

NIST : Fall 2004 rich transcription (RT-04F) evaluation plan, October 2004a.

NIST : Spring 2004 (RT-04S) rich transcription meeting recognition evaluation plan, February 2004b.

NIST : Spring 2005 (RT-05S) rich transcription meeting recognition evaluation plan, May 2005.

NIST : Spring 2006 (RT-06S) rich transcription meeting recognition evaluation plan, February 2006.

NIST : Spring 2007 (RT-07) rich transcription meeting recognition evaluation plan, February 2007.

NIST : The 2009 (RT-09) rich transcription meeting recognition evaluation plan, February 2009.

NIST : NIST 2018 speaker recognition evaluation plan. https://www.nist.gov/sites/default/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf, August 2018.

Och, F. J., Zens, R. et Ney, H. : Efficient search for interactive statistical machine translation. Dans *Proceedings of the 10th ACL Conference on European Chapter*, pages 387–393. ACL, 2003.

Ogata, J., Goto, M. et Eto, K. : Automatic transcription for a web 2.0 service to search podcasts. Dans *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2007.

Orosanu, L. et Jouvet, D. : Adding new words into a language model using parameters of known words with similar behavior. Dans *Proceedings of the 2015 IEEE International Conference on Natural Language and Speech Processing (NLSP)*. IEEE, 2015.

Ortmanns, S., Ney, H. et Aubert, X. : A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech & Language*, 11(1):43–72, 1997.

Parris, E. S. et Carey, M. J. : Speaker verification using connected words. *Institute of Acoustics*, 14:95–100, 1992.

Pasumarthi, N. et Malleswari, L. : An empirical study and comparative analysis of content based image retrieval (CBIR) techniques with various similarity measures. Dans *Proceedings of the 3rd IEEE International Conference on Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences (EEECOS)*, pages 373–379. IEEE, 2016.

Patino, J., Delgado, H. et Evans, N. : Speaker change detection using binary key modelling with contextual information. Dans *Proceedings of the 5th LIUM International Conference on Statistical Language and Speech Processing (SLSP)*, pages 250–261. LIUM, 2017.

Patino, J., Delgado, H. et Evans, N. : The EURECOM submission to the first dihard challenge. Dans *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2018a.

Patino, J., Delgado, H., Evans, N. et Anguera, X. : EURECOM submission to the Albayzin 2016 speaker diarization evaluation. Dans *Proceedings of the 3rd International Conference on Speech and Language Technologies for Iberian Languages (IberSPEECH)*. ISCA & INESC-ID & RTTH, 2016.

Patino, J., Delgado, H., Yin, R., Bredin, H., Barras, C. et Evans, N. : ODESSA at Albayzin speaker diarization challenge 2018. Dans *Proceedings of the 4th International Conference on Speech and Language Technologies for Iberian Languages (IberSPEECH)*. ISCA & INESC-ID & RTTH, 2018b.

Patry, A. et Langlais, P. : Prediction of words in statistical machine translation using a multilayer perceptron. Dans *Proceedings of the 12th EAMT Machine Translation Summit (MT Summit)*, pages 101–111. EAMT, 2009.

Peddinti, V., Povey, D. et Khudanpur, S. : A time delay neural network architecture for efficient modeling of long temporal contexts. Dans *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2015.

Prince, S. J. et Elder, J. H. : Probabilistic linear discriminant analysis for inferences about identity. Dans *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

Puigcerver, J. : Are multidimensional recurrent layers really necessary for handwritten text recognition ? Dans *Proceedings of the 14th IEEE International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 67–72. IEEE, 2017.

Rani, P. et Jindal, S. : Features extraction using local binary patterns and steerable pyramids for efficient image retrieval. *International Journal of Computer Applications*, 975:8887, 2018.

Reynolds, D. A. : *A Gaussian mixture modeling approach to text-independent speaker identification*. Thèse de doctorat, Georgia Institute of Technology, 1993.

Reynolds, D. A. : Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108, 1995.

Reynolds, D. A., Quatieri, T. F. et Dunn, R. B. : Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

Reynolds, D. A., Singer, E., Carlson, B. A., O'Leary, G. C., McLaughlin, J. J. et Zissman, M. A. : Blind clustering of speech utterances based on speaker and language characteristics. Dans *Proceedings of the 5th ISCA International Conference on Spoken Language Processing (ICSLP)*. ISCA, 1998.

Richardson, F., Reynolds, D. et Dehak, N. : Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675, 2015.

Rodríguez, L., Casacuberta, F. et Vidal, E. : Computer assisted transcription of speech. Dans *Proceedings of the 3rd IAPR International Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pages 241–248. IAPR, 2007.

Rose, R. C. et Reynolds, D. A. : Text independent speaker identification using automatic acoustic segmentation. Dans *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 90, pages 293–296. IEEE, 1990.

Ross, D. A., Lim, J., Lin, R.-S. et Yang, M.-H. : Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141, 2008.

Rouvier, M. et Meignier, S. : A global optimization framework for speaker diarization. Dans *Proceedings of the 2012 ISCA Workshop on Speaker and Language Recognition (Odyssey)*. ISCA, 2012.

Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S. et Liberman, M. : First DIHARD challenge evaluation plan, 2018.

SchlumbergerSema, S. A. : IT de Informática, R.W.T.H. Aachen, University of Montreal, Celer Soluciones, Société Gamma and Xerox Research Centre Europe : TT2. *TransType2-Computer Assisted Translation*, 2001.

Schwarz, G. et others : Estimating the dimension of a model. *The Annals of Statistics*, 6 (2):461–464, 1978.

Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., Watanabe, S. et others : Diarization is hard : Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. Dans *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2808–2812. ISCA, 2018.

Shete, D. S., Chavan, M. S. et Kolhapur, K. : Content based image retrieval. *International Journal of Emerging Technology and Advanced Engineering*, 2(9):85–90, 2012.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D. et Glass, J. : Exploiting intra-conversation variability for speaker diarization. Dans *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2011.

Shum, S. H., Campbell, W. M. et Reynolds, D. A. : Large-scale community detection on speaker content graphs. Dans *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7716–7720. IEEE, 2013.

Shum, S. H., Dehak, N. et Glass, J. R. : Limited labels for unlimited data : Active learning for speaker recognition. Dans *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 383–387. ISCA, 2014.

Sibson, R. : SLINK : An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

Siegler, M. A., Jain, U., Raj, B. et Stern, R. M. : Automatic segmentation, classification and clustering of broadcast news audio. Dans *Proceedings of the 1997 DARPA International Workshop on Speech Recognition*. DARPA, 1997.

Simard, M. et Isabelle, P. : Phrase-based machine translation in a computer-assisted translation environment. Dans *Proceedings of the 12th EAMT Machine Translation Summit (MT Summit)*, pages 120–127. EAMT, 2009.

Siu, M. H., Yu, G. et Gish, H. : An unsupervised, sequential learning algorithm for the segmentation of speech waveforms with multiple speakers. Dans *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 189–192. IEEE, 1992.

Slocum, J. : A survey of machine translation : Its history, current status and future prospects. *Computational linguistics*, 11(1):1–17, 1985.

Snyder, D., Garcia-Romero, D., Povey, D. et Khudanpur, S. : Deep neural network embeddings for text-independent speaker verification. Dans *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 999–1003. ISCA, 2017.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D. et Khudanpur, S. : X-vectors : Robust DNN embeddings for speaker recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y. et Khudanpur, S. : Deep neural network-based speaker embeddings for end-to-end speaker verification. Dans *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*, pages 165–170. IEEE, 2016.

Soldi, G., Beaugeant, C. et Evans, N. : Adaptive and online speaker diarization for meeting data. Dans *Proceedings of the 23rd IEEE European Conference on Signal Processing (EUSIPCO)*, pages 2112–2116. IEEE, 2015.

Solomonoff, A., Mielke, A., Schmidt, M. et Gish, H. : Clustering speakers by their voices. Dans *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 757–760. IEEE, 1998.

Stadelmann, T. et Freisleben, B. : Fast and robust speaker clustering using the earth mover's distance and mixmax models. Dans *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2006.

Stevens, S., Volkmann, J. et Newman, E. : The mel scale equates the magnitude of perceived differences in pitch at different frequencies. *Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

Tomás, J. et Casacuberta, F. : Statistical phrase-based models for interactive computer-assisted translation. Dans *Proceedings of the 2006 ACL International Conference on Computational Linguistics (COLING)*, pages 835–841. ACL, 2006.

Toselli, A., Romero, V., Rodríguez, L. et Vidal, E. : Computer assisted transcription of handwritten text images. Dans *Proceedings of the 9th IEEE International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 944–948. IEEE, 2007.

Toselli, A. H., Romero, V., Pastor, M. et Vidal, E. : Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

Toselli, A. H., Vidal, E. et Casacuberta, F. : Computer assisted transcription of speech signals. Dans *Multimodal Interactive Pattern Recognition and Applications*, pages 99–117. Springer, 2011a.

Toselli, A. H., Vidal, E. et Casacuberta, F. : Computer assisted transcription of text images. Dans *Multimodal Interactive Pattern Recognition and Applications*, pages 61–98. Springer, 2011b.

Toselli, A. H., Vidal, E. et Casacuberta, F. : Interactive image retrieval. Dans *Multimodal Interactive Pattern Recognition and Applications*, pages 209–226. Springer, 2011c.

Toselli, A. H., Vidal, E. et Casacuberta, F. : Interactive machine translation. Dans *Multimodal Interactive Pattern Recognition and Applications*, pages 135–152. Springer, 2011d.

Toselli, A. H., Vidal, E. et Casacuberta, F. : *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011e.

Trost, H., Matiasek, J. et Baroni, M. : The language component of the fasty text prediction system. *Applied Artificial Intelligence*, 19(8):743–781, 2005.

Vandecatseye, A., Martens, J. P., Neto, J. P., Meinedo, H., Garcia-Mateo, C., Dieguez-Tirado, J., Mihelic, F., Zibert, J., Nouza, J., David, P. et others : The COST278 Pan-European broadcast news database. Dans *Proceedings of the 4th ELRA International Conference on Language Resources and Evaluation (LREC)*. ELRA, 2004.

Verdejo, J. E. D., Peinado, A. M., Rubio, A. J., Segarra, E., Prieto, N. et Nolla, F. C. : Albayzin : A task-oriented Spanish speech corpus. Dans *Proceedings of the 1st ELRA International Conference on Language Resources and Evaluation (LREC)*, pages 497–502. ELRA, 1998.

Viterbi, A. : Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

Wan, L., Wang, Q., Papir, A. et Moreno, I. L. : Generalized end-to-end loss for speaker verification. Dans *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

Wandmacher, T. et Antoine, J. Y. : Modèle adaptatif pour la prédiction de mots. *Traitement Automatique des Langues*, 48(2):71–95, 2007.

Wang, Q., Downey, C., Wan, L., Mansfield, P. A. et Moreno, I. L. : Speaker diarization with LSTM. Dans *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243. IEEE, 2018.

