Attelage de systèmes de transcription automatique de la parole

Fethi Bougares

Attelage de systèmes de transcription automatique de la parole

THÈSE

Présentée et soutenue publiquement le 23 Novembre 2012
pour l'obtention du

Doctorat de l'Université du Maine
(spécialité : Informatique)

préparée par

FETHI BOUGARES

Jury :

Rapporteurs :	Mme. Régine André-Obrecht	Professeur	IRIT, UPS Toulouse.
	M. Denis Jouvet	Directeur de recherche	LORIA, INRIA Nancy.
Examinateurs :	M. Gilles Boulianne	Conseiller senior en recherche et développement	CRIM, Montréal-Canada.
	M. Paul Deléglise	Professeur	LIUM, Université du Maine.
	M. Yannick Estève	Professeur	LIUM, Université du Maine.
	M. Georges Linarès	Professeur	LIA, Université d'Avignon.

Laboratoire d'Informatique de l'Université du Maine

Remerciement

Cette thèse a été réalisée dans le cadre d'un projet reliant plusieurs laboratoires de recherche : Le Laboratoire d'Informatique de l'Université du Maine (LIUM), le Laboratoire d'Informatique d'Avignon (LIA), et l'Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA). Ce contexte ma permis de travailler dans un cadre particulièrement agréable avec plusieurs personnes que je remercie tous.

J'adresse en premier lieu mes plus vifs sentiments de gratitude à mes encadrants de thèse Monsieur Paul DELÉGLISE, Professeur à l'université du Maine, Monsieur Yannick ESTÈVE, Professeur à l'université du Maine et Monsieur Georges LINARÈS, Professeur à l'université d'Avignon. Ce travail n'aurait pu aboutir sans leur aide. Je suis particulièrement très reconnaissant à Yannick ESTÈVE qui a toujours su trouver les mots pour me motiver et me soutenir tout au long de cette thèse.

Je tiens aussi à exprimer mes remerciement à Madame Régine ANDRÉ-OBRECHT, Professeur à l'université de Toulouse, de l'Institut de Recherche en Informatique de Toulouse (IRIT) ainsi que Monsieur Denis JOUVET Directeur de recherche au Laboratoire lOrrain de Recherche en Informatique et ses Applications (LORIA) à l'Université de Nancy qui ont accepté de juger ce travail et d'en être les rapporteurs. Je remercie également Monsieur Gilles BOULIANNE directeur du Centre de Recherche Informatique de Montréal pour avoir accepté de juger cette thèse et pour l'intérêt qu'il a porté à mon travail.

Ce travail n'aurait pu aboutir sans l'aide de nombreuses personnes. Que me pardonnent celles que j'oublie ici, mais j'adresse une pensée particulière a mes collègues de bureau Grégor Dupy, qui a participé activement à la relecture de ce manuscrit, et Michael Rouvier. Je remercie également les anciens du bureau Richard Dufour, Antoine Laurent, Thierry Bazillon et Vincent Jousse pour leur accueil chaleureux et leur amitié.

J'ai pu travailler dans un cadre particulièrement agréable, grâce à l'ensemble des membres de l'équipe *LST* (Language and Speech Technologie) du LIUM. Je pense particulièrement à Sylvain Meignier, Loïc Barrault, Kashif Shah, Nathalie Camelin, Bruno Jacob, Teva Merlin et Étienne Micoulaut, Daniel Luzatti, Carole lallier, Holger Schwenk, Frédéric Blain, Anthony Rousseau. Mes remerciements s'adressent aussi à Martine Turmeau, secrétaire du laboratoire, qui ma toujours aidé dans les tâches administratives. Je terminerai cette partie en remerciant tous mes amis et amies, ceux et celles que j'ai eu la chance de côtoyer et qui m'ont toujours encouragé et supporté moralement.

Ces remerciements ne seraient pas complets sans une pensée pour mes amis de longue date, Mourad Bougares, Ammar Mahdhaoui et Salim Jouili. Merci de m'avoir aidé et encouragé, et pour m'avoir changé les idées quand j'en avais besoin.

Mes dernières pensées iront vers ma famille, et surtout mes parents, qui m'auront permis de poursuivre mes études jusqu'à aujourd'hui.

Merci à tous et bonne lecture.
Fethi BOUGARES

Dédicace

À ma famille,
À mes amis
pour la patience et le dévouement dont ils ont fait preuve.

“”Le seul individu formé, c’est celui qui a appris comment apprendre“”
(Karl Rogers, 1976)

Table des matières

II Mes contributions

Résumé

Nous abordons, dans cette thèse, les méthodes de combinaison de systèmes de transcription de la parole *à Large Vocabulaire*. Notre étude se concentre sur l'attelage de systèmes de transcription hétérogènes dans l'objectif d'améliorer la qualité de la transcription à latence contrainte. Les systèmes statistiques sont affectés par les nombreuses variabilités qui caractérisent le signal de la parole. Un seul système n'est généralement pas capable de modéliser l'ensemble de ces variabilités. La combinaison de différents systèmes de transcription repose sur l'idée d'exploiter les points forts de chacun pour obtenir une transcription finale améliorée. Les méthodes de combinaison proposées dans la littérature sont majoritairement appliquées *a posteriori*, dans une architecture de transcription multi-passes. Cela nécessite un temps de latence considérable induit par le temps d'attente requis avant l'application de la combinaison.

Récemment, une méthode de combinaison intégrée a été proposée. Cette méthode est basée sur le paradigme de décodage guidé (DDA :*Driven Decoding Algorithm*) qui permet de combiner différents systèmes durant le décodage. La méthode consiste à intégrer des informations en provenance de plusieurs systèmes dits *auxiliaires* dans le processus de décodage d'un système dit *primaire*.

Notre contribution dans le cadre de cette thèse porte sur un double aspect : d'une part, nous proposons une étude sur la robustesse de la combinaison par décodage guidé. Nous proposons ensuite, une amélioration efficacement généralisable basée sur le décodage guidé par sac de n-grammes, appelé *BONG*. D'autre part, nous proposons un cadre permettant l'attelage de plusieurs systèmes mono-passe pour la construction collaborative, à latence réduite, de la sortie de l'hypothèse de reconnaissance finale. Nous présentons différents modèles théoriques de l'architecture d'attelage et nous exposons un exemple d'implémentation en utilisant une architecture *client/serveur* distribuée. Après la définition de l'architecture de collaboration, nous nous focalisons sur les méthodes de combinaison adaptées à la transcription automatique à latence réduite. Nous proposons une adaptation de la combinaison *BONG* permettant la collaboration, à latence réduite, de plusieurs systèmes mono-passe fonctionnant en parallèle. Nous présentons également, une adaptation de la combinaison *ROVER* applicable durant le processus de décodage *via* un processus d'alignement local suivi par un processus de vote basé sur la fréquence d'apparition des mots. Les deux méthodes de combinaison proposées permettent la réduction de la latence de la combinaison de plusieurs systèmes mono-passe avec un gain significatif du WER.

Abstract

This thesis presents work in the area of Large Vocabulary Continuous Speech Recognition (LVCSR) system combination. The thesis focuses on methods for harnessing heterogeneous systems in order to increase the efficiency of speech recognizer with reduced latency.

Automatic Speech Recognition (ASR) is affected by many variabilities present in the speech signal, therefore single ASR systems are usually unable to deal with all these variabilities. Considering these limitations, combination methods are proposed as alternative strategies to improve recognition accuracy using multiple recognizers developed at different research sites with different recognition strategies. System combination techniques are usually used within multi-passes ASR architecture. Outputs of two or more ASR systems are combined to estimate the most likely hypothesis among conflicting word pairs or differing hypotheses for the same part of utterance.

The contribution of this thesis is twofold. First, we study and analyze the integrated driven decoding combination method which consists in guiding the search algorithm of a primary ASR system by the one-best hypotheses of auxiliary systems. Thus we propose some improvements in order to make the driven decoding more efficient and generalizable. The proposed method is called *BONG* and consists in using Bag Of N-Gram auxiliary hypothesis for the driven decoding.

Second, we propose a new framework for low latency paralyzed single-pass speech recognizer harnessing. We study various theoretical harnessing models and we present an example of harnessing implementation based on *client/server* distributed architecture. Afterwards, we suggest different combination methods adapted to the presented harnessing architecture: first we extend the *BONG* combination method for low latency paralyzed single-pass speech recognizer systems collaboration. Then we propose, an adaptation of the *ROVER* combination method to be performed during the decoding process using a local vote procedure followed by voting based on word frequencies.

Chapitre 1
Introduction

1.1 Introduction

Le domaine de la reconnaissance de la parole demeure toujours un sujet de recherche d'actualité. Plusieurs efforts de recherche ont été réalisés au cours de ces dernières années pour proposer des solutions permettant de construire des systèmes robustes et performants. Bien que le système de transcription idéal soit toujours inexistant, la reconnaissance de la parole est aujourd'hui intégrée dans des applications concrètes, largement utilisées.

Un Système de Reconnaissance Automatique de la Parole (SRAP) est dit robuste s'il est capable de faire face à des événements imprévus. Ceci étant dit, les principales contraintes qui freinent le développement des systèmes robustes sont généralement liées à la variabilité présente dans le signal de la parole. La dégradation des performances est généralement engendrée par l'hétérogénéité des enregistrement audio et par leur vocabulaire diversifié. De plus, les conditions d'enregistrements sont disparates, avec des bruits additifs ou convolutifs et des locuteurs non connus à l'avance.

Aujourd'hui, la plupart des systèmes de transcription à large vocabulaire sont basés sur des méthodes statistiques avec des techniques d'apprentissage à partir des corpus oraux où la transcription correcte est connue à l'avance. Un SRAP statistique est constitué de plusieurs composants permettant la modélisation acoustique et linguistique de signal de la parole en vue de sa reconnaissance. De nombreuses techniques ont été développées pour améliorer chaque composante du système afin de prendre en compte ou d'atténuer les problèmes liés à la variabilité de la parole. Cependant, chaque technique présente certaines faiblesses et met l'accent sur quelques caractéristiques du signal de la parole. De ce fait, un seul système de reconnaissance n'est pas en mesure de prendre compte l'ensemble des variabilités possibles du signal de la parole.

Face à ces faiblesses de modélisation de l'ensemble de variabilités de la parole, de nombreuses stratégies de combinaisons de systèmes ont été proposées. La combinaison de systèmes de reconnaissance s'est montrée efficace pour améliorer la qualité de la transcription et augmenter la robustesse de systèmes. Cependant, les techniques de combinaison les plus développées sont les combinaisons *a posteriori* qui consistent, généralement, à confronter les sorties de plusieurs systèmes et à sélectionner, chaque fois, la meilleure proposition.

Ce travail de thèse s'inscrit dans le cadre du projet ASH[1] (Attelage de Systèmes Hétérogènes). Les travaux réalisés et présentés dans ce mémoire sont directement liés à la problématique de ce projet, détaillée dans la section suivante.

1. http://projet-ash.univ-lemans.fr/

1.2 Le projet ANR ASH

Financé par l'ANR [2] (Agence National de la Recherche), le projet ASH s'inscrit dans le domaine de la reconnaissance de la parole. L'objectif du projet est de proposer une méthode de combinaison de systèmes de reconnaissance automatique durant le processus de décodage de plusieurs systèmes de reconnaissance. Trois laboratoires académiques sont impliqués dans ASH :

- Le Laboratoire d'Informatique de l'Université du Maine (LIUM) ;
- Le Laboratoire d'Informatique d'Avignon (LIA) ;
- L'Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA).

La collaboration entre ces trois laboratoires de recherche permet l'échange scientifique et technologique entre trois partenaires majeurs à l'échelle nationale dans le domaine de la reconnaissance de la parole. Chaque partenaire possède un système de reconnaissance plus ou moins différent des autres, d'où l'intérêt de proposer une méthode de combinaison pour exploiter cette diversité et améliorer ainsi la qualité de la transcription finale.

Plus particulièrement, l'objectif du projet ASH consiste à offrir un cadre d'échange d'informations inter-systèmes, durant la tâche de transcription, qui nécessiterait un faible temps de latence. L'échange d'informations durant la recherche de la meilleure hypothèse de transcription nécessite une réflexion préalable sur plusieurs points :

1. la pertinence des informations échangées ;
2. le type d'informations à échanger entre systèmes ;
3. le format d'échange d'informations ;
4. le moment d'échange d'informations ;
5. la méthode d'intégration des informations échangées.

2. http://www.agence-nationale-recherche.fr

1.3 Structure du document

Ce document est organisé en deux grandes parties. Nous présentons dans un premiers temps l'état de l'art des systèmes de reconnaissance de la parole et des méthodes de combinaison de systèmes. Nous développons ensuite le travail réalisé durant cette thèse, à savoir l'analyse et l'amélioration de combinaison de systèmes par décodage guidé ainsi que la proposition et l'expérimentation d'une architecture de combinaison à latence réduite.

Nous présentons dans le premier chapitre les principales composantes d'un système de reconnaissance statistique de la parole. Nous détaillons également le système de transcription du LIUM, sur lequel s'appuient les différentes expériences menées durant ce travail de thèse.

Le deuxième chapitre détaille les différents travaux menés sur la combinaison des systèmes de reconnaissance. Nous expliquons tout d'abord l'objectif et les intérêts des méthodes de combinaison de systèmes et nous détaillons ensuite les méthodes de combinaison développées dans la littérature. Nous répartissons les méthodes de combinaison sur trois classes : nous choisissons l'endroit d'application de la méthode de combinaison comme critère de classification et nous présentons les méthodes de combinaison avant, après et durant le processus de décodage.

La première partie du travail est présentée dans le chapitre 4. Elle consiste à étudier et à analyser la combinaison de systèmes par décodage guidé en s'appuyant sur la formalisation proposée dans [Lecouteux 2007]. Pour cela, la combinaison a été d'abord intégrée dans le système de reconnaissance du LIUM. L'efficacité et la robustesse de cette méthode sont ensuite analysées et une amélioration de cette méthode permettant une généralisation efficace et performante est proposée.

L'étude de l'attelage de systèmes à latence réduite est présentée dans le chapitre 5. Afin de réduire la latence induite par les méthodes de combinaison classiques, nous proposons un cadre d'attelage de systèmes hétérogènes mono-passe, fonctionnant en parallèle. La combinaison est effectuée durant le décodage à travers un échange d'informations *via* un espace de partage d'informations. Nous présentons également, les méthodes de combinaison adaptées à l'attelage de systèmes à latence réduite.

Finalement, un résumé des points clés de la thèse est présenté, ainsi que quelques perspectives pour les travaux de recherche futurs.

Première partie

État de l'art

Chapitre 2

Méthodes statistiques pour la reconnaissance de la parole

Sommaire

Ce chapitre présente les Systèmes de Reconnaissance Automatique de la Parole (SRAP) continue basés sur les modèles de Markov cachés (MMC). Nous présentons dans un premier temps l'architecture d'un SRAP et nous détaillons par la suite les composants d'un tel système, tout en exposant les grandes approches utilisées. Une description détaillée du SRAP du LIUM terminera ce chapitre.