Ward Jr, J. H. : Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Whitelock, P. J., Wood, M. M., Chandler, B. J., Holden, N. et Horsfall, H. J. : Strategies for interactive machine translation : The experience and implications of the UMIST Japanese project. Dans *Proceedings of the 11th ACL Conference on Computational Linguistics*, pages 329–334. ACL, 1986.

Wilks, D. S. : Cluster analysis. Dans *International Geophysics*, volume 100, pages 603–616. Elsevier, 2011.

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A. et Sloetjes, H. : ELAN : A professional framework for multimodality research. Dans *Proceedings of the 5th ELRA International Conference on Language Resources and Evaluation (LREC)*, pages 1556–1559. ELRA, 2006.

Wood, M. E. et Lewis, E. : Windmill-the use of a parsing algorithm to produce predictions for disabled persons. *Institute of Acoustics*, 18:315–322, 1996.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. et others : Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*, 2016.

Yu, D. et Deng, L. : *Automatic Speech Recognition*. Springer, 2016.

Zelenák, M., Schulz, H. et Hernando, J. : Speaker diarization of broadcast news in albayzin 2010 evaluation campaign. *EURASIP Journal on Audio, Speech and Music Processing*, page 19, 2012.

Zhang, Z., Deng, J., Marchi, E. et Schuller, B. : Active learning by label uncertainty for acoustic emotion recognition. Dans *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2013.

Annexe A

Compléments à l'état de l'art : segmentation et regroupement en locuteurs

Dans cette annexe, nous introduisons des compléments à l'état de l'art : segmentation et regroupement en locuteurs. Plus précisément, nous présentons des compléments à la partie de la paramétrisation de la parole et à la partie de la modélisation du locuteur.

A.1 Paramétrisation de la parole : BottleNeck Features et d-vecteur

Schématiquement inspirés du fonctionnement des neurones biologiques, les réseaux de neurones artificiels ont été étudiés depuis de nombreuses années dans divers domaines de recherche. Au fil des ans, ils ont réussi à surpasser les performances des systèmes à l'état de l'art dans nombre d'entre eux. Dans le domaine de la parole, les réseaux de neurones ont été étudiés depuis au moins une vingtaine d'années pour capturer les caractéristiques des locuteurs [Hermansky et Sharma, 1999; Hinton *et al.*, 2012]. Ce n'est qu'au vu des avancées de ces quatre dernières années que cette paramétrisation est devenue intéressante.

Deux vecteurs de paramètres extraits à partir des réseaux de neurones sont couramment utilisés dans le domaine de la parole. Ce sont les *BottleNeck Features* (BNF) [Himawan *et al.*, 2018; McLaren *et al.*, 2016b, 2015; Richardson *et al.*, 2015] et les *d-vecteurs* [Wan *et al.*, 2018; Wang *et al.*, 2018]. Ces deux vecteurs ne sont pas ou pas encore communément utilisés en SRL. L'extraction de ces deux vecteurs peut être divisée en deux étapes.

La première étape, qui est commune aux BNF et d-vecteurs, consiste tout d'abord à extraire les paramètres MFCC du signal audio. Ensuite, elle vise à appliquer une fenêtre glissante d'une taille fixe sur le signal afin de regrouper les paramètres extraits consécutifs. Chaque fenêtre est alors utilisée comme entrée d'un réseau de neurones qui vise à classifier chacune d'entre elles en fonction de leur contexte phonétique, voire plus rarement, en fonction du locuteur. L'intérêt d'utiliser une fenêtre, généralement de l'ordre du centième de seconde, est de donner au réseau un contexte temporel plus large.

La seconde étape vise à extraire les BNF ou d-vecteurs d'un réseau de neurones. Un réseau de neurones comprend généralement de 3 à 7 couches cachées, dont une de dimension réduite. Cette couche de dimension réduite est appelée *bottleneck*. Les BNF correspondent alors à la sortie de cette couche. Les d-vecteurs, quant à eux, correspondent à la couche de sortie du réseau.

Contrairement aux paramètres MFCC qui ne requièrent aucun apprentissage, il est nécessaire d'apprendre un réseau pour obtenir les BNF et d-vecteurs. En outre, les BNF et d-vecteurs comprennent déjà un contexte temporel large. Ainsi, il n'est pas utile d'enrichir ces données avec leurs dérivées.

A.2 Modélisation du locuteur

Dans cette partie, nous donnons des compléments au modèle i-vecteur présenté en 2.4.3. Plus précisément, nous présentons d'autres mesures que la PLDA pour comparer des modèles i-vecteurs. Enfin, nous présentons le modèle x-vecteur ainsi que le modèle binary key.

A.2.1 Modèle : i-vecteur

Pour comparer des i-vecteurs, il existe d'autres mesures classiques que la PLDA : la similarité cosinus et la distance de Mahalanobis.

Similarité Cosinus [Dehak *et al.*, 2011] est une mesure simple et classique permettant de déterminer si deux i-vecteurs proviennent d'un même locuteur ou non. Pour deux i-vecteurs donnés ω_i et ω_j, plus le score de la mesure de similarité cosinus se rapproche de la valeur 1, plus il est vraisemblable qu'ils représentent un seul et même locuteur. À l'inverse, plus le score se rapproche de -1, plus il est vraisemblable qu'ils sont issus de deux locuteurs distincts.

$$score_{cos}(\omega_i, \omega_j) = \frac{< \omega_i, \omega_j >}{||\omega_i||.||\omega_j||} \in [-1; 1], \tag{A.1}$$

où $||\omega_i|| = \sqrt{<\omega_i, \omega_i>}$ est la norme euclidienne d'un vecteur ω_i. Une normalisation WCCN [Dehak *et al.*, 2011] est généralement appliquée sur les i-vecteurs ω_i et ω_j.