2.1 Architecture d'un SRAP

Étant donné un signal de la parole, l'objectif d'un système de reconnaissance est d'identifier la séquence de mots prononcée. La majorité des SRAP actuels se basent sur l'approche statistique formulée par [Jelinek 1977b] comme suit : à partir d'un ensemble d'observations acoustiques X, le système cherche la séquence de mots W^* maximisant l'équation suivante :

$$W^* = \arg\max_W P(W|X) \tag{2.1}$$

Après application du théorème de Bayes, cette équation devient :

$$W^* = \arg\max_W \frac{P(X|W)P(W)}{P(X)} \tag{2.2}$$

$P(X)$ est considérée constante et retirée de l'équation 2.2.

$$W^* = \arg\max_W P(X|W)P(W) \tag{2.3}$$

où $P(W)$ est estimée *via* un modèle de langage, $P(X|W)$ est estimée par un ou plusieurs modèles acoustiques et la maximisation $argmax$ est réalisée par le processus de décodage. La figure 2.1 présente les différents éléments d'un SRAP qui seront présentés dans les sections suivantes.

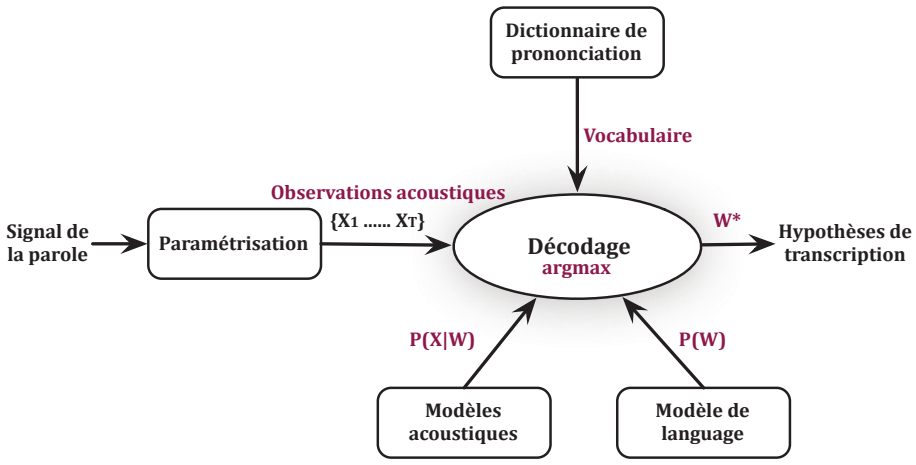


FIGURE 2.1 – Architecture d'un système de reconnaissance de la parole

Dans la formule 2.3, le modèle acoustique et le modèle de langage sont combinés à travers une simple multiplication. En pratique, les probabilités fournies par les deux modèles sont sous-estimées et ne représentent qu'une approximation des vraies probabilités. De plus, l'écart de la dynamique des valeurs de deux modèles est assez important, surtout lorsque la modélisation acoustique est basée sur des MMC à espace d'états continu [Huang 2001]. Par conséquent, la multiplication directe donnera au modèle de langage un poids assez faible et ne permettra pas d'obtenir des bons résultats de reconnaissance. La solution la plus couramment utilisée pour atténuer ce problème consiste à ajouter un poids noté *lw* appelé *linguistic weight* ou *fudge factor*, au modèle de langage.

Afin de gérer la longueur des phrases générées par le système et d'ajuster au mieux la tendance du système à insérer ou supprimer des mots, une pénalité linguistique *lp* (*linguistic penalty*) est aussi insérée dans la formule 2.3 qui devient :

$$W^* = \arg\max_W P(X|W) \times lp^{N(W)} \times P(W)^{lw} \tag{2.4}$$

lw et *lp* (*linguistic weight* et *linguistic penalty*) sont déterminés empiriquement et $N(W)$ représente le nombre de mots dans la séquence W.

2.2 Paramétrisation

Le signal de la parole est extrêmement redondant et variable. Afin de reconnaître correctement le contenu linguistique d'un tel signal, il est nécessaire de transformer ce signal et d'en extraire uniquement les paramètres utiles pour la reconnaissance. Le signal de parole est généralement représenté dans le domaine fréquentiel où le spectre est considéré stationnaire durant des intervalles de temps de 25 ms, avec un recouvrement de 10 ms en général.

Dans la littérature, plusieurs techniques de paramétrisation ont été développées dans le but d'extraire uniquement les paramètres qui seront dépendants du message linguistique. Les MFCC *"Mel-Frequency Cepstral Cœfficients "* (domaine cepstral) [Davis 1990], les PLP *"Perceptual Linear Prediction"* (domaine spectral) [Hermansky 1991] et les LPCC *Linear Prediction Cepstral Cœfficients* (domaine temporel) [Markel 1982] représentent les techniques de paramétrisation les plus utilisées. Le jeu de paramètres obtenu est couramment augmenté par leurs dérivées premières (Δ) et secondes ($\Delta\Delta$) qui permettent de mieux modéliser les caractéristiques dynamiques des paramètres acoustiques (vitesse et accélération).

2.3 Modélisation acoustique

Le modèle acoustique permet de calculer la vraisemblance du signal étant donné une séquence de mots. Dans les SRAP statistiques, la modélisation acoustique est généralement basée sur la théorie des Modèles de Markov Cachés (MMC). Après la construction de modèles acoustiques, des techniques d'adaptation peuvent être appliquées pour modéliser la variabilité interlocuteurs ainsi que les différences de conditions d'enregistrement.

2.3.1 Modèles de Markov Cachés

Les modèles de Markov cachés correspondent à des graphes d'états, dotés d'une fonction de transition entre états et d'une fonction de génération d'observations [Rabiner 1990]. Un modèle de Markov peut être vu comme un automate stochastique : chaque état est associé à une densité de probabilité qui modélise les formes rencontrées, et les transitions assurent les contraintes d'ordre temporel des formes pouvant être observées. La propriété importante des processus markoviens est que l'évolution de l'automate après l'instant t ne dépend que de la valeur de l'état où il se trouve à l'instant (t) et des commandes qui lui sont appliquées ensuite. En particulier, le futur ne dépend pas de la façon dont l'automate s'est retrouvé dans l'état en question.

2.3.1.1 Structure d'un MMC

En reconnaissance de la parole continue, l'unité acoustique la plus utilisée est le triphone, qui représente une réalisation acoustique particulière d'un phonème. Chaque triphone est généralement représenté par un MMC M gauche-droite à 3 états émetteurs (figure 2.2) défini par le quintuplet (S, Σ, T, G, π) :

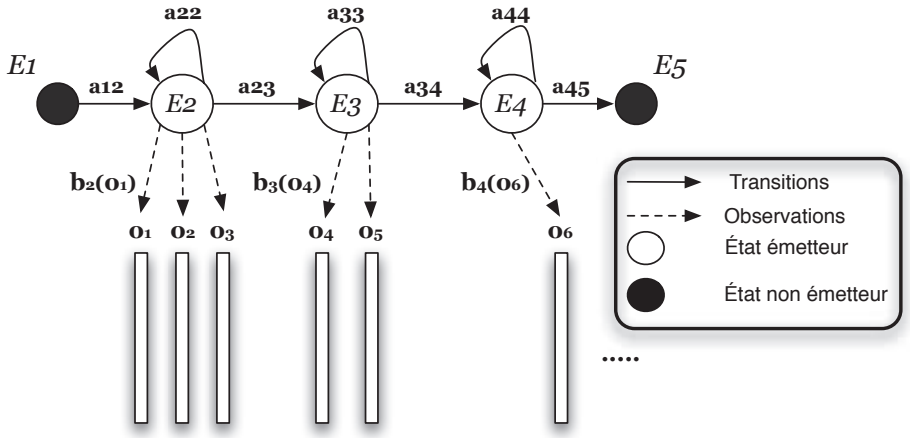


FIGURE 2.2 – MMC à 5 états dont 3 émetteurs.

- S est un ensemble de N états,
- Σ est un alphabet de M symboles,
- $T = S \times S \rightarrow [0,1]$ est la matrice des transitions a_{ij} entre états. La somme des probabilités de transitions entre un état i et tous les N autres états doit être égale à 1, *i.e* $\forall i, \sum_j a_{ij} = 1$,
- $G = S \times \Sigma \rightarrow [0,1]$ est la matrice d'observation indiquant les probabilités de génération associées aux états ; $b_i(o_t)$ est la probabilité de générer le symbole $o_t \in \Sigma$ à partir de l'état i. La somme des probabilités des émissions partant d'un état est égale à 1, *i.e* $\forall i, \sum_{o_t} b_i(o_t) = 1$,
- $\pi : S \rightarrow [0,1]$ est un vecteur de probabilités initiales. La somme de ces N probabilités doit être égale à 1, *i.e* $\forall i, \sum_i \pi_i = 1$.

La génération d'une séquence d'observations $o_1.....o_T$ à l'aide d'un MMC M consiste à partir d'un état S en suivant la distribution initiale π, à boucler sur le même état ou se déplacer vers l'état suivant ($i \rightarrow i$ ou $i \rightarrow j$) en suivant les probabilités de transition a_{ij} et à générer un symbole sur chaque état rencontré en utilisant la distribution de probabilités de la matrice d'observation associée à l'état j. La distribution discrète présentée précédemment doit être modifiée lorsque les observations viennent d'un espace continu. En effet, le type des observations modélisées définit la nature discrète ou continue d'un MMC. Dans le cadre de la reconnaissance de la parole, les observations sont continues et la densité de probabilité d'observation $b_i(o_t)$ est généralement estimée par une somme pondérée de K fonctions de densité gausienne $\mathcal{N}(\mu, \Sigma)$ (Gaussian Mixture Model - GMM), d'espérance μ et de matrice de covariance Σ. La probabilité d'observation est définie par :

$$b_i(o_t) = \sum_{k=1}^{K} c_{ik}\, \mathcal{N}(o_t, \mu_{ik}, \Sigma_{ik}), \qquad \sum_{k=1}^{K} c_{ik} = 1 \tag{2.5}$$

Chaque gaussienne ayant une densité de probabilité continue égale à :

$$\frac{1}{\sqrt{(2\pi)^D det(\Sigma)}} exp(-\frac{1}{2}(o_t - \mu)\Sigma^{-1}(o_t - \mu)) \tag{2.6}$$

avec o_t le vecteur d'observation de dimension D, μ le vecteur moyen de la gausienne et Σ la matrice de covariance.

La topologie des MMC est définie par la matrice de transition. Dans l'exemple de la figure 2.2, le modèle contient 3 états émetteurs et deux états non émetteurs avec des transitions gauche-droite et une boucle possible sur chaque état émetteur.

Les MMC soulèvent trois problèmes de base :

1. L'apprentissage de l'ensemble des paramètres d'un modèle : étant donné un jeu d'entraînement $X = \{O^k\}_k$ contenant des séquences d'observations, quel est le modèle $\lambda = (S, \Sigma, T, G, \pi)$ du MMC qui aurait vraisemblablement généré le jeu X ?

2. L'évaluation de capacité d'un modèle : étant donné un MMC $\lambda = (S, \Sigma, T, G, \pi)$, quelle est la probabilité $P(O|\lambda)$ d'avoir une séquence d'observations $O = \{o_1, o_2, ..., o_T\}$?

3. La reconnaissance : étant donné un MMC $\lambda = (S, \Sigma, T, G, \pi)$ et une séquence d'observations O, quelle est la séquence d'état $S = \{s_1, s_2, ..., s_T\}$ qui a le plus vraisemblablement généré O ?

2.3.2 Estimation des paramètres

L'estimation des paramètres d'un MMC est généralement effectuée à l'aide d'un gros volume de données annotées, considérées comme séquences représentatives des données que l'on souhaite modéliser. On peut distinguer dans le problème de l'apprentissage d'un MMC deux cas de figure distincts, suivant que la structure (nombre d'états du MMC et transitions autorisées) est connue ou non. Lorsque la structure est connue, le problème se réduit à un problème d'estimation des paramètres numériques de manière à expliquer au mieux les séquences d'apprentissage. Si la structure du modèle n'est pas connue à l'avance l'apprentissage devient encore plus difficile.

En reconnaissance de la parole, la structure est déterminée avant de commencer le processus d'entraînement. Le processus d'apprentissage se réduit donc à déterminer les paramètres définissant le Modèle de Markov Caché de chaque unité phonétique. Il faudra donc estimer pour chaque modèle :

- les probabilités initiales π_i
- les probabilités de transitions a_{ij}
- les probabilités d'émissions $b_i(o_t)$ définies par :
  - les moyennes μ_i,
  - les matrices de covariance Σ_i
  - les cœfficients du mélange des gaussiennes c_i

Il existe différentes méthodes d'entraînement. La méthode la plus utilisée est basée sur une estimation par maximum de vraisemblance (MLE : Maximum Likelihood Estimation) qui peut être employée seule ou couplée à des méthodes complémentaires, de type discriminant, tel le MMIE (Maximum Mutual Information Estimation) [Valtchev 1997].

L'estimation par MLE suppose que les paramètres sont fixes mais inconnus. L'objectif est alors de trouver un modèle M qui maximise l'ensemble de paramètres $\lambda =< a_{ij}, b_i(o_t), \pi >$:

$$M = \arg\max_{\lambda} P(O|\lambda) \tag{2.7}$$

Il n'existe pas de méthode directe pour résoudre ce problème de maximisation. La maximisation est réalisée avec l'algorithme *Baum Welch* [Baum 1966] qui permet d'ajuster itérativement les paramètres du modèle λ afin qu'il coïncide avec le phénomène qu'il doit modéliser.

2.3.3 Adaptation acoustique

Les données d'apprentissage sont généralement limitées en quantité et ne permettent pas de représenter toute la variabilité entre différents locuteurs et différentes conditions d'enregistrement. Pour remédier à ces difficultés, des techniques d'adaptation du modèle acoustique ont été mises en place. L'idée de base est de pouvoir créer de nouveaux modèles spécialisés à partir de modèles généraux et de données homogènes, en quantité limitée.

Dans la littérature, il existe plusieurs techniques d'adaptation, appliquées, soit en utilisant un corpus différent, soit à partir d'un sous-ensemble de données, extraites de corpus d'apprentissage. D'autres méthodes permettent l'adaptation après une première transcription dans une architecture de reconnaissance multi-passes. Dans la suite, nous présentons les techniques d'adaptation les plus utilisées dans les SRAP.

2.3.3.1 Adaptation par Maximum *A Posteriori* : MAP

L'adaptation MAP permet de réduire la divergence entre le modèle acoustique initial et un corpus de données spécifiques généralement plus proches des données de test que celles du corpus d'apprentissage. Proposée par [Gauvain 1994], cette adaptation consiste à converger itérativement vers des paramètres optimaux $\hat{\lambda}$ en utilisant l'approche de maximum de vraisemblance : étant donné l'ensemble de paramètres du modèle acoustique de base λ et le

corpus de données spécifiques X, la méthode adapte le modèle selon la formule suivante :

$$\hat{\lambda}_{MAP} = \arg\max_{\lambda} P(\lambda|X) \tag{2.8}$$

$$\hat{\lambda}_{MAP} = \arg\max_{\lambda} P(X|\lambda)P(\lambda) \tag{2.9}$$

L'inconvénient majeur de la méthode MAP classique est que seuls les paramètres observés dans les données d'adaptation sont ajustés. Si la taille de données n'est pas assez importante, seul un faible pourcentage des paramètres peut être adapté et l'amélioration de la précision du système est limitée. En contrepartie, si la quantité de données d'adaptation utilisées est suffisante, le système devient beaucoup plus précis après l'adaptation MAP.