Distance de Mahalanobis [Bousquet *et al.*, 2011] est une mesure de similarité visant à comparer deux i-vecteurs ω_i et ω_j :

$$score_{maha}(\omega_i, \omega_j) = (\omega_i - \omega_j)W^{-1}(\omega_i - \omega_j)', \tag{A.2}$$

où W représente une matrice de covariance intra-classe extraite à partir des i-vecteurs du corpus d'apprentissage. Une normalisation EFR [Bousquet *et al.*, 2012] est généralement appliquée s'applique sur les i-vecteurs ω_i et ω_j.

A.2.2 Modèle : x-vecteur

Les i-vecteurs sont des *embeddings*, c'est-à-dire des projections d'un ensemble de paramètres acoustiques dans un espace de dimension réduite. Depuis plusieurs années, les réseaux de neurones ont été étudiés afin de recréer ce type de projection. Ces recherches ont abouti à des résultats intéressants et ont notamment donné naissance à la récente modélisation x-vecteur [Snyder *et al.*, 2017, 2018]. Cette modélisation est d'ores et déjà couramment utilisée dans le domaine de la reconnaissance du locuteur, le domaine dans lequel elle a été initialement développée. Depuis son apparition, elle commence également à être introduite dans d'autres domaines de la parole tels que la tâche de SRL [Patino *et al.*, 2018b; Sell *et al.*, 2018].

Un x-vecteur est obtenu à partir d'un réseau de neurones profond (*Deep Neural Network - DNN*). Ce réseau est divisible en trois principaux blocs.

Le premier bloc correspond à un réseau de neurones TDNN (*Time-delay neural network*) [Peddinti *et al.*, 2015] qui prend en entrée les paramètres acoustiques (MFCC par exemple) d'un signal audio. Le second bloc est une couche dite de *pooling*. Pour un segment audio, elle vise à collecter les sorties du précédent bloc afin de produire des statistiques au niveau de l'énonciation. Ces statistiques prennent la forme d'un vecteur correspondant à la moyenne et l'écart type des sorties du TDNN associées au segment considéré. Le dernier bloc correspond à un réseau de neurones *feed-forward* simple [Mohamed *et al.*, 2012] où les vecteurs de chaque segment du signal audio issus du deuxième bloc sont utilisés comme entrée. La particularité de ce réseau est qu'il utilise pour sa dernière couche une fonction *softmax* donnant le locuteur le plus probable parmi les locuteurs présents dans les données d'apprentissage. Ce réseau est ainsi donc entraîné pour discriminer les locuteurs présents dans un corpus d'apprentissage donné. Dès qu'il a été entraîné, le réseau est utilisé pour

extraire les x-vecteurs. Les x-vecteurs correspondent alors à la sortie d'une des couches intermédiaires. La sortie de la couche *softmax* est, quant à elle, ignorée.

Habituellement, le premier et le dernier bloc comportent respectivement 3 et 5 couches.

A.2.2.1 Rapports de vraisemblance

Dans la littérature [Snyder *et al.*, 2017, 2018], le moyen le plus commun pour comparer des x-vecteurs est d'utiliser la méthode PLDA que nous avons présentée en 2.4.3.3 dans le cadre des i-vecteurs. Appliquées à la modélisation x-vecteur, les équations 2.14 et 2.15 se voient ainsi légèrement modifiées. L'équation 2.14 prend alors la forme suivante :

$$\chi_{(s,h)} = \mu + Cc_{s,h} + Ll_s + \epsilon_{s,h}, \tag{A.3}$$

où $\chi_{(s,h)}$ représente maintenant le x-vecteur pour une observation h d'un locuteur s et μ la moyenne globale des x-vecteurs, indépendante de s et de h. L'équation 2.15, quant à elle, se résume alors ainsi :

$$score_{PLDA}(\chi_i, \chi_j) = \log \frac{p(\chi_i, \chi_j|\mathcal{H}_0)}{p(\chi_i, \chi_j|\mathcal{H}_1)}, \tag{A.4}$$

où χ_i et χ_j représentent deux modèles x-vecteurs, $\mathcal{H}_0$ l'hypothèse que les deux modèles χ_i et χ_j sont issus d'un seul et même locuteur tandis que $\mathcal{H}_1$ l'hypothèse inverse.

A.2.3 Modèle : binary key

La modélisation binary key (*BK*), datant de 2010, a été initialement développée pour la tâche de reconnaissance du locuteur [Anguera et Bonastre, 2010; Hernandez-Sierra *et al.*, 2014]. Après son apparition, elle a été étendue à la tâche de SRL [Anguera et Bonastre, 2011; Delgado *et al.*, 2015; Patino *et al.*, 2017, 2018b]. Cette modélisation permet de représenter un segment de parole sous la forme d'un vecteur binaire discriminant de faible dimension. La modélisation Vecteur Cumulatif (*Cumulative Vector - CV*), quant à elle, est une pré-modélisation de la modélisation BK. Plus précisément, elle correspond à la sortie de l'avant-dernière étape du processus de calcul d'un vecteur BK. Un vecteur CV est identique à un vecteur BK à l'exception qu'il n'est pas binaire.

La représentation d'un segment de parole sous forme d'un vecteur BK ou CV est réalisée à partir d'un modèle *binary key background model* (*KBM*), qui a un rôle semblable à un modèle du monde (voir 2.4.2.2). Ce modèle KBM est une collection de modèles mono-gaussiens discriminants et complémentaires, sélectionnés à partir d'un ensemble de modèles mono-gaussiens appris sur une fenêtre glissante appliquée sur les données de test.