Dans l'équation 2.9, si la distribution *a priori* $P(\lambda)$ est uniforme l'adaptation MAP est équivalente à un apprentissage par maximum de vraisemblance (équation 2.7).

2.3.3.2 Adaptation par transformation linéaire

Contrairement à l'adaptation MAP, où toutes les gaussiennes de tous les états sont adaptées indépendamment les unes des autres, l'adaptation par transformation linéaire est unique pour l'ensemble des modèles. Les différences entre les conditions d'entraînement et les conditions de test, supposées linéaires, sont modélisées par des matrices de transformation estimées en utilisant les données d'adaptation.

Il existe plusieurs variantes de l'adaptation par transformation linéaire. Ces variantes se différencient par la méthode d'estimation des matrices de transformation et par les composantes sur lesquelles la transformation est appliquée. Dans la suite, nous présentons les méthodes d'adaptation les plus utilisées.

- **Adaptation MLLR**

  L'adaptation MLLR (*Maximum Likelihood Linear Regression*) [Leggetter 1995] transforme les paramètres du modèle initial afin de construire un modèle plus approprié aux nouvelles conditions (nouveau locuteur, nouvel environnement...). Les moyennes des gaussiennes sont supposées contenir les différences acoustiques principales entre les conditions d'apprentissage et les conditions de tests. En effet, chaque

moyenne $\hat{\mu}$ du système adapté est obtenue par une transformation linéaire de la moyenne μ du modèle initial en utilisant une transformation affine.

$$\hat{\mu} = A\mu + b \tag{2.10}$$

Si les observations sont de dimension n, A est alors une matrice de dimension $n \times n$ et b un vecteur de dimension n.

L'équation peut être écrite sous la forme suivante :

$$\hat{\mu} = W_i \xi \tag{2.11}$$

avec W une matrice de taille $n \times (n+1)$ et ξ le vecteur de moyennes étendu.

$$\xi^T = [\ 1\ \mu_1\ \ ....\ \ \mu_n\ ] \tag{2.12}$$

La matrice W peut être estimée en utilisant l'algorithme E-M (*Expectation-Maximisation)* [Dempster 1977] dans un cadre de maximisation de vraisemblance de données d'adaptation.

L'adaptation MLLR a l'avantage d'être efficace même avec peu de données d'adaptation. Si la quantité de données d'adaptation est limitée, l'adaptation MLLR peut être réalisée avec une seule matrice W. En présence d'une grande quantité de données d'adaptation, une adaptation plus robuste est possible avec l'utilisation de plusieurs transformations. Les gaussiennes sont groupées selon un critère donné (espace acoustique, unité phonétique) et une transformation est appliquée par groupe de gaussiennes [Gales 1996a].

Comme nous l'avons précisé précédemment, l'adaptation MLLR suppose que les moyennes des gaussiennes encodent les différences, supposées linéaires, entre les conditions d'entraînement et celles de test. Une extension de cette méthode est proposée dans [Gales 1996b], où une transformation linéaire est appliquée aussi aux matrices de covariances Σ selon la formule suivante.

$$\hat{\Sigma} = H\Sigma H^T \tag{2.13}$$

avec H la matrice de transformation estimée et H^T sa transposée.

- **Adaptation CMLLR**

  Avec l'adaptation CMLLR (*Constrained Maximum Likelihood Linear Regression*) [Digalakis 1995], les moyennes et les variances du modèle initial sont transformées en utilisant la même matrice A_c comme suit :

$$\hat{\mu} = A_c\mu + b_c \tag{2.14}$$

$$\hat{\Sigma} = A_c\Sigma A_c^T \tag{2.15}$$

  L'adaptation CMLLR a l'avantage d'être directement applicable sur les vecteurs de paramètres acoustiques. Ce type d'adaptation est appelée fMLLR (*feature-space Maximum Likelihood Linear Regerssion*) [Yongxin 2002].

- **Apprentissage adaptatif SAT**

  L'apprentissage adaptatif SAT [Anastasakos 1996] vise à réduire l'impact des variations inter-locuteurs lors de l'estimation des modèles acoustiques. Pour ce faire, une transformation linéaire est estimée pour chaque locuteur du corpus d'apprentissage en maximisant la vraisemblance des données transformées étant donné le modèle multilocuteur.

  Un nouveau modèle est alors construit en utilisant les données d'apprentissage transformées. Ce modèle est utilisé lors du décodage pour faciliter l'adaptation non-supervisée. La transformation linéaire est obtenue *via* la technique d'adaptation CMLLR.

  L'apprentissage adaptatif (SAT) se déroule comme suit [Gales 1998] :

  1. Apprentissage d'un modèle initial indépendant du locuteur ;
  2. Estimation des matrices de transformation CMLLR sur les données d'apprentissage en utilisant le modèle indépendant du locuteur ;
  3. Apprentissage d'un nouveau modèle en utilisant les matrices de transformation estimées à l'étape précédente ;
  4. Tester le nouveau modèle ;
  5. Retour à l'étape 2 en utilisant le nouveau modèle jusqu'à stabilisation des résultats.

2.4 Modélisation linguistique

Un modèle de langage (ML) permet d'évaluer le degré de justesse d'une suite de mots dans une langue donnée. Il modélise les contraintes linguistiques de formalisation des énoncés corrects dans cette langue. Dans le cadre de la reconnaissance de la parole, le modèle de langage guide le décodage acoustique avec l'estimation de $P(W)$ dans l'équation 2.3.

Dans un système de RAP, les modèles de langage les plus utilisés sont les modèles probabilistes, que nous présentons ci-dessous.

2.4.1 Modèles de langage n-grammes

Le modèle de langage *n-grammes* permet d'estimer la probabilité *a priori* $P(W)$ d'une séquence de mots S. Formellement, la probabilité de la séquence de mots $S = w_1^n = w_1 ... \ w_i \ ... w_n$ est calculée par :

$$P(w_1^n) = P(w_1) \prod_{i=2}^{n} P(w_i | w_1 \ ... \ w_{i-1}) \tag{2.16}$$

avec $P(w_1)$ la probabilité d'observer le mot w_1 et $P(w_i|w_1 \ ... \ w_{i-1})$ celle de rencontrer le mot w_i après la séquence $w_1 \ ... \ w_{i-1}$.

Dans ce type de modèle, la probabilité d'un mot est calculée en fonction de son historique. Comme il est impossible de prendre en compte tous les historiques, car il n'existe généralement pas de corpus d'apprentissage suffisamment volumineux, le calcul est approché par un historique limité constitué des $n-1$ mots précédents. Le calcul de ces probabilités est basé sur le comptage de chaque séquence observée. L'ordre du modèle de langage définit la taille de l'historique h à prendre en considération dans le calcul.

Soit N la fonction qui, pour une séquence de mots, indique le nombre de fois où cette séquence a été observée dans le corpus d'apprentissage. Le calcul de la probabilité d'apparition du mot w en fonction de son historique h s'exprime alors sous la forme :

$$P(w|h) = \frac{N(h,w)}{N(h)} \tag{2.17}$$

2.4.2 Techniques de lissage

L'estimation des probabilités linguistiques dépend de la taille du corpus d'entraînement. Plus le corpus utilisé pour ces estimations est grand, plus les estimations de ces probabilités sont significatives. Cependant, quelle que soit la taille du corpus d'entraînement, il y a toujours des mots ou des séquences de mots absents du corpus pour lesquelles le modèle attribuera une probabilité nulle. De plus, même avec un historique réduit à quelques mots, beaucoup de *n- grammes* possibles sont absents des corpus d'entraînement.

Pour pallier ce problème un processus de lissage est mis en place. Ce processus consiste à attribuer une probabilité non nulle aux mots et séquences de mots non rencontrés. Le lissage peut aussi être vu comme une façon d'éviter le surentraînement d'un modèle sur un corpus, et de doter le modèle d'une plus grande capacité de généralisation.

Pour contourner le problème de séquences de mots inconnues, les techniques de lissage changent la distribution de la masse de probabilités. Au lieu de distribuer la totalité de la masse de probabilités sur les *n-grammes* vus dans le corpus d'entraînement, une partie de cette masse est retirée et distribuée aux *n-grammes* non vus dans le corpus. Dans la littérature, une série de méthodes de lissage ont été proposées. Dans [Chen 1999], les auteurs proposent une comparaison empirique de plusieurs techniques de lissage avec différents corpus, de tailles différentes avec différents paramètres. Ils ont ensuite proposé une modification de la technique de décompte proposée par [Kneser 1995]. Cette approche, couramment appelée lissage de Kneser-Ney modifié, offre les meilleures performances parmi les méthodes testées par [Chen 1999].

2.4.3 Modèles de langage n-classes

Dans l'objectif de contourner le problème de manque de données d'apprentissage, les mots qui présentent un comportement similaire (sémantique ou grammatical par exemple) peuvent être regroupés dans des classes. Un tel regroupement permet de changer l'échelle de modélisation et de créer des modèles n-classes [Brown 1992]. La méthode de classification peut se baser sur des informations syntaxiques (nom commun, verbe, préposition, etc.) mais aussi en utilisant des méthodes de classification automatique [Kneser 1993]. Avec ce principe de classification, le modèle prédit une classe C_i en fonction des $n-1$ classes qui la précèdent.

En posant C_i la classe courante, l'équation 2.17 du modèle trigramme se transforme, pour donner :

$$P(C_i|C_{i-2}, C_{i-1}) = \frac{N(C_{i-2}C_{i-1}C_i)}{N(C_{i-2}C_{i-1})} \tag{2.18}$$

L'avantage de cette méthode est le fait qu'un mot d'une classe donnée, ne se trouvant pas forcément dans le corpus d'apprentissage, hérite de la probabilité de tous les autres représentants de sa classe. Il est possible, en plus, d'ajouter des mots dans les classes sans avoir besoin de ré-estimer les probabilités du modèle.

Dans l'équation 2.18 le modèle considère que tous les mots d'une même classe sont équiprobables. Cependant, certains mots de même classe sont plus probables que d'autres. De plus, certains mots peuvent appartenir à différentes classes. Les modèles n-classes actuels intègrent la probabilité de chacun des mots au sein de leur classe ainsi que la probabilité d'appartenance d'un mot à une classe. La probabilité d'un mot au sein d'une séquence est alors obtenue par la formule 2.19 :

$$P(w_1^n) = \sum_{c \in C} \prod_{i=1}^{n} P(w_i|c_i) P(c_i|c_{i-N+1}^{i-1}) \tag{2.19}$$

Les classes peuvent être construites à l'aide de méthodes de classification automatiques, en maximisant par exemple l'information mutuelle entre les classes [Brown 1992] ou en minimisant la perplexité [Kneser 1993].

2.4.4 Adaptation linguistique

À ce jour, les modèles *n-grammes* sont les modèles de langage les plus utilisés vu leur efficacité et leur simplicité. Ces modèles sont simplement basés sur l'énumération de fréquences de *n-grammes* sur un corpus d'apprentissage. Cette simplicité représente, à la fois, l'avantage et l'inconvénient de ces modèles : la fréquence de *n-grammes* varie selon l'époque de production du corpus d'apprentissage utilisé, son style et les sujets traités. Ce problème de variabilité est d'autant plus important avec le langage naturel, qui évolue avec le temps : les règles grammaticales changent, des nouveaux styles émergent, des mots apparaissent et d'autres disparaissent. D'autres variabilités s'ajoutent dans le cadre du traitement de la parole, des variabilités liées à la différence de vocabulaire entre locuteurs, les disfluences, les répétitions, etc...

Idéalement, pour chaque type de texte, le modèle de langage utilisé devrait être appris sur des corpus d'apprentissage du domaine considéré. Il est difficile de se procurer des données propres à un domaine spécifique en quantité raisonnable et suffisante. Généralement, les données spécifiques ne constituent qu'une sous-partie d'un grand corpus utilisé pour apprendre un modèle de langage générique. Toutefois, les modèles de langage appris sur des données du domaine, malgré leurs tailles réduites, sont équivalents, voire même meilleurs que les modèles de langages appris sur une grande quantité de données hors domaine. Cela dit, pour avoir des modèles robustes, il est préférable d'utiliser l'ensemble des données disponibles (que ce soit dans le domaine ou hors domaine) en procédant par une interpolation linéaire d'un modèle générique appris sur le grand corpus (hors domaine) avec un modèle spécifique appris sur le petit corpus (dans le domaine). Cette combinaison est connue sous le nom d'adaptation de modèles de langages et le schéma général du processus d'adaptation est présenté dans la figure 2.3 [Estève 2002].

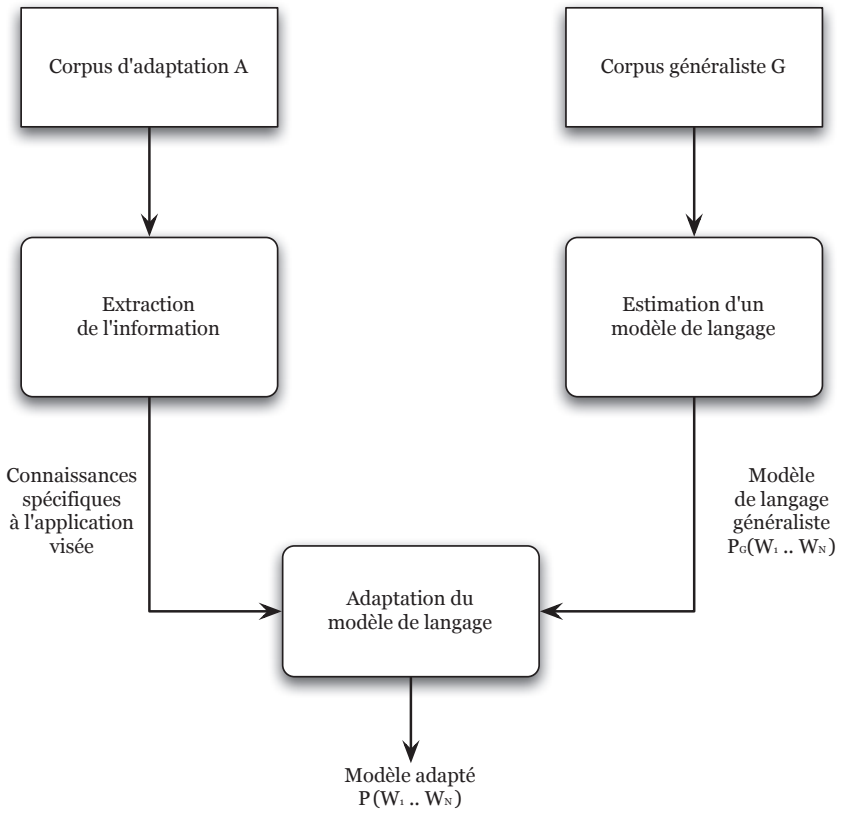


FIGURE 2.3 – Schéma général du processus d'adaptation d'un modèle de langage.

2.4.5 Évaluation du Modèle de Langage

Comme nous l'avons indiqué précédemment, le modèle de langage permet de guider le décodage pour améliorer la qualité de la transcription. De ce fait, la méthode d'évaluation la plus directe consiste à répéter le processus de transcription avec différents modèles de langage et à comparer la performance du système en termes de taux d'erreur mot (WER : Word Error Rate). L'utilisation de WER comme métrique d'évaluation pour les modèles de langage n'est pas pratique puisqu'elle implique, pour chaque évaluation, le lancement d'un processus de transcription complet.

La perplexité (PPL) est une méthode rapide pour évaluer les modèles de langage [Jelinek 1977a] couramment employée depuis plusieurs années pour juger la qualité d'un modèle de langage. Cette métrique d'évaluation permet de mesurer la capacité de prédiction d'un modèle de langage sur un corpus de test non vu au cours de l'apprentissage. La PPL calcule, pour chaque position d'un mot W, le nombre moyen de choix possibles. Naturellement, un modèle de langage est meilleur lorsque, pour une position, le nombre de choix entre mots est réduit (peu perplexe). La perplexité est définie par :

$$PPL = 2^{-\frac{1}{n}\sum_{t=1}^{n} log_2 P(w_t|h)} \tag{2.20}$$

où $P(w_t|h)$ représente la probabilité proposée par le modèle de langage pour le mot w_t sachant l'historique h.

Contrairement à l'évaluation par WER, la PPL est une mesure purement «*linguistique*». Pour être utilisable, elle doit être en corrélation avec le WER. Les travaux de *Iyer et al.* dans [Iyer 1996] montrent que les deux mesures sont corrélées lorsque le corpus d'évaluation est du même domaine que celui utilisé pour apprendre le modèle de langage.

2.5 Dictionnaire de prononciation

Le dictionnaire de prononciation définit l'ensemble des mots qu'un SRAP est capable de reconnaître ainsi que leurs prononciations. En effet les mots qui n'appartiennent pas à ce lexique ne figureront jamais dans la solution fournie par le système. Ce dictionnaire regroupe alors tous les mots nécessaires au décodage.

Le choix du nombre d'entrées lexicales définit le type de système de reconnaissance. Pour les systèmes dits *à Large Vocabulaire*, la taille du lexique est de l'ordre de plusieurs dizaines de milliers de mots. En plus de la définition de

l'ensemble des mots connus par le système, le dictionnaire de prononciation associe à chaque entrée sa décomposition en unités phonétiques, en tenant compte du fait qu'il peut exister plusieurs prononciations possibles pour un même mot.

Le jeu de phonèmes choisi dépend de la langue. Il en faut environ 45 pour l'anglais, 48 pour l'allemand, 26 pour l'espagnol et 35 pour le français. La création d'un dictionnaire de prononciation pour la langue française est détaillée plus précisément dans la section 2.10.1.2.

2.6 Décodeur

Le décodeur est le module qui effectue la reconnaissance. Ce module cherche dans un espace d'hypothèses très grand, le meilleur chemin qui donnera la séquence de mots la plus probable. Il existe de nombreuses stratégies de décodage, et l'emploi de l'une ou l'autre dépend d'un ensemble des contraintes liées à la taille du vocabulaire, aux ressources informatiques et au temps de traitement.

À partir des informations contenues dans le dictionnaire de prononciation et des modèles acoustiques et linguistiques, le décodeur cherche dans un premier temps à construire un graphe des mots ayant pu être prononcés dans le signal à analyser. L'espace de recherche construit est, par la suite examiné dans l'objectif de trouver l'hypothèse la plus probable. Dans cette section, nous présentons les principales méthodes de construction de l'espace de recherche, les techniques de contrôle de sa taille ainsi que les principaux algorithmes d'exploration.

2.6.1 Espace de recherche

Le choix de la structure de l'espace de recherche représente une caractéristique centrale de n'importe quel décodeur. Ce choix révèle les éléments communs ainsi que les différences réelles entre les différents schémas de décodage. La structure choisie doit représenter l'ensemble de l'espace de recherche et permettre une bonne représentation des liaisons phonétiques (inter- et intra-mots) et linguistiques (inter-mots).

Les méthodes de construction de l'espace de recherche sont généralement classées en deux catégories en fonction de l'expansion, statique ou dynamique, de l'espace de recherche [Aubert 2002, Ney 1999].

2.6.1.1 Espace de recherche statique

La séparation entre la phase de construction de l'espace de recherche et le processus de décodage, a été longtemps l'approche la plus naturelle. Différentes techniques d'optimisation de l'espace de recherche ont été proposées dans l'objectif d'exploiter la redondance pour permettre une représentation compacte et efficace de l'espace de recherche.

Dans [Povey 2011], l'espace de recherche est représenté par des transducteurs à états finis pondérés (WFST : weighted finite-state transducer). Les WFST offrent un cadre bien défini permettant d'appliquer différentes techniques d'optimisation et de compactage de l'espace de recherche.

Un tour d'horizon complet sur l'utilisation des WFSTs pour un système de reconnaissance de la parole avec un modèle de langage *tri-grammes* et un vocabulaire de 40,000 mots est présenté dans [Mohri 2002]. La représentation de l'espace de recherche par des WFSTs a été aussi comparée à l'utilisation d'un espace de recherche dynamique (section 2.6.1.2) et le processus de décodage d'un système à 40K mot de vocabulaire est 3 fois plus rapide pour une performance équivalente [Kanthak 2002] .

2.6.1.2 Espace de recherche dynamique

L'accroissement de la taille du vocabulaire, ainsi que l'utilisation des modèles linguistiques et acoustiques plus complexes a motivé l'intégration de l'expansion de l'espace de recherche dans le processus de décodage. Cela dit, dans les SRAP *à Large Vocabulaire*, l'espace de recherche est construit dynamiquement pendant le décodage. Les structures de recherche utilisées par les systèmes actuels sont généralement classées en trois groupes : les structures bouclées, les structures en arbre et les treillis [Lacouture 1995].

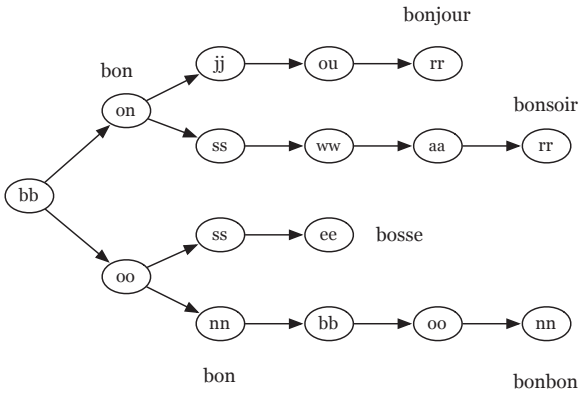


FIGURE 2.4 – Exemple d'arbre lexical

Pour des raisons d'efficacité algorithmique, le lexique phonétisé est communément compilé sous la forme d'un arbre lexical dont un exemple est illustré par la figure 2.4. Ce type d'arbre permet une représentation compacte de l'espace de recherche en partageant les états entre tous les mots ayant un préfixe commun. À l'exception de la racine, chacun des nœuds de cet arbre représente un phonème et les arcs correspondent aux transitions entre phonèmes répertoriés dans le lexique phonétisé.

2.6.2 Stratégies de décodage

Le SRAP explore l'espace de recherche afin de trouver le chemin qui maximisera une fonction de coût, constituée à la fois par les scores acoustiques et linguistiques. Nous présentons dans la suite les stratégies de décodage les plus utilisées.

2.6.2.1 Décodage synchrone

Le décodage synchrone est le plus répandu dans les SRAP *à Large Vocabulaire*. Il est généralement utilisé avec une structure de recherche dynamique et un algorithme de recherche Viterbi [Viterbi 1967] en faisceau (le *Viterbi beam search*).

L'algorithme de décodage en faisceau est un algorithme de recherche de style «en largeur d'abord» (*breadth-first*). Contrairement à l'algorithme *breadth-first* traditionnel, le *beam search* n'explore que les meilleurs chemins à chaque étape. L'algorithme construit *à la volée* et met à jour en permanence un graphe d'états (figure 2.5) représentant l'ensemble des hypothèses de transcription en recherchant l'hypothèse optimale.

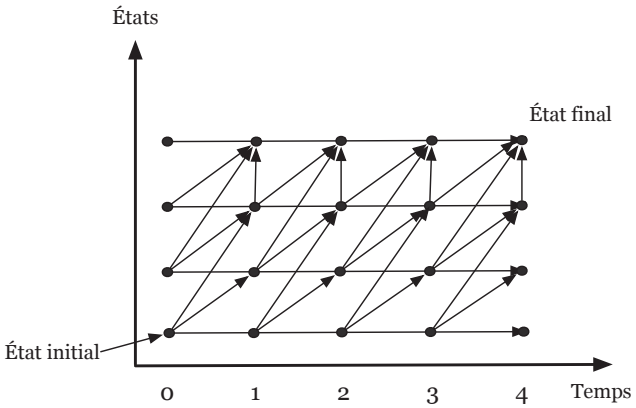


FIGURE 2.5 – Décodage Viterbi

La nature du décodeur *Viterbi* (time-synchronous) implique que l'espace de recherche soit parcouru dans l'ordre chronologique à partir de $t = 0$. La mise en oeuvre consiste à chercher le chemin qui a le meilleur score dans un tableau $T * N$ (T : nombre d'observations, N : nombre d'états des modèles) appelé *treillis d'hypothèses*.

Soit $b_i(o_t)$ la probabilité d'être à l'état S_i et de générer l'observation o_t, $\delta_t(i)$ la probabilité maximale d'une séquence qui se termine dans l'état S_i à un temps t et π_i la probabilité initiale d'être à l'état i (avec $1 \leq t \leq T$ et $1 \leq i \leq N$).

1. **Initialization :**
$$\delta_1(i) = \pi_i b_i(o_1) \text{ et } \Phi_1(i) = 0$$
2. **Récurrence :**
$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t) \qquad 2 \leq t \leq T$$
$$\Phi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \qquad 2 \leq t \leq T$$
3. **Fin :** $\hat{p} = \max_{1 \leq i \leq N} \delta_T(i)$
$$\hat{q}_T = \arg\max_{1 \leq i \leq N} \delta_T(i)$$

La séquence la plus probable est : $\hat{q}_t = \phi_{t+1}(\hat{q}_{t+1})$ pour $t = T-1, T-2, ..., 1$.

Chaque état traversé pointe vers le meilleur prédécesseur, le chemin le plus probable est récupéré en parcourant le treillis à l'envers. La taille du treillis d'hypothèse augmente rapidement et l'évaluation exhaustive devient trop coûteuse, et même impossible pour les systèmes *à Large Vocabulaire*. Pour résoudre ce problème, l'algorithme de décodage explore uniquement une partie de l'espace de recherche (seules les hypothèses prometteuses sont maintenues dans *le faisceau*).

2.6.2.2 Décodage asynchrone

Le décodage asynchrone est basé sur une exploration de type *best-first* (le meilleur d'abord), l'espace de recherche est parcouru en explorant le nœud le plus "prometteur" selon une règle spécifique. Les hypothèses sélectionnées sont étendues mot par mot sans contraindre leur terminaison à un même temps t.

En pratique, ce type d'algorithmes de décodage se base sur des piles qui contiennent les hypothèses à explorer. Les hypothèses sont ordonnées et les plus prometteuses (qui ont le meilleur score de vraisemblance par exemple) seront développées avant les autres. Cette méthode d'exploration nécessite la définition de :

– un critère d'ordonnancement des hypothèses ;
– une méthode d'évaluation de la qualité des chemins explorés ;
– un nombre d'hypothèses à garder en compétition.

L'algorithme A^* [Hart 1968] est l'exemple le plus utilisé de ces types de décodage. Cet algorithme est fondé sur une fonction de coût f définie pour chaque nœud n_i du graphe de recherche. La fonction $f(n_i)$ représente une estimation du coût du chemin passant par le nœud n_i et liant le nœud initial n_0 au nœud final n_F, comme présenté dans la figure 2.6. La fonction de coût f est définie par :

$$f(n_i) = g(n_i) + h(n_i) \tag{2.21}$$

avec $g(n_i)$, la fonction qui calcule la probabilité du chemin envisagé entre le nœud initial et le nœud n_i, et $h(n_i)$, est une fonction qui calcule une estimation (ici une maximisation) de la probabilité du meilleur chemin entre le nœud n_i et le nœud final.

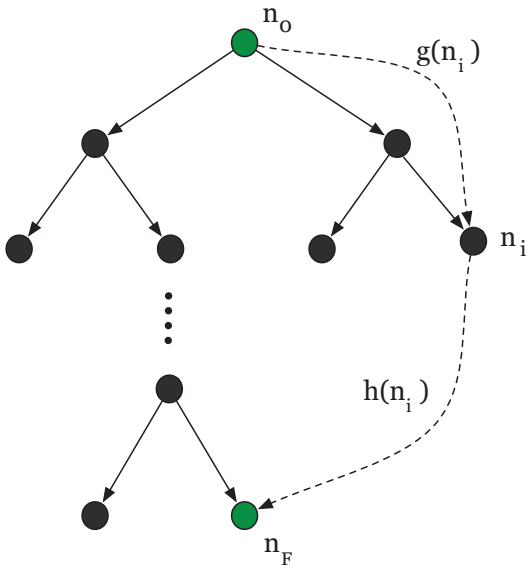


FIGURE 2.6 – Graphe de recherche et fonction de coût dans l'algorithme A^*.

Dans la procédure de recherche de l'algorithme A^*, nous distinguons deux sous-ensembles de nœuds :

- Les nœuds qui ont été développés (dont les successeurs sont disponibles pour la procédure de recherche) sont appelés "fermés". Ils sont placés dans la liste FERMÉ ;
- Les nœuds qui ont été générés et qui attendent leur développement sont appelés "ouverts". Ils sont placés dans la liste OUVERT par ordre de priorité (le premier à développer en tête de liste).

L'algorithme A^* se déroule de la manière suivante :

Algorithme A^*

---

1. Initialisation :

Mettre le nœud initial n_0 (le premier nœud du graphe de recherche $i = 0$) dans OUVERT.
Calculer $f(n_i) = g(n_i) + h(n_i)$
2. Test d'échec ? :
1. Si OUVERT est vide Alors sortir avec échec.
3. Verification fin ? :
Si n (le premier nœud de la liste OUVERT) est un nœud but (nœud terminal du graphe de recherche figure 2.6), sortir avec la solution obtenu en remontant les pointeurs vers n_0.
4. Développement :
Si n n'est pas le nœud but, explorer tous les nœud adjacents $n^{'}$. Pour chaque nœud $n^{'}$ successeur de n :

a. : Si $n^{'}$ n'est pas encore dans OUVERT ni dans FERMÉ, estimer $h(n^{'})$, calculer $f(n^{'}) = g(n^{'}) + h(n^{'})$ et ajouter $n^{'}$ dans OUVERT.

b. : Si $n^{'}$ est dans OUVERT, ajuster ses pointeurs pour le rattacher au chemin fournissant $g(n^{'})$ maximum.

c. : Si $n^{'}$ est dans FERMÉ, le remettre dans OUVERT.

5. Boucle :
Recommencer le processus à l'étape 2.

L'efficacité de l'algorithme A* est liée à la manière dont la fonction h est estimée. Si h est un estimateur parfait, alors l'algorithme A* convergera immédiatement vers le but sans développer des chemins inutiles.