Grâce à ses composantes discriminantes et complémentaires, le modèle KBM permet d'avoir une couverture large de l'espace acoustique. Il est à noter que la taille d'un KBM, le nombre $\mathcal{K}$ de modèles mono-gaussiens, est fixée expérimentalement et est exprimée sous la forme d'un pourcentage représentant le nombre de modèles mono-gaussiens sélectionnés sur le nombre de modèles disponibles [Patino *et al.*, 2016].

Dès que nous avons un modèle KBM, nous pouvons alors transformer les vecteurs de paramètres acoustiques présents dans un segment de parole en un vecteur BK ou CV. Pour chaque vecteur de paramètres présent dans le segment, la vraisemblance avec chaque composante $\mathcal{K}$ du KBM est calculée. Pour un vecteur donné, les $\mathcal{K}_{best}$ meilleurs modèles mono-gaussiens du KBM ($\mathcal{K}_{best}$ fixé expérimentalement), c'est-à-dire les $\mathcal{K}_{best}$ plus vraisemblables, sont ensuite sélectionnés et utilisés pour créer une version binaire du vecteur. Dans un vecteur binaire, les $\mathcal{K}_{best}$ meilleurs modèles sont associés à une valeur de 1 tandis que le reste à une valeur de 0. Une fois qu'on a obtenu une version binaire de tous les vecteurs x, nous nous retrouvons avec une matrice binaire de dimension $x \times \mathcal{K}$ dont chaque colonne possède le même nombre de 1 et de 0. Chaque ligne de cette matrice est alors sommée pour donner un vecteur, le vecteur CV. Le vecteur CV illustre le nombre de fois que chaque modèle mono-gaussien dans le KBM a été choisi comme meilleur modèle pour tous les vecteurs de paramètres acoustiques présent dans le segment de parole. Le vecteur BK, quant à lui, est obtenu à partir des γ positions avec les plus grandes valeurs dans le vecteur CV (γ fixé expérimentalement). Ces γ positions dans le vecteur BK sont ainsi associées à la valeur 1 tandis que le reste à la valeur 0.

Pour la tâche de SRL, les articles Patino *et al.* [2018a] et Delgado *et al.* [2015] ont prouvé que les vecteurs CV fournissent de meilleurs résultats que les vecteurs BK. L'article Delgado *et al.* [2015] a également montré que les modélisations BK et CV fournissent des résultats légèrement moins bons que le modèle x-vecteur et le modèle i-vecteur. Cependant, les auteurs mettent en évidence que les modélisations BK et CV possèdent un avantage non négligeable. En effet, elles ne nécessitent pas d'apprentissage et peuvent ainsi être facilement appliquées dans un cadre où les données d'apprentissages sont limitées [Patino *et al.*, 2018b]. Dans ces mêmes conditions, l'obtention de x-vecteurs ou de i-vecteurs robustes est plus difficile.

A.2.3.1 Rapports de vraisemblance

Puisque les vecteurs CV fournissent de meilleurs résultats que les vecteurs BK pour la tâche de SRL, nous nous intéressons uniquement à eux dans cette partie. La mesure la plus utilisée pour comparer des vecteurs CV est la similarité cosinus [Delgado *et al.*, 2015; Patino *et al.*, 2018a, 2016]. Cette mesure est quasi identique à celle utilisée dans le cadre

des i-vecteurs (voir 2.4.3.3). La seule différence est le score de la mesure qui varie entre $[0; 1]$. Ce changement de valeurs du score est dû à la nature des vecteurs CV qui sont des vecteurs d'entiers positifs. Pour deux vecteurs CV $\mathcal{V}_i$ et $\mathcal{V}_j$, la similarité cosinus s'écrit comme suit :

$$score_{cos}(\mathcal{V}_i, \mathcal{V}_j) = \frac{\mathcal{V}_i, \mathcal{V}_j}{||\mathcal{V}_i||.||\mathcal{V}_j||} \in [0; 1]. \tag{A.5}$$

Annexe B

Mesure du coût des actions indépendantes de tout logiciel en unité élémentaire

Dans cette annexe, nous présentons et nous évaluons en unité élémentaire le jeu d'actions indépendantes de toutes applications que nous avions défini au début de nos travaux de recherche.

B.1 Actions humaines indépendantes de tout logiciel

Comme jeu d'actions indépendantes de toutes applications, nous avions défini les actions suivantes :

- "*Créer un label locuteur*" ;
- "*Changer le label locuteur*" ;
- "*Créer une frontière*" ;
- "*Supprimer une frontière*" ;
- "*Créer un segment*" ;
- "*Supprimer un segment*".

Les quatre premières actions fonctionnent comme décrit dans la partie 4.1.3. Les actions "*Créer un segment*" et "*Supprimer un segment*", quant à elles, consistent respectivement à ajouter un segment et à enlever un segment dans une SRL. Il est à noter que l'action "*Créer un segment*" requiert de définir la position de début et de fin du segment. Elle nécessite en outre à associer un label au segment créé.

Au début de nos travaux de recherche, on pensions que cet ensemble d'actions était obligatoirement présent dans n'importe quel logiciel de correction de SRL.