2.7 Sorties d'un SRAP

Les systèmes de reconnaissance fournissent un texte représentant la transcription d'un signal sonore (la *one-best*). Il est également possible de retenir plusieurs hypothèses de reconnaissance. La sortie pourra alors être une liste de N meilleures hypothèses, un graphe de mots ou encore un réseau de confusion.

2.7.1 Liste de N meilleures hypothèses

Cette liste continent les N hypothèses trouvées pour chaque segment de parole. Ces hypothèses sont ordonnées selon le score établi par le système de transcription. Il est bien constaté que cette liste présente en général de nombreuses redondances.

2.7.2 Graphe de mots

Le graphe de mots est généralement utilisé comme un espace de recherche compact. Ce graphe est généré après une première passe de décodage, en utilisant des modèles simples. La première passe permet de réduire rapidement l'espace de recherche en ne conservant que la partie susceptible de contenir l'hypothèse optimale. L'espace de recherche ainsi obtenu fait l'objet d'une deuxième exploration exhaustive avec des modèles plus performants (un modèle de langage d'ordre supérieur par exemple).

Les graphes sont des structures dont les nœuds représentent des instants du signal et les arcs des hypothèses de mots accompagnés de leur vraisemblance acoustique et de leur probabilité linguistique. Ces structures contiennent beaucoup d'informations exploitables pour d'autres applications comme le calcul des mesures de confiance, la recherche d'informations, l'interprétation sémantique, la combinaison de systèmes *etc.*

2.7.3 Réseau de confusion

Bien que les graphes de mots offrent une représentation compacte de l'espace de recherche, ils peuvent être très gros en terme de nombre de nœuds et d'arcs, devenant ainsi difficilement manipulables. Les réseaux de confusion ont été proposés par [Mangu 1999], comme une transformation des graphes vers une une structure plus compacte.

Les réseaux de confusion peuvent être vus comme une transformation des graphes de mots où les nœuds sont fusionnés pour former un graphe linéaire avec des points de rencontre regroupant tous les nœuds du graphe initial

finissant approximativement à un même temps t. Les arcs entre deux nœuds successifs représentent les hypothèses de mots en compétition pour chaque tranche de temps. Un exemple de réseau de confusion est présenté dans la figure 2.7. Les arcs sont pondérés par la probabilité *a posteriori* de chaque mot w (y compris l'absence de mot modélisé par ε), et la meilleure solution est obtenue en minimisant les erreurs de transcription, et non en déterminant la succession de mots $\hat{W}$ possédant la plus grande probabilité $P(A|\hat{W})P(\hat{W})$, comme dans le cas des graphes de mots.

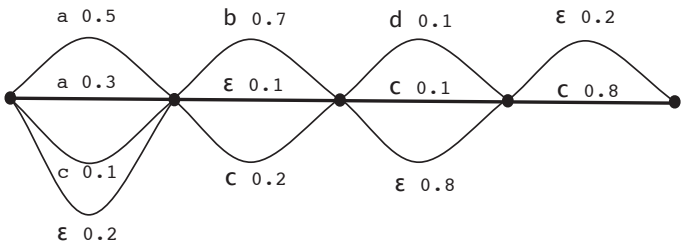


FIGURE 2.7 – Exemple de réseau de confusion.

2.7.4 Mesure de confiance

La mesure de confiance est un indice permettant d'estimer la qualité de la sortie d'un SRAP. Cet indice donne un aspect qualitatif de la sortie du système en associant à chaque mot w un score $CM(w)$ dans l'intervalle $[0, 1]$. Plus la valeur de $CM(w)$ est proche de 1 plus le mot est considéré comme correct. Ces scores sont généralement calculés à partir des probabilités *a posteriori* calculées sur les graphes de mots, les listes de N-meilleures hypothèses ou les réseaux de confusion. [Wessel 2001]. Un état de l'art sur les mesures de confiance est donné dans [Mauclair 2006].

Il est à noter toutefois que les mesures de confiance ne sont pas toujours fiables. Elles doivent donc être considérées avec prudence. Dans la littérature, plusieurs métriques pour évaluer les mesures de confiance on été développées. L'entropie croisée normalisée (NCE : Normalized Cross Entropy) [Stemmer 2002] est la métrique d'évaluation la plus utilisée. La NCE a été proposée par NIST [1] et utilisée pendant les campagnes d'évaluation. Elle permet d'évaluer l'information supplémentaire apportée par la mesure de confiance utilisée.

1. www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

2.8 Évaluation d'un SRAP

Les systèmes de transcription sont généralement évalués en réalisant un alignement entre les hypothèses de transcription et la transcription de référence. La métrique couramment utilisée est le taux d'erreur sur les mots (WER). Trois types d'erreurs sont prise en compte :

- Insertion (I) : le système propose un mot qui n'est pas présent dans la référence ;
- Suppression (D) : le système omet un mot présent dans la référence ;
- Substitution (S) : le système remplace un mot de la référence par un autre.

Le taux d'erreur est calculé par la formule suivante :

$$\textit{Taux d'erreur} = \frac{I + D + S}{\textit{nombre de mots dans la référence}} \tag{2.22}$$

2.9 Techniques de reconnaissance rapide

La reconnaissance de la parole grand vocabulaire est une tâche qui consomme beaucoup des ressources. Dans la littérature, plusieurs techniques ont été étudiées dans l'objectif de satisfaire des contraintes liées aux ressources disponibles et de répondre aux scénarios d'utilisation particuliers tout en préservant un niveau de performance acceptable. L'accélération du temps de décodage est généralement effectuée *via* la réduction de l'espace de recherche ou l'optimisation du temps de calcul des vraisemblances. Il existe également d'autres méthodes d'accélération basées sur la parallélisation des traitements en utilisant des nouvelles plateformes émergentes.

2.9.1 Réduction de l'espace de recherche

Même avec l'amélioration de la modélisation de l'espace de recherche et l'optimisation des algorithmes de décodage, l'exploration complète de l'espace de recherche d'un système *à Large Vocabulaire* reste irréaliste. C'est pourquoi les algorithmes de recherche n'essaient pas seulement de déterminer l'hypothèse de probabilité maximale, mais aussi de développer le moins possible de chemins du graphe de recherche. Les algorithmes de reconnaissance incluent souvent des stratégies de réduction de l'espace de recherche pour éliminer les hypothèses peu prometteuses, diminuer le temps de décodage et réduire l'espace mémoire occupé. Pour ce faire, les techniques d'élagage et de prédiction sont souvent employées.

L'élagage consiste à supprimer les hypothèses partielles les moins prometteuses dans le graphe de recherche. L'approche consiste à évaluer rapidement le potentiel de chaque branche et à supprimer les états dont le score est inférieur à un seuil donné, fixé empiriquement, ou bien les états dont le score peut être considéré comme faible par rapport au score du meilleur état courant.

La prédiction consiste à évaluer rapidement le potentiel de chaque branche. Durant le processus de décodage, lorsque la fin d'un phonème est atteinte, plusieurs hypothèses du phonème suivant sont envisageables. L'objectif de la prédiction [Ortmanns 1997] est de prédire la liste des hypothèses valables en utilisant un score approximatif calculé pour chacun des phonèmes suivants au moyen de quelques échantillons. Les scores calculés sont utilisés par la suite pour un élagage à seuil permettant la suppression des transitions correspondantes aux hypothèses de phones les moins prometteuses. La même technique, appliquée au niveau des mots, est appelée «*language model look-ahead*» [Ortmanns 1998]. Cette technique permet, pour un mot w_i avec un modèle de langage bigrammes par exemple, d'utiliser la probabilité du mot le plus important parmi tous les mots suivant w_k. La probabilité introduite est remplacée par la suite par celle du bigramme effectif.

En pratique, la réduction de l'espace de recherche permet d'obtenir de bons compromis entre la qualité des transcriptions automatiques générées, la durée de cette génération et la taille des graphes de mots. En revanche, cet élagage peut introduire des erreurs en éliminant des hypothèses potentiellement justes.

2.9.2 Optimisation de calcul de vraisemblance

L'utilisation des MMC avec un très grand nombre de gaussiennes permet d'obtenir de bonnes performances, mais engendre aussi un temps de calcul de vraisemblance pouvant atteindre 70% du temps global du décodage [Knill 1996].

Dans la littérature, plusieurs méthodes de calcul rapide des vraisemblances ont été proposées. Ces méthodes peuvent être classées en deux catégories selon qu'elles sont basées sur la réduction du nombre de paramètres ou sur l'utilisation de différentes astuces pour l'accélération du temps de calcul. Une description détaillée des méthodes d'optimisation de calcul de vraisemblance est présentée dans [Zouari 2007].

2.9.3 Parallélisation d'un SRAP

Différents travaux ont été effectués dans l'objectif d'utiliser efficacement les possibilités de parallélisme offertes par les nouveaux processeurs multicore. Dans le processus de la reconnaissance de la parole statistique, et durant l'évaluation des milliers d'interprétations possibles d'un énoncé de parole pour trouver l'interprétation la plus probable, il existe plusieurs possibilités de parallélisme. Dans [Phillips 1999], les auteurs présentent une re-implémentation parallèle d'un système de reconnaissance en utilisant une architecture multiprocesseur à mémoire partagée. L'implémentation, détaillée dans [Phillips 1999], permet de paralléliser l'algorithme de décodage *Viterbi*, le calcul de vraisemblance et la construction de l'espace de recherche.

Dans leurs travaux, les auteurs de [Vogelgesang 2011], proposent également une stratégie de parallélisation basée sur la séparation du processus de décodage en deux phases : le calcul des scores et le processus de recherche.

2.10 Système de reconnaissance du LIUM

Le développement d'un système de transcription complet est long, fastidieux et coûteux. Heureusement, il existe plusieurs SRAP, plus ou moins aboutis et performants, dans le monde du logiciel libre. Un ensemble, non exhaustif de logiciels libres est présenté dans [Estève 2004]. Pour réduire le coût de développement, le LIUM a choisi d'utiliser l'un de ces systèmes libres : le système CMU Sphinx. Libre depuis 2001, le système Sphinx est régulièrement maintenu par une grande communauté.

En se basant sur la boîte à outils Sphinx, plusieurs modifications ont été apportées par le LIUM afin de créer un système à l'état de l'art, performant et adapté à la langue française [Deléglise 2005]. La figure 2.8 [Estève 2009] résume l'architecture générale du système du LIUM ainsi que les étapes de développement du système avec des précisions sur les outils modifiés, ou créés, par le LIUM. Dans cette section nous allons détailler les composantes du système du LIUM utilisé pendant la deuxième campagne d'évaluation ESTER [2] [Deléglise 2009], son architecture, ainsi que les modifications et améliorations apportées pour augmenter sa performance.

2. Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques

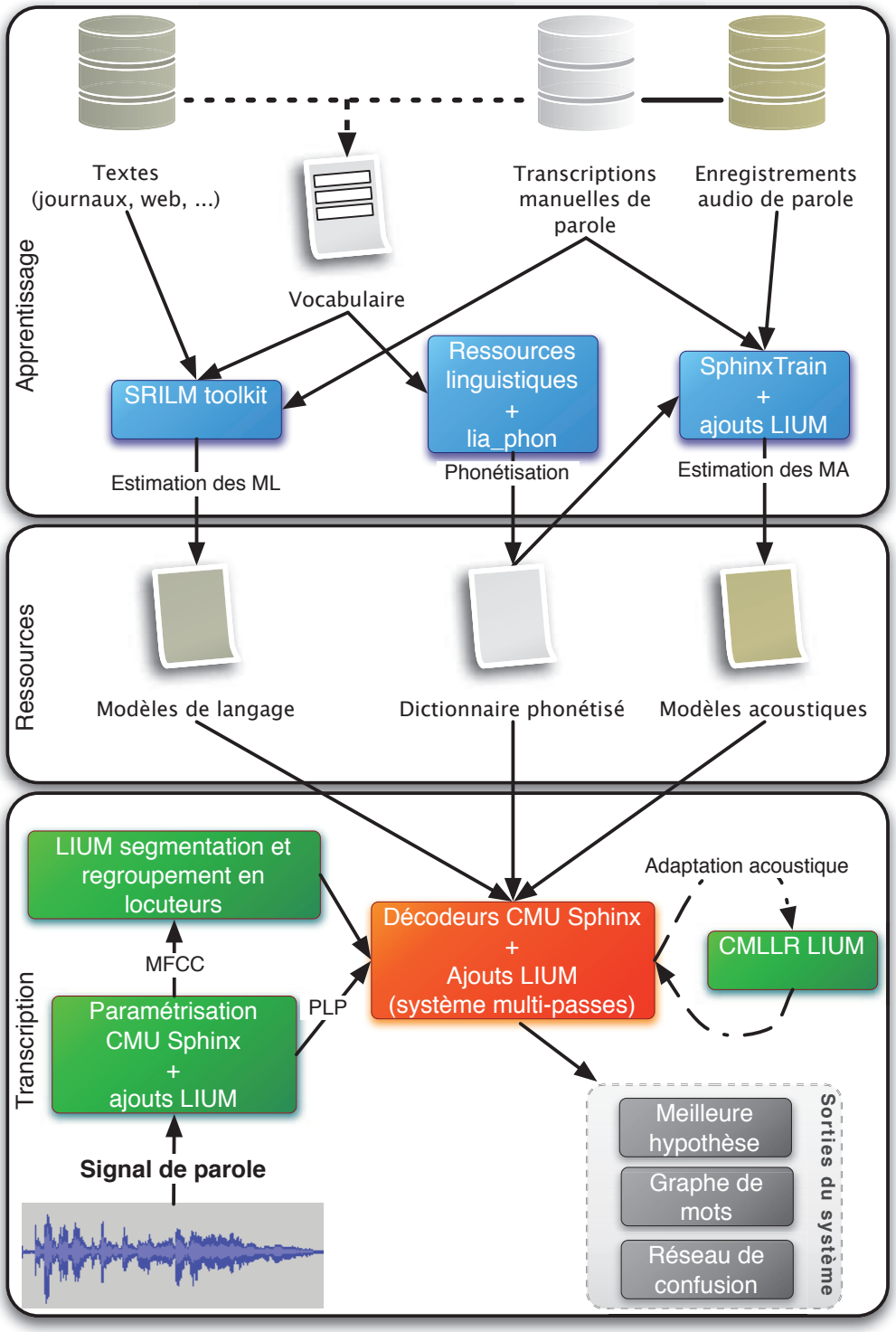


FIGURE 2.8 – Architecture générale de système du système de transcription du LIUM.

2.10.1 Apprentissage des modèles

Les modèles utilisés par un SRAP statistique nécessitent différents types de données d'apprentissage. Les données nécessaires pour estimer les modèles de langage sont généralement disponibles en grande quantité et à faible coût (sur le Web par exemple) mais souvent «hors domaine». En revanche, les corpus d'apprentissage des modèles acoustiques (transcription manuelle des données audio) sont rares et coûteux. Dans cette section nous présentons les données d'apprentissage, la construction du dictionnaire de prononciation et la phase d'apprentissage des modèles acoustiques et linguistiques.