B.2 Évaluation des actions indépendantes avec le logiciel *Transcriber*

Pour évaluer les actions indépendantes que nous avons définies en unité élémentaire, nous nous sommes appuyés sur notre expérience de l'utilisation du logiciel *Transcriber*, introduit en 4.1.3. Afin d'évaluer plus précisément les actions, les interactions composant les actions ont été mesurées :

- "*Créer un label locuteur*" :
  - un clic sur le segment voulu dans le flux ;
  - un clic sur le label du segment ;
  - un clic sur le bouton "*Create Speaker*" ;
  - édition du nom du label ;
  - un clic sur "OK".
- "*Changer le label locuteur*" :
  - un clic sur le segment voulu dans le flux ;
  - un clic sur le label du segment ;
  - choix du label correct dans une liste ;
  - un clic sur "OK".
- "*Créer une frontière*" :
  - un sélection dans la position du texte ;
  - un sélection dans la position de l'audio ;
  - appui sur la touche "*Entrée*" ;
  - appui sur la combinaison "*Ctrl+t*" ;
  - application des actions pour "*Créer un label locuteur*" ou pour "*Changer le label locuteur*".
- "*Supprimer une frontière*" :
  - un clic sur le segment voulu dans le flux ;
  - appui sur la combinaison "*Ctrl+Delete*".

Comme on peut le constater, les actions humaines "*Créer un segment*" et "*Supprimer un segment*" ne sont pas détaillées. En effet, ces actions sont inexistantes dans le logiciel *Transcriber*.

En s'appuyant sur ce descriptif et en considérant que "*l'appui sur une touche*" et "*le clic sur la souris*" sont des unités élémentaires, une estimation des coûts des actions a été établie.

L'action "*Créer un label locuteur*" demande 4 actions élémentaires avec une édition du nom du label. Cette édition correspond aux nombres de lettres frappées. Nous en avons déduit qu'elle coûte environ 8 fois le coût d'une unité élémentaire. Cette estimation s'appuie sur la longueur moyenne des noms de personne en France.

L'action "*Changer le label locuteur*" a besoin de 3 actions élémentaires et d'une sélection du bon label dans une liste déroulante. Le choix dans une liste étant moins fastidieux que l'édition d'un nouveau label de locuteur, nous estimons qu'il coûte moins cher que cette dernière en unité élémentaire. Nous jugeons également que la sélection dans une liste est plus coûteuse que quelques actions élémentaires. Par conséquent, nous lui attribuons 3 fois le coût d'une unité élémentaire.

L'action "*Supprimer une frontière*" est seulement composée de 2 interactions élémentaires. Nous lui attribuons alors la valeur de 2 unités élémentaires.

Pour l'action "*Créer une frontière*", la sélection où aura lieu la séparation au niveau du texte et au niveau de l'audio est une tâche contraignante. Elle demande beaucoup d'implication. Nous l'évaluons à deux fois le coût de l'action "*Créer un label locuteur*", soit 24 unités élémentaires. Cette action nécessite également l'utilisation de l'action "*Changer le label locuteur*" dans le meilleur des cas ou de l'action "*Créer un label locuteur*" dans le pire des cas. Dans le dernier cas, on arrive à un coût de 36 unités élémentaires.

Annexe C

Détails de logiciels d'annotation

Dans cette annexe, nous présentons divers logiciels en lien avec la thèse, que ce soit des logiciels que nous avons utilisés (*Transcriber* et *ELAN*) ou des logiciels développés et utilisés par l'INA (*Hyperbase*, *MediaCorpus* et *MediaScope*).

C.1 Transcriber

FIGURE C.1 Illustration du logiciel *Transcriber*

Transcriber est un outil sous licence publique générale GNU pour segmenter, étiqueter et transcrire la parole de ressources audio [Barras *et al.*, 2001]. Développé en Tcl/Tk et C, il a été initialement pensé pour annoter des enregistrements d'émissions d'information et

pour créer des corpus dédiés au développement de systèmes de transcription automatiques. La figure C.1 présente une illustration du logiciel.

Le code source de *Transcriber* est disponible à partir d'un portail web à l'adresse *http ://trans.sourceforge.net/*.

Depuis le 11 juillet 2018, le logiciel *TranscriberAG* a succédé à *Transcriber*. *TranscriberAG* est disponible à l'adresse *http ://transag.sourceforge.net/*.

C.2 ELAN

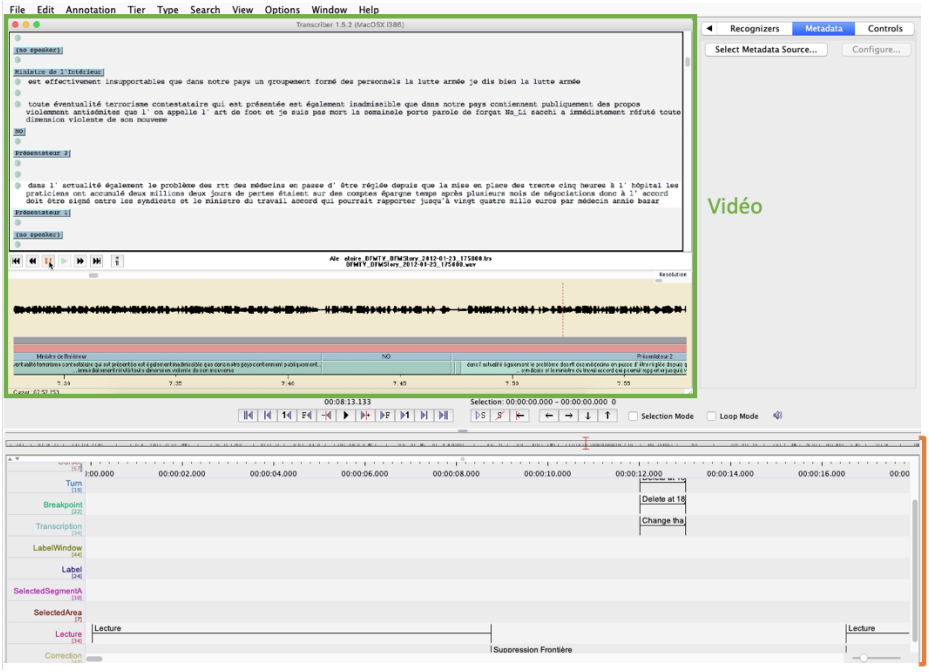


FIGURE C.2 Illustration du logiciel *ELAN* avec une annotation sur une vidéo d'une session de *Transcriber*

ELAN pour *EUDICO Linguistic Anotator* est un outil Java sous licence publique générale GNU 3 dédié à l'annotation complexe de ressources audio et vidéo [Wittenburg *et al.*, 2006]. Par rapport au logiciel *Transcriber*, il a l'avantage de permettre d'annoter une ressource sur plusieurs niveaux qui peuvent être hiérarchiquement interconnectés. La figure C.2 présente une illustration du logiciel.

Le code source de *ELAN* est disponible à l'adresse *https ://tla.mpi.nl/tools/tla-tools/elan*. À cette adresse, on peut entre autres retrouver une documentation complète permettant de faciliter l'installation et l'utilisation du logiciel.

C.3 Hyperbase

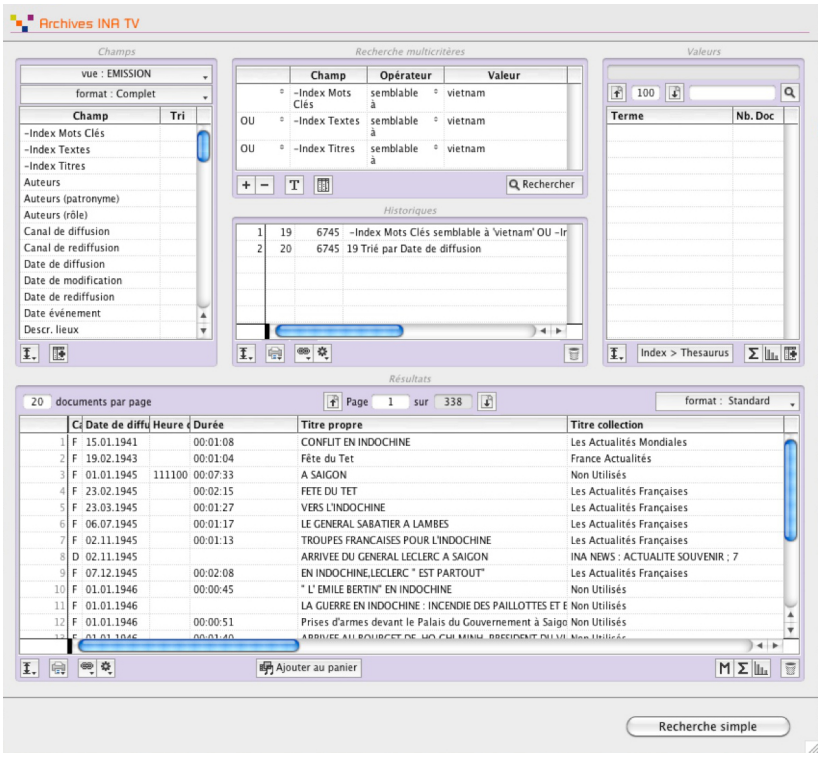


FIGURE C.3 Illustration du logiciel *Hyperbase*

Hyperbase est un outil développé par l'INA pour interroger ses bases de données documentaires réalisées par des documentalistes de l'institut. Cet outil permet la recherche multicritères, le tri, l'impression, mais également l'export de données documentaires. La figure C.3 présente une illustration de ce logiciel.

Hyperbase est disponible uniquement dans des centres de consultation INA.

C.4 MediaCorpus

MediaCorpus est un logiciel développé par l'INA pour créer une base de données documentaires personnalisée à partir d'un export d'*Hyperbase*, l'enrichir d'informations adaptées à une problématique, en déduire des statistiques et la visualiser sous forme de graphes. Plus précisément concernant l'enrichissement d'informations, *MediaCorpus* offre la possibilité d'ajouter des descripteurs ou mots-clefs personnalisés aux archives en plus de ceux déjà prédéfinis dans *Hyperbase*. Le temps d'une consultation, les mots-clefs personnalisés

peuvent être transférés au logiciel *Hyperbase* pour d'autres traitements ou recherches. Il est à noter que cet outil ne modifie pas les données d'*Hyperbase*.

MediaCorpus est disponible uniquement dans des centres de consultation INA.