2.10.1.1 Données d'apprentissage

Pour apprendre ses modèles, le LIUM utilise principalement les corpus distribués par les organisateurs de la campagne d'évaluation ESTER augmentés d'autres données. Les corpus ESTER sont répartis comme suit :

- 200h d'enregistrements d'émissions radiophoniques transcrites manuellement (provenant principalement de radios françaises, mais contenant également des enregistrements de radio africaines francophones) ;
- 40h d'enregistrements d'émissions radiophoniques de radio africaines francophones avec transcriptions manuelles rapides ;
- les articles du journal Le Monde de 1987 à 2006.

L'ensemble des participants à ESTER 2 ont également eu accès à 40h d'enregistrements radiophoniques transcrits manuellement, issus du projet EPAC [3] [Estève 2010] , contenant principalement de la parole conversationnelle.

Nous avons également eu accès aux données French Giga Word Corpus, qui regroupent un très grand nombre de dépêches AFP (Agence France Presse) et AWP (informations financières) diffusées entre les années 1990 et 2000. Ces corpus ont été enrichis ensuite par des données récoltées sur le Web pour améliorer la modélisation linguistique :

- les archives du journal L'Humanité de 1990 à 2007 ;
- les articles du site Libération disponibles en 2007 ;
- les articles du site L'internaute disponibles en 2007 ;
- les articles du site Rue89 disponibles en 2007 ;
- les articles du site Afrik.com disponibles en 2007.

3. http://projet-epac.univ-lemans.fr/doku.php

L'ensemble des articles du journal Le Monde et du French Giga Word Corpus représente environ 1 milliard de mots, et les données provenant du Web représentent 80 millions de mots.

2.10.1.2 Dictionnaire de prononciation

Le dictionnaire de prononciation constitue le lien entre la modélisation acoustique et la modélisation linguistique. Le développement de ce dictionnaire nécessite, généralement, trois étapes :

1. La sélection de l'ensemble des entrées lexicales : le vocabulaire ;
2. La définition des unités acoustiques élémentaires : jeu de phonèmes ;
3. La description de chaque entrée lexicale en utilisant ses unités acoustiques : la phonétisation.

La constitution du vocabulaire est une tâche très importante, d'une part ce dernier définit l'ensemble de mots qu'un SRAP est capable de reconnaître ; d'autre part, il influence les autres composants d'un SRAP, en particulier le processus de décodage : un grand vocabulaire augmente la taille de l'espace de recherche et alourdit en conséquence le processus de décodage.

Ayant comme objectif la réduction du taux des mots hors vocabulaire tout en gardant une couverture maximale du domaine visé, le vocabulaire du système du LIUM a été construit en suivant l'approche proposée dans [Allauzen 2004]. En résumé, cette approche consiste à :

1. Estimer un modèle unigramme par source d'apprentissage ;
2. Calculer les cœfficients d'interpolation entre ces modèles en minimisant la perplexité sur le corpus de développement (ici le corpus de développement d'ESTER 2) ;
3. Construire le modèle de langage unigramme avec les cœfficients d'interpolation calculés lors de l'étape précédente ;
4. Choisir la taille du vocabulaire N et extraire les N mots les plus probables du modèle de l'étape précédente. N est fixé à 120 000 pour notre système.

Après la sélection du vocabulaire, la deuxième étape consiste à définir le jeu de phonèmes. Dans notre système, nous utilisons un jeu de 35 phonèmes définis dans l'outil de phonétisation automatique LIA_PHON [Béchet 2001].

Le jeu de phonèmes choisi est utilisé pour phonétiser toutes les entrées du vocabulaire comme suit :

1. Si le mot existe dans BDLEX [Perennou 1987], nous utilisons la ou les phonétisation proposées par BDLEX ;
2. Sinon, nous utilisons l'outil LIA_PHON.

2.10.1.3 Modèles acoustiques

Pour apprendre les modèles acoustiques, nous disposons au total d'environ 280h d'enregistrements audio transcrits manuellement. La paramétrisation appliquée au signal audio est de type *PLP* constituée par 12 descripteurs issus de l'analyse *PLP* plus l'énergie et les dérivées premières et secondes de ces descripteurs.

Les données d'apprentissage contiennent l'information sur le type de canal ainsi que le genre des locuteurs. Ces informations peuvent être utilisées pour spécialiser les modèles. La répartition des données d'apprentissage en fonction de la taille de la bande passante est présentée dans le tableau 2.1.

Canal	**Durée**
Bande Large (Studio)	**220h**
Bande Étroit (Téléphone)	**60h**

TABLE 2.1 – Répartition des données d'apprentissage acoustique par type de canal.

Partant du fait que les modèles acoustiques appris sur des données homogènes offrent de meilleures performances, nous utilisons l'information présentée dans le tableau précédent pour apprendre des modèles acoustiques spécialisés. Les premiers modèles appris sont des modèles dépendant du type de la bande passante (Studio/Téléphone).

En plus de l'information sur le type de canal, le corpus d'entraînement contient aussi l'information sur le genre du locuteur. Nous avons utilisé cette information pour adapter les modèles dépendant de la taille de la bande passante au genre du locuteur. Les modèles sont adaptés par Maximum *A Posteriori* (MAP). Au final, quatre modèles acoustiques sont construits :

- Homme-Studio ;
- Femme-Studio ;
- Homme-Téléphone ;
- Femme-Téléphone.

Tous ces modèles sont composés de 6500 états partagés, chaque état étant modélisé par un mélange de 22 gaussiennes.

Par la suite, chaque modèle a été utilisé dans un processus d'apprentissage adaptatif (SAT : Speaker Adaptative Training). Les quatre modèles, déjà estimés, représentent les modèles initiaux dans le processus d'adaptation décrit dans la section 2.3.3.2. À l'issue de cette adaptation, quatre nouveaux modèles ont été créés en augmentant le nombre d'états partagés à 7500, tout en gardant le même nombre de gaussiennes par état.

2.10.1.4 Modèle de langage

Comme pour la majorité des SRAP, les modèles de langage du système du LIUM sont des modèles *n-grammes*. Des modèles *3-grammes* sont utilisés pour les trois premières passes et des modèles *4-grammes* sont utilisés pour les dernières. Le tableau 2.2 présente la répartition des mots du corpus d'apprentissage en fonction de leur origine.

	Transcription manuelle d'émissions radiophoniques	Presse écrite et dépêches	données récoltées sur le Web
Nombre de mots	**3,3 M**	**1,0 G**	**80 M**

TABLE 2.2 – Nombre de mots dans le corpus d'apprentissage du modèle de langage en fonction de la source.

Avant la création des modèles, les données présentées dans le tableau 2.2 sont d'abord normalisées. La normalisation sert, entre autre, à harmoniser les données (mettre sous forme de mots certains symboles par exemple) et à éviter les graphies multiples des mêmes mots.

Chaque corpus d'apprentissage a été utilisé pour estimer un modèle n-gramme en utilisant la technique de *discounting* dite de Kneser-Ney modifiée [Kneser 1995, Chen 1999] avec interpolation des *n-grammes* d'ordres inférieurs. Tous les *n-grammes* observés dans le corpus d'apprentissage, même une seule fois, sont pris en compte (*i.e* pas de cut-off). Ensuite, sur le corpus de développement approprié, les coefficients d'interpolation ont été optimisés avec l'algorithme E.M (Expectation Maximisation) afin de minimiser la mesure de perplexité du modèle interpolé sur ce corpus. Ces manipulations ont été réalisées à l'aide de la boîte à outils SRILM [Stolcke 2002].

2.10.2 Transcription

Les modèles précédemment appris sont utilisés pendant la transcription. La transcription est réalisée sur deux étapes ; la segmentation et le décodage. Nous décrivons dans cette section le processus de segmentation et les étapes de décodage.

2.10.2.1 Segmentation et regroupement en locuteur

En plus de la parole, le signal audio contient souvent d'autres événements sonores (musiques, publicité, pauses ...). La transcription de ces types d'événements est inutile et introduit des insertions qui réduisent la performance du système. Pour éviter la dégradation de performance et supprimer les segments de non-parole, le signal audio est d'abord traité par un système de segmentation.

Le système de segmentation permet aussi de regrouper les segments acoustiquement homogènes, et de donner des informations supplémentaires sur le locuteur, son genre et la largeur de bande utilisée. Ces informations sont généralement utilisées pour mettre en œuvre une adaptation acoustique plus efficace (voir section 2.3.3) et utiliser ainsi le modèle acoustique approprié à chaque segment.

Le LIUM a développé un outil interne pour la segmentation et le regroupement en locuteur. L'outil est basé sur le Critère d'Information Bayesien (BIC) et se compose de trois étapes :

- Découpage du signal en petits segments acoustiquement homogènes ;
- Regroupement des segments en classes (une par locuteur) ;
- Ajustement des frontières.

Le système de segmentation et regroupement en locuteur du LIUM a terminé premier lors de la campagne d'évaluation ESTER 2. Il est présenté en détails dans [Meignier 2010] et une documentation en ligne[4] est disponible.

2.10.2.2 Processus de décodage

Le SRAP du LIUM est, comme la majorité de systèmes actuels, un système de reconnaissance multi-passes. Le système est constitué de 5 passes séquentielles. Chaque passe utilise en entrée la sortie de la passe précédente et propose une nouvelle hypothèse de reconnaissance :

4. http://lium3.univ-lemans.fr/diarization/doku.php/welcome

1. La première passe consiste en un traitement utilisant la version 3.7 du décodeur rapide de Sphinx 3 appliquée sur des paramètres acoustiques PLP ; cette passe utilise un modèle de langage 3-gramme et des modèles acoustiques adaptés au genre du locuteur (homme/femme) et aux conditions acoustiques (studio/téléphone).

2. La seconde passe utilise de nouveau la version Sphinx 3.7 appliqué aux mêmes paramètres acoustiques PLP, avec une matrice de transformation CMLLR calculée de façon à adapter les paramètres acoustiques aux modèles acoustiques. Ces modèles ont été estimés en utilisant les méthodes SAT et MPE (Minimum Phone Error).

3. La troisième passe permet de pallier les approximations inter-mots faites par le décodeur Sphinx 3.7 lors du calcul des scores acoustiques des phonèmes : afin d'accélérer fortement le traitement, les scores des phonèmes situés en fin de mots ne sont pas calculés en utilisant leur véritable contexte droit, mais à partir d'une approximation (décodeur *lextree*). En utilisant le graphe de mots généré lors de la seconde passe comme espace de recherche, il est possible de corriger ces imprécisions inter-mots en utilisant, puisqu'il est connu *a priori*, le vrai contexte droit des phonèmes en fin de mot. Ce sont les mêmes modèles acoustiques et linguistiques que lors de la seconde passe qui sont utilisés, avec bien entendu l'application de la même matrice de transformation CMLLR sur les paramètres acoustiques.

4. La quatrième passe consiste à recalculer à l'aide d'un modèle *4-grammes* les scores linguistiques des mots du graphe générés lors de la passe précédente.

5. Enfin, la passe 5 transforme le graphe de mots issu de la quatrième passe en un réseau de confusion. La méthode de consensus [Mangu et al., 2000] est alors appliquée afin d'obtenir l'hypothèse de reconnaissance finale avec, pour chaque mot, des probabilités *a posteriori* utilisables comme mesures de confiance.

2.10.3 Performance du système

Le système de transcription précédemment décrit a été utilisé pendant la campagne d'évaluation ESTER 2 et le le tableau 2.3 montre l'évolution du taux d'erreur sur les mots au fur et à mesure des passes de traitement sur l'ensemble du corpus de test. Entre la première passe et la dernière passe, nous rapportons une baisse relative de 29,15% du taux d'erreur mot.

Passe	WER
1. Modèles acoustiques génériques et modèle de langage trigramme	27,1 %
2. Adaptation acoustique	22,5 %
3. Rescoring acoustique de graphe de mots	20,4 %
4. Rescoring de graphe de mots avec modèle quadrigramme	19,4 %
5. Extraction de réseau de confusion (CN) et décodage du CN	19,2 %

TABLE 2.3 – Evolution du WER en fonction de la passe de décodage du SRAP sur l'ensemble du corpus de test ESTER 2

Le système du LIUM est le système utilisé, entre autres, pour mener les expériences présentées dans ce manuscrit. C'est un système de reconnaissance à l'état de l'art utilisé dans plusieurs campagnes d'évaluation, dans différentes langues.

2.11 Conclusion

Dans ce chapitre, nous avons introduit les principes de fonctionnement et les différents constituants d'un système de reconnaissance automatique de la parole. Nous avons présenté, ensuite, le système de transcription du LIUM comme exemple de système de transcription *à Large Vocabulaire* et nous avons détaillé les différentes étapes de construction du système.

Le développement d'un SRAP *à Large Vocabulaire* nécessite la mise en place d'une chaine de traitement complexe constituée d'un ensemble d'étapes faisant appel à des choix arbitraires. Ces choix constituent une source de variabilité car la modification d'une décision engendre la création d'un système différent avec un comportement différent.

Étant donné qu'il n'existe pas de chaîne de traitements permettant de construire un système de transcription idéal, des méthodes de combinaison de systèmes ont été proposées pour exploiter les éventuelles complémentarités entre différents systèmes et augmenter la robustesse des SRAP. Nous présentons dans le chapitre suivant, un état de l'art sur les méthodes de combinaison des SRAP.

Chapitre 3
Combinaison des SRAP

Sommaire

Bien que la majorité des systèmes de reconnaissance de la parole soient, à l'heure actuelle, basés sur des méthodes statistiques, ils peuvent se différencier sur plusieurs points (méthodes de paramétrisation du signal, modélisation acoustique et linguistique, algorithmes de décodage...). Dans le but d'exploiter ces différences et d'accroître la robustesse des systèmes, il a été proposé de combiner plusieurs systèmes de reconnaissance afin de profiter de leurs éventuelles complémentarités pour construire une transcription finale améliorée.

La combinaison des systèmes de reconnaissance de la parole a suscité un intérêt croissant ces dernières années et diverses approches ont été proposées. Ces approches opèrent à différents niveaux. Dans ce chapitre, nous présentons, dans un premier temps, les méthodes de génération des systèmes complémentaires ainsi que les mesures de degré de complémentarité entre systèmes. Nous exposons par la suite un tour d'horizon des différentes stratégies de combinaison des SRAP.

3.1 Complémentarité des systèmes

L'amélioration obtenue après la combinaison de différents SRAP n'est pas liée uniquement à la performance de la méthode de combinaison utilisée, mais aussi au degré de complémentarité des systèmes combinés. En effet, la combinaison des systèmes complémentaires permet d'augmenter la précision de la transcription finale [Breslin 2006]. Dans cette section nous présentons les méthodes de génération des systèmes complémentaires ainsi que les mesures de leur degré de complémentarité.

3.1.1 Génération des systèmes complémentaires

Comme nous l'avons précisé précédemment, la combinaison de plusieurs systèmes n'est pas utile à moins qu'ils soient complémentaires. En conséquence, différentes méthodes de génération de systèmes complémentaires ont été proposées. Ceux-ci peuvent être obtenus par une modification directe de l'une des composantes d'un SRAP (paramétrisation, modèles linguistiques et acoustiques, algorithmes d'apprentissage et de décodage), tout en espérant une complémentarité entre les composantes modifiées. Toutefois, il existe des méthodes d'apprentissage explicites où les systèmes sont construits pour être complémentaires et, par conséquent, leur combinaison garantit la réduction du taux erreur mot final.

3.1.1.1 Arbre de décision et partage d'états aléatoire (*tying*)

Durant le processus d'apprentissage des modèles acoustiques, l'augmentation du nombre des modèles nécessite une grande quantité de données d'apprentissage pour modéliser tous les contextes (par exemple pour les 35 unités phonétiques de la langue française il faut estimer théoriquement $35^3 = 42875$ modèles triphones au maximum). Pour résoudre ce problème, des modèles contextuels avec partage d'états (*tying*) sont généralement utilisés. Le principe est de regrouper les états des modèles qui sont proches en utilisant un arbre de décision phonétique.

Dans [Siohan 2005], le partage d'états a été utilisé pour la génération des systèmes complémentaires. Les auteurs proposent un partage d'états aléatoire permettant la modélisation de différents groupes d'unités acoustiques. La méthode consiste à sélectionner aléatoirement une question parmi les N meilleures questions linguistiques. Cela dit, le *tying* aléatoire ne garantit pas la complémentarité entre les systèmes générés. Cependant, la génération de plusieurs systèmes augmente la probabilité que certains arbres puissent capter différentes informations et générer, par conséquent, des systèmes complémentaires. En pratique, l'application du *tying* aléatoire permet une réduction de taux d'erreur mot en combinant *a posteriori* les différents systèmes obtenus.

Contrairement au *tying* aléatoire, [Breslin 2009] propose une approche permettant de générer des modèles explicitement complémentaires. Les arbres de décision sont générés de manière à résoudre explicitement les confusions faites par le système de base. Par conséquent, les questions linguistiques ne sont plus choisies aléatoirement, mais en fonction de la modélisation de base. Le choix des questions linguistiques permet de modéliser le sous-ensemble de données d'apprentissage mal modélisées par le système de base.

3.1.1.2 Combinaison de classifieurs (*Boosting*)

Le *Boosting* est un principe issu du domaine de l'apprentissage automatique [Schapire 2003]. L'objectif principal est d'améliorer la précision de classification en utilisant un grand nombre de classifieurs dits «faibles», c'est-à-dire ceux qui obtiennent des résultats de classification légèrement meilleurs que le hasard. *Adaboost* [Freund 1995] est l'algorithme de *Boosting* le plus utilisé. L'algorithme change itérativement la distribution des données d'apprentissage tout en augmentant le poids des exemples d'apprentissage mal modélisés. L'apprentissage est par la suite appliqué en respectant la distribution obtenue, permettant une concentration sur les exemples «difficiles». Les classifieurs «faibles» sont par la suite combinés, par exemple, en utilisant une combinaison *ROVER* pondérée.

Pour appliquer le *Boosting* à la reconnaissance automatique de la parole, le problème est généralement reformulé comme une tâche de classification des phonèmes. Dans [Zweig 2000], plusieurs modèles sont appris, un par phonème, et le *Boosting* est appliqué au niveau de la trame. Dans [Suh 2007], le *Boosting* est appliqué à la partie réseau de neurones (NN) dans un système de reconnaissance hybride (HMM/NN).

D'autres méthodes de génération de systèmes complémentaires ont été

développées, comme les méthodes basées sur l'apprentissage de plusieurs systèmes en parallèle. Au final, [Venkataramani 2005] propose une méthode appelée «diviser pour régner» (*divide-and-conquer approach*). Cette approche est basée sur l'analyse des treillis issus de la première passe d'un SRAP pour repérer les régions où le système a une grande incertitude et apprendre ensuite des modèles spécifiques pour ces régions.

3.1.2 Mesures de complémentarité entre systèmes

La combinaison de systèmes est une technique performante pour améliorer le taux de reconnaissance. Cependant, l'amélioration n'est pas systématique puisqu'elle dépend du degré de complémentarité entre les systèmes combinés et de l'efficacité de la méthode de combinaison utilisée.

Dans ce contexte, l'évaluation du degré de complémentarité entre systèmes offre une connaissance *a priori* permettant la sélection des systèmes à combiner pour une meilleure performance. Deux SRAP sont dits complémentaires s'ils font des erreurs différentes. En effet, le fait d'avoir deux systèmes de reconnaissance différents ne garantit pas leur complémentarité. Dans cette section nous présentons les méthodes utilisées pour évaluer le degré de complémentarité entre systèmes.

3.1.2.1 Dépendance d'erreurs entre systèmes

Dans [Burget 2004], une méthode de mesure de complémentarité entre SRAP a été proposée. Les auteurs cherchent à estimer la dépendance des erreurs entre les systèmes. Les erreurs de reconnaissance sont classées en deux catégories :

- Erreurs simultanées (*simultaneous error*) : lorsque les deux systèmes font une erreur en même temps ;
- Erreurs dépendantes (*dependent error*) : lorsqu'ils font la même erreur.

La mesure de complémentarité entre deux systèmes est basée sur le calcul des erreurs *simultanées* et des erreurs *dépendantes*. La mesure est estimée sur un ensemble de segments comme suit :

1. chaque segment est décodé par les deux systèmes ;
2. les sorties de deux systèmes sont alignées segment par segment ;
3. les erreurs dépendantes et simultanées sont comptées.

La dépendance d'erreur est estimée avec les deux mesures suivantes : le $LBWER$ (*Lower Bound Word Error Rate*), qui représente le ratio entre les erreurs simultanées et le nombre total des mots dans la référence, et le $DWER$ (*Dependent Word Error Rate*), qui correspond au ratio entre les erreurs dépendantes et le nombre total des mots dans la référence.

Pour deux systèmes i et j, le $LBWER$ et le $DWER$ sont calculés par :

$$LBWER(i,j) = \frac{N_{sim}(i,j)}{N_{ref}} \times 100 \tag{3.1}$$

$$DWER(i,j) = \frac{N_{dep}(i,j)}{N_{ref}} \times 100 \tag{3.2}$$

avec $N_{sim}(i,j)$ le nombre total d'erreurs simultanées, $N_{dep}(i,j)$ le nombre total d'erreurs dépendant et N_{ref} le nombre total des mots dans la référence.

Si l'on dispose d'un ensemble de systèmes S, une matrice par type d'erreur est construite. Par exemple, la matrice $LBWER$ contient la valeur $LBWER$ pour chaque paire de systèmes ($LBWER(i,j)$ $\forall i,j \in S$).

Ces mesures ont été utilisées dans le cadre d'une combinaison de type $ROVER$ (détaillée dans la section 3.3.1) pour sélectionner l'ensemble optimal de systèmes à utiliser dans la combinaison. Les travaux de [Burget 2004] ont montré une corrélation entre ces mesures de complémentarité et les résultats de la combinaison $ROVER$ basée sur la fréquence de mots.

3.1.2.2 Taux d'erreur *oracle*

Afin de mieux juger le degré de complémentarité entre les systèmes combinés ainsi que la performance de la méthode de combinaison, il est intéressant de calculer le gain *potentiel*.

Le gain *potentiel* est généralement calculé avec le taux *oracle* en utilisant la transcription de référence. L'*oracle* correspond à une combinaison idéale. Dans le cadre de la combinaison $ROVER$, cela correspond au choix de la meilleure hypothèse à chaque branche du réseau de confusion. L'*oracle* fournit la borne maximale que l'on peut atteindre avec la méthode de combinaison utilisée.

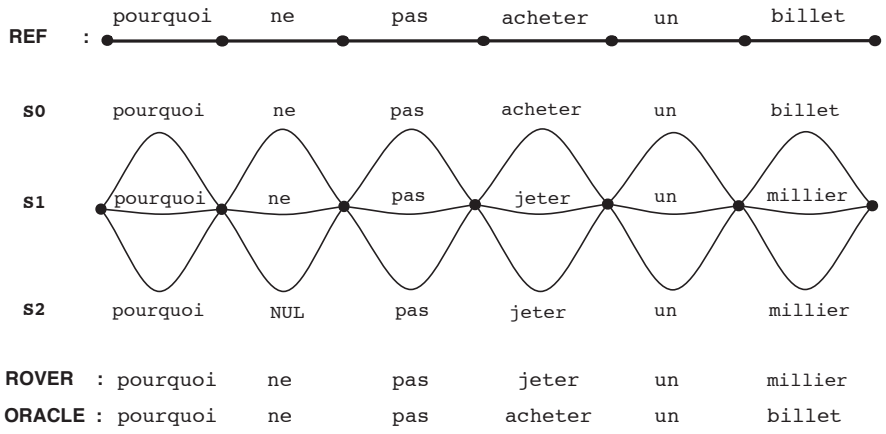


FIGURE 3.1 – La sortie de la combinaison *ROVER* entre trois systèmes plus la sortie oracle en utilisant la transcription de référence.

Dans la figure 3.1, la combinaison *ROVER* est basée sur la fréquence de mots. Bien que le système *S0* propose la transcription correcte, le processus de vote, basé sur la fréquence des mots, introduit des erreurs dans la sortie finale. Contrairement à la combinaison *ROVER*, la combinaison *oracle* utilise la référence pour produire la sortie finale et garantir la sélection de la bonne hypothèse si elle est proposée par, au moins, un des systèmes combinés. La sortie *oracle* correspond donc à une combinaison *ROVER* avec un processus de décision parfait.

3.2 Combinaison avant le processus décodage

3.2.1 Combinaison des paramètres acoustiques

L'extraction de paramètres acoustiques est la première étape de construction d'un SRAP. Elle consiste à transformer le signal de la parole en vecteurs de paramètres acoustiques représentant l'information utile pour la reconnaissance. Dans la section 2.2 du chapitre précédent, nous avons décrit les méthodes de paramétrisation les plus utilisées dans les SRAP actuels. Chaque méthode de paramétrisation correspond à une manière différente de représenter le signal de la parole. Dans le but d'accroître la robustesse des systèmes, il a été proposé de combiner plusieurs méthodes de paramétrisation [Ellis 2000]. Cette combinaison a été motivée par l'hypothèse que certaines caractéristiques du signal de parole sont accentuées par certains jeux de paramètres et ignorées par d'autres.

La combinaison consiste à concaténer différents jeux de paramètres acoustiques en un seul flux avant l'apprentissage du modèle acoustique comme présenté dans la figure 3.2. Le modèle acoustique est appris par la suite de manière classique, avec le jeu de paramètres résultant de la concaténation.

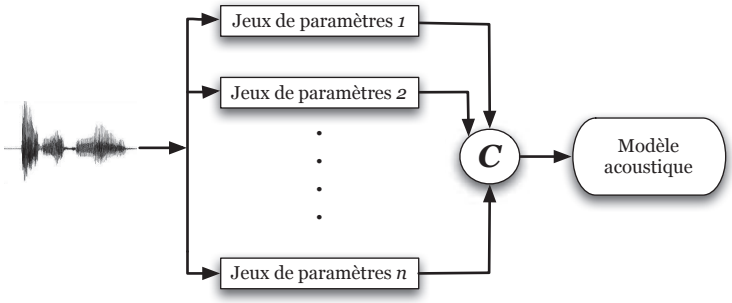


FIGURE 3.2 – Combinaison des paramètres acoustiques

Bien que la concaténation des différents vecteurs acoustiques permette de capter davantage d'information du signal de parole, elle rend la modélisation plus complexe et augmente la taille des vecteurs de paramètres obtenus. Pour remédier à cela, la dimension des vecteurs est généralement réduite en utilisant par exemple une analyse de composante principale (*Principal Component Analysis*) ou une analyse linéaire discriminante (*Linear Discriminant Analysis*).

3.2.2 Combinaison des modèles acoustiques

Une autre méthode de combinaison avant le processus de décodage consiste à utiliser différents modèles acoustiques. Les modèles acoustiques peuvent se différencier par la méthode de modélisation utilisée (des modèles HMM/GMM ou des modèles HMM/ANN par exemple), ou encore par le jeu de paramètres acoustiques en entrée de chaque modèle.

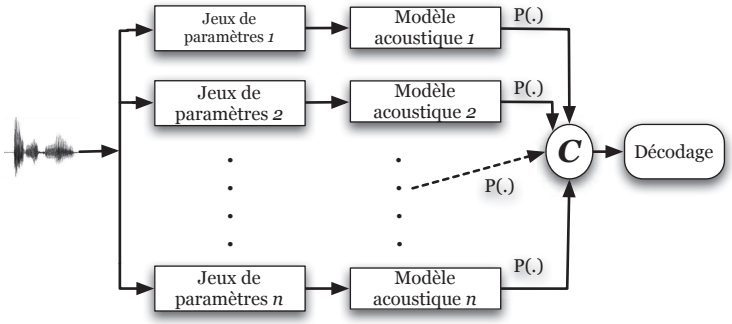


FIGURE 3.3 – Combinaison des modèles acoustiques

La combinaison des modèles acoustiques a été proposée par [Zolnay 2005] et [Kirchhoff 1998]. L'approche consiste à combiner les probabilités obtenues avec les différents modèles (voir figure 3.3). Les probabilités obtenues varient selon le type de modèle acoustique utilisé, des probabilités *a posteriori* (systèmes hybrides HMM/ANN) ou des vraisemblances (modèles HMM/GMM classiques). Pour une combinaison cohérente, il faut prendre en compte plusieurs aspects dépendant des flux d'observations acoustiques. De plus, une étape de normalisation des probabilités estimées par les différents modèles est indispensable.

3.2.3 Combinaison et adaptation des modèles de langage

La qualité d'un modèle de langage dépend de son domaine d'application. Lorsque les données d'apprentissage et de test sont très différentes, il est nécessaire d'adapter le modèle générique aux conditions de test. Le principe général d'adaptation est présenté dans le chapitre 1 (figure 2.3).

Différentes techniques d'adaptation ont été proposées dans la littérature. Nous présentons dans cette section les techniques les plus utilisées dans le domaine de la reconnaissance de la parole.

3.2.3.1 Interpolation de modèles

Pour la construction d'un modèle de langage, on dispose généralement de plusieurs corpus différents. L'approche la plus simple consiste à fusionner les différents corpus dont on dispose en un grand corpus utilisé pour estimer le modèle *n-grammes*. Cependant, les différents corpus n'ont pas la même taille ni le même contenu. Par conséquent, les corpus moins volumineux seront mal représentés dans le modèle, même si ces corpus sont plus pertinents pour la tâche visée. Contrairement à l'approche précédente, [Jelinek 1980] propose d'interpoler plusieurs modèles estimés sur chacun des corpus comme suit : si l'on dispose d'un ensemble de M modèles de langage représentés par leur distribution P_m avec $1 \leq m \leq M$, l'interpolation est définie par :

$$P_{interpola}(w_i|h_i^n) = \sum_{m=1}^{M} \lambda_m \; P(w_i|h_i^n) \tag{3.3}$$

avec λ_m le coefficient de pondération qui définit l'importance de chaque modèle m dans l'interpolation, la somme des coefficients de pondération doit être égale à 1 ($i.e \sum_{m=1}^{M} \lambda_m = 1$). Les coefficients d'interpolation sont généralement estimés pour réduire la perplexité du modèle sur un corpus représentatif de la tâche visée.