C.5 MediaScope

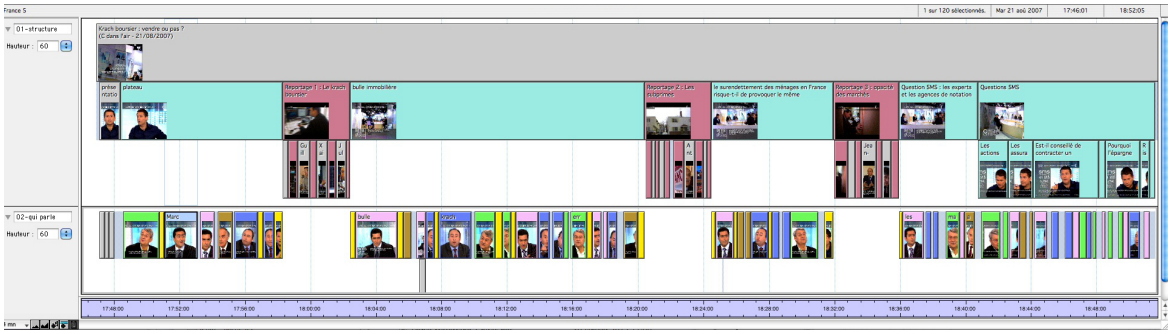


FIGURE C.4 Illustration du logiciel *MediaScope*

MediaScope est un outil d'aide à l'analyse de programmes audiovisuels développé par l'INA. Plus précisément, il permet de segmenter, d'annoter et de comparer des médias. Les documentalistes de l'INA utilisent ce logiciel pour créer les notices documentaires (voir 1.1). Cette application présente un inconvénient majeur pour le travail des documentalistes. En effet, elle n'offre pas la possibilité d'annoter la parole superposée comme peut le faire *ELAN* ou la dernière version de *Transcriber*. Pour annoter la parole superposée, les documentalistes utilisent alors une annotation particulière qui n'était pas prévue à cet effet. La figure C.4 présente une illustration du logiciel *MediaScope*. *MediaScope* est disponible à l'adresse *http ://www.inatheque.fr/consultation/mediascope.html*.

Annexe D

Outil S4D

Pendant la thèse, j'ai contribué au développement et au maintien de S4D [Broux *et al.*, 2018a], un outil Python en code source ouvert dédié à la tâche de la SRL. L'outil S4D fournit divers composants à l'état de l'art et la possibilité de développer aisément des systèmes de SRL complets. Il offre un large choix d'algorithmes de regroupement en locuteurs, de segmentation, d'évaluation et de visualisation. L'outil S4D a été pensé pour être facilement compréhensible, mis en place, modifiable et utilisable afin de permettre un passage rapide des technologies de SRL à l'industrie et pour faciliter le développement de nouvelles approches. L'outil S4D est une extension de l'outil pour la reconnaissance du locuteur : SIDEKIT [Larcher *et al.*, 2016].

S4D est disponible au moyen du dépôt PyPI avec la ligne de commande "*pip install s4d*" qui installera immédiatement toutes les dépendances Python requises. Le code source de S4D peut également être obtenu en clonant le dépôt GIT de S4D avec la ligne de commande "*git clone https ://git-lium.univ-lemans.fr/Meignier/s4d.git*".

Un portail web pour faciliter l'utilisation de S4D est disponible à l'adresse *http ://www-lium.univ-lemans.fr/sidekit/s4d/*. Ce portail inclut notamment des tutoriels, une documentation complète et des liens sur les outils utilisés.

Titre : Segmentation et regroupement en locuteurs dans des documents audiovisuels, en interaction avec des annotateurs humains

Mots clés : Segmentation et regroupement en locuteurs (SRL) ; Système assisté ; Interaction homme-machine (IHM) ; Annotation

Résumé : La tâche de segmentation et de regroupement en locuteur (SRL) consiste à déterminer le nombre de locuteurs ainsi que leurs interventions dans un document audio. Cette tâche intéresse de nombreuses entreprises qui souhaitent indexer leurs contenus audiovisuels. En particulier, l'institut national de l'audiovisuel (INA) désire appliquer cette tâche sur ses archives afin d'en améliorer l'accessibilité mais également l'annotation. Cependant, les usages de l'institut requièrent une qualité minimum qui n'est, la plupart du temps, pas encore atteinte par les systèmes automatiques de SRL à l'état de l'art. Pour atteindre les performances voulues, un humain peut corriger la sortie d'un système de SRL. Néanmoins, une intervention humaine est généralement chronophage et coûteuse. Afin de réduire ces coûts, une solution possible est d'utiliser un système assisté par l'humain : un humain donne des informations à un système afin qu'il améliore ses prédictions pour faire décroître son coût de correction. Le présent manuscrit s'articule autour de la SRL assistée par l'humain. Il propose une mesure afin d'évaluer le coût d'intervention humain pour corriger une SRL, un protocole pour évaluer les interactions d'un humain pour la SRL, un automate simulant les corrections humaines à faire pour une SRL et des systèmes de SRL assistés réduisant le coût d'intervention humain total. Plus précisément, les systèmes de SRL assistés présentés réévaluent soit uniquement le regroupement en locuteurs, soit la segmentation et le regroupement en locuteurs.

Title: Speaker diarization in audiovisual files in interaction with human annotators

Keywords: Diarization; Computer-assisted system; Human-computer interaction (HCI); Annotation

Abstract: The diarization task tries to determine the number of speakers as well as their interventions in an audio file. It is an interesting task for any enterprise willing to index its audiovisual contents. Especially, the French National Audiovisual Institute (INA) desires to apply this task on its archives so as to improve its accessibility and its annotation. However, the uses of the institute need a minimal quality which, most of the time, is not reached by the state-of-the-art automatic diarization systems yet. In order to reach the wanted effectiveness, a human can correct the output of a diarization system. Nevertheless, a human intervention is generally time-consuming and expensive. In order to reduce these costs, a possible solution is to use a computer-assisted a human gives some information to a system in order that it can improve its predictions so as to decrease its intervention cost. The present manuscript revolves around the computer-assisted diarization. It proposes a metric so as to assess the human intervention cost to correct a diarization, a framework to evaluate the human corrections of a speaker diarization, an automaton simulating the human corrections to do for a diarization and some computer-assisted diarization systems decreasing the total human intervention cost. More precisely, the proposed computer-assisted diarization systems reassess either only the speaker clustering or the segmentation and the speaker clustering.