3.2.3.2 Modèles caches et modèles «triggers»

Le modèle cache se base sur l'idée que les mots apparus récemment ont plus de chance d'être ré-utilisés, dans le même discours et dans un futur proche. Ce modèle a été présenté par [Kuhn 1990] qui propose de renforcer les probabilités des mots utilisés dans un passé proche durant le processus de recherche de mot courant. Le renforcement est effectué avec un changement de probabilité linguistique comme suit : la probabilité d'un mot étant donné un historique de taille n et un cache sur une fenêtre de taille m est définie par :

$$P(w_i|w_1, ..., w_{i-1}) \simeq \lambda\ P_c(w_i|w_{i-m+1}, ..., w_{i-1}) + (1-\lambda)P(w_i|w_{i-n+1}, ..., w_{i-1}) \tag{3.4}$$

avec λ le poids accordé au modèle cache, par rapport à $1-\lambda$ le poids du modèle initial.

Le modèle cache a été généralisé par [Lau 1993] puis amélioré par [Singh-miller 2007]. Cette généralisation apparaît sous le nom de modèle *trigger*. Contrairement aux modèles caches, le renforcement n'est pas appliqué uniquement aux mots gardés dans le cache, mais aussi pour l'ensemble des mots du lexique appartenant au même champ lexical ou à la même thématique prédéfinit en avance.

3.2.4 Adaptation croisée

L'adaptation des SRAP est une technique largement utilisée dans une architecture de reconnaissance multi-passes. Cette technique permet l'amélioration des performances du système en utilisant les sorties de la passe précédente pour l'adaptation des modèles de la passe en cours. Le processus peut être appliqué itérativement d'une passe à l'autre. Toutefois l'adaptation d'un système sur ses propres sorties n'apporte plus d'amélioration au bout de deux ou trois itérations.

L'adaptation croisée (*cross adaptation*) permet de surmonter cette limitation. Elle est basée sur l'utilisation de la sortie d'un premier système pour adapter les modèles d'un deuxième. Cela permet de propager les informations entre différents systèmes et d'améliorer ainsi les performances du système adapté. Néanmoins, pour une combinaison efficace, il est important que les deux SRAP utilisés aient des performances comparables [Stüker 2006]. La figure 3.4, présente le principe de l'adaptation croisée où les modèles du *système 2* sont adaptés en utilisant la sortie du *système 1*. Cependant, dans la pratique, rien n'empêche l'application de l'adaptation dans les deux sens.

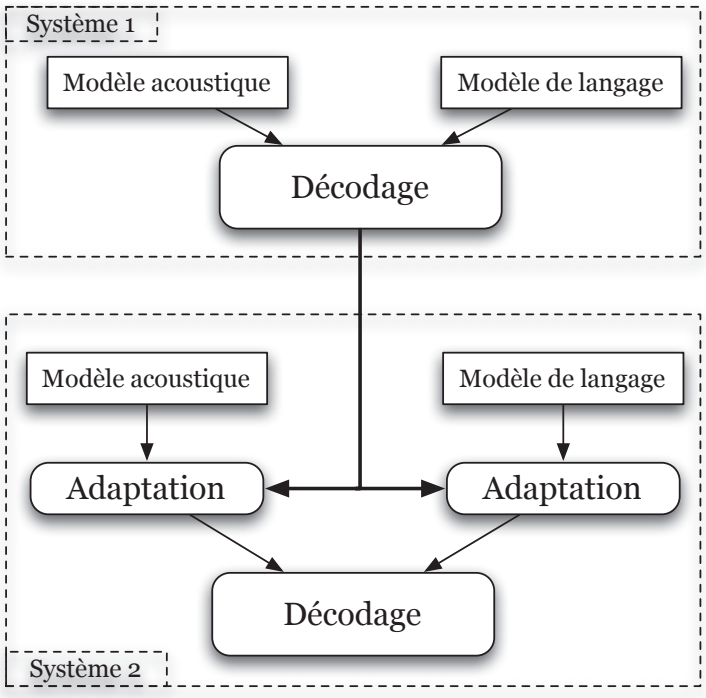


FIGURE 3.4 – Principe de l'adaptation croisée entre deux systèmes

3.2.4.1 Adaptation croisée des modèles acoustiques

L'adaptation croisée des modèles acoustiques est utilisée dans de la majorité des SRAP à l'état de l'art. L'adaptation peut être appliquée en utilisant les sorties de la passe précédente du même système dans le cadre d'une *auto-adaptation*, mais aussi en utilisant les sorties d'un autre système dans le cadre d'une *cross-adaptation*. Après la récupération de données d'adaptation, toutes les techniques présentées dans la section 2.3.3 peuvent être utilisées pour appliquer l'adaptation.

3.2.4.2 Adaptation croisée du modèle de langage

Dans le cadre des SRAP, le modèle de langage utilisé est généralement la mélange de plusieurs modèles appris sur différents corpus. L'adaptation non supervisée du modèle de langage semble être une approche intéressante pour adapter le système à une tâche particulière ou un style de parole particulier et pour améliorer, par conséquent, la robustesse du système.

Puisque l'adaptation directe des probabilités des *n-grammes* nécessite une grande quantité de données, non disponibles dans le cas d'une adaptation non supervisée, l'adaptation est plutôt appliquée avec une mise à jour des

poids d'interpolation du modèle en utilisant les sorties d'une passe précédente, du même système dans le cadre d'une *auto-adaptation*, ou avec les sorties d'un autre système dans le cadre d'une *cross-adaptation*. Dans [Liu 2010], les auteurs présentent une extension de la fonction d'interpolation pour ajuster dynamiquement la contribution de chaque modèle de langage dans le modèle final. La fonction 3.3, où les poids d'interpolation globaux λ_m ne tiennent pas compte des contextes, est transformée comme suit :

$$P_{interpola}(w_i|h_i^n) = \sum_{m=1}^{M} \phi_m(h_i^n) P(w_i|h_i^n) \tag{3.5}$$

avec $\phi_m(h_i^n)$ le $m^{\text{ème}}$ poids de contexte h_i^n.

Les poids $\phi_m(h_i^n)$ peuvent être estimés par maximum de vraisemblance [Liu 2009]. L'adaptation croisée a été aussi étendue pour adapter des modèles de langage neuronaux en ajoutant une couche d'adaptation [Liu 2011].

3.3 Combinaison après le processus de décodage

La combinaison post-décodage opère sur les sorties des SRAP. Comme nous l'avons vu dans le chapitre précédent (section 2.7), un système de reconnaissance de la parole peut fournir des sorties sous différents formats. Dans l'objectif d'améliorer la qualité de la transcription, les sorties de différents systèmes de reconnaissance peuvent être combinées. Dans cette section, nous décrivons les méthodes de combinaison des sorties les plus utilisées dans les systèmes à l'état de l'art.

3.3.1 Combinaison par vote majoritaire : *ROVER*

La combinaison par vote majoritaire a été proposée par [Fiscus 1997]. L'algorithme consiste à combiner *a posteriori* les meilleures sorties de plusieurs systèmes de reconnaissance. Le *ROVER* (*Recognizer Output Voting Error Reduction*) se déroule en deux étapes successives : une étape d'alignement suivie d'une étape de vote. Les hypothèses sont d'abord itérativement alignées pour obtenir un réseau de confusion, l'algorithme d'alignement est identique à celui utilisé pour aligner les hypothèses de transcription à la référence pendant le calcul de taux d'erreur mots. Le réseau de confusion construit dans l'étape précédente est ensuite parcouru et le processus de vote est appliqué au niveau des mots. En l'absence d'information *a priori* sur la qualité de sortie de chaque système le choix est basé sur la fréquence d'apparition (vote majoritaire). L'algorithme est présenté dans la figure 3.5 où les sorties de 4 systèmes $S0...S3$ sont alignées, et les mots avec le plus grand nombre d'occurrences sont choisis.

S3	NUL	est	pourquoi	il	NUL	NUL
S2	NUL	est	pourquoi	NUL	le	pays
S1	NUL	ce	pourquoi	il	a	payé
S0	c'	ce	pourquoi	il	a	payé
ROVER	NUL	ce	pourquoi	il	a	payé

FIGURE 3.5 – Combinaison *ROVER* de sorties *one−best* de plusieurs systèmes.

Avec *ROVER*, l'ordre des systèmes combinés est important pour le module d'alignement et celui de vote. Dans la figure 3.5, la procédure de vote est basée sur la fréquence des mots. La transcription du système $S0$ est utilisée comme *pivot* durant le processus d'alignement. En effet, l'hypothèse du système *pivot* est sélectionnée, lorsqu'un ou plusieurs mots ont la même fréquence d'apparition (par exemple le deuxième mot dans la figure 3.5).

Le *ROVER* basé sur la fréquence de mots ne donne pas forcément la meilleure solution. En effet, lorsque plusieurs mots ont la même fréquence d'apparition, l'algorithme choisit l'hypothèse proposée par le système *pivot* qui n'est pas toujours la meilleure hypothèse. Pour éviter ce problème, d'autres critères de vote sont utilisés dans *ROVER* comme *maxconf* (vote par maximum de score de confiance) ou *avgconf* (vote par score de confiance moyen). Une description détaillée de l'ensemble des critères de vote possible avec *ROVER* est disponible dans [Fiscus 1997].

3.3.2 *ROVER* assisté par un modèle de langage

Dans *ROVER*, le choix du mot numéro k s'effectue entre les n transcriptions disponibles sans prendre en considération les $k-1$ mots déjà choisis. Pour prendre en compte l'information linguistique une amélioration de *ROVER* a été proposée dans [Schwenk 2000]. Les informations linguistiques sont introduites au sein de l'algorithme de décision. Lorsque le *ROVER* ne peut pas prendre une décision, le modèle linguistique apporte sa contribution, améliorant nettement le résultat de la combinaison.

3.3.3 *iROVER* : combinaison *ROVER* améliorée

La performance de la combinaison *ROVER* est sensible à la méthode de vote utilisée (fréquence des mots, mesures de confiance...). Dans [Hillard 2007], les auteurs augmentent l'ensemble de paramètres utilisés durant le processus de vote. Six classes de paramètres sont utilisées :

1. Le nombre d'hypothèses identiques entre les systèmes ;
2. La distance d'édition entre les hypothèses de différents systèmes ;
3. Scores de confiance étendus (le nombre de choix sur le noeud du réseau de confusion, score acoustique, score linguistique...) ;
4. Information liée à l'aspect temporel de la meilleure hypothèse de chaque système (nombre de caractère, durée des trames, nombre de trames par caractère ...) ;
5. Les erreurs les plus fréquentes : cet critère définit, pour chaque mot, s'il appartient à la liste de n mots les plus fréquemment mal reconnus par le système ;
6. Des scores de confiance basés sur le décodage par min-fWER (*minimum Time Frame Word Error*) (voir section 3.4.2).

Après le calcul de l'ensemble de ces paramètres pour chaque système, un classifieur est entraîné sur un corpus de développement dont la transcription de référence est fournie. L'objectif est d'entraîner le classifieur à prendre la bonne décision en cas de désaccords entre les systèmes combinés.

Dans [Utsuro 2005], une méthode de combinaison similaire a été proposée en utilisant un classifieur différent. Les auteurs utilisent un classifieur de type *SVM*[1]. 26 systèmes, différenciés par leurs modèles acoustiques, ont été utilisés pour la combinaison. Les résultats obtenus montrent un gain relatif de l'ordre de 23% par rapport à une combinaison *ROVER*.

3.3.4 Combinaison d'hypothèses

Dans [Singh 2001], les auteurs proposent une méthode de combinaison basée sur la construction d'un graphe à partir des meilleures solutions de différents systèmes. Les hypothèses sont d'abord fusionnées dans un graphe où chaque mot est représenté par un nœud. Les liens entre les nœud sont par la suite ajoutés en utilisant l'information temporelle : un arc est ajouté entre deux noeuds $n1$ et $n2$ si le mot $n2$ commence au moins après 30 ms de la fin de $n1$. Après la construction de graphe de mots, la meilleure hypothèse est obtenue *via* un processus de ré-évaluation linguistique de ce graphe.

1. Support Vector Machine.

3.3.5 *BAYCOM* : Combinaison bayésienne

L'algorithme de combinaison BAYCOM a été proposé par Ananth Sankar dans [Sankar 2005]. La combinaison est basée sur la théorie de décision Bayésienne pour une combinaison optimale entre les sorties de plusieurs SRAP. Contrairement à la méthode de combinaison *ROVER*, la combinaison BAYCOM est basée sur une théorie standard de reconnaissance de forme.

La combinaison BAYCOM suppose que les systèmes mis en concurrence sont indépendants, elle calcule un nouveau score de confiance optimal par mot pour chaque système combiné. Les sorties sont par la suite alignées et les mots qui possèdent le score de confiance le plus élevé sont choisis. La combinaison BAYCOM permet de combiner plusieurs systèmes selon un critère Bayésien et elle obtient de meilleurs résultats qu'un *ROVER*.

3.3.6 Combinaison des réseaux de confusion : *CNC*

Une des limitations de la combinaison *ROVER* réside dans le fait qu'elle est limitée aux séquences des meilleures solutions (les *one-best*). En conséquence, seules les hypothèses de mots qui ont été choisies, individuellement, par l'un des systèmes pourront être choisies comme résultat final. Afin de surmonter cette limitation, la combinaison a été étendue à des réseaux de confusion [Evermann 2000]. En effet, les sorties *one-best* sont remplacées par des réseaux de confusion provenant de différents systèmes. Le processus d'alignement, basé sur une comparaison des mots dans *ROVER*, a été remplacé par un calcul de la probabilité de similarité entre mots, étant donné les réseaux de confusion.

Le processus de vote est effectué par une maximisation de la somme des probabilités *a posteriori* sur l'ensemble des nœuds comme suit :

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{N} P(S_i) P(w|X, S_i) \tag{3.6}$$

Tout comme dans la combinaison *ROVER*, l'ordre dans lequel les systèmes sont combinés est important dans la combinaison *CNC*. Les deux méthodes restent aujourd'hui les plus utilisées dans les systèmes de reconnaissance à l'état de l'art.

3.3.7 Combinaison des treillis

Dans la partie précédente, les treillis sont d'abord transformés en réseaux de confusion, puis combinés. Contrairement à cela, [Li 2002] propose de combiner directement les treillis issus de plusieurs systèmes. La combinaison des treillis nécessite des opérations de composition : les treillis sont d'abord fusionnés dans un seul treillis (fusion des noeuds de départ et d'arrivée de chaque treillis). Les noeuds et les arcs sont par la suite édités et trois opérations sont effectuées :

1. Fusion des arcs : les arcs de différents treillis sont fusionnés lorsqu'ils ont le même mot sur leurs noeuds de départ, les mêmes informations temporelles (trame de début et de fin), et lorsqu'il pointent vers des noeuds ayant le même phonème de départ.

2. Création des nouveaux arcs : un arc est ajouté entre les deux nœud A et B s'il existe un arc entre A et C et si les mots sur les noeuds B et C commencent par le même phonème et si la différence entre leur temps de départ ne dépasse pas un certain seuil (*i.e* 30 ou 40 ms). Le score acoustique de nouvel arc est assigné selon la formule suivante :

$$W_{A\to B} = \frac{D_{A\to B}}{D_{A\to C}} \,.\, W_{A\to C} \tag{3.7}$$

   avec $W_{A\to B}$ et $W_{A\to C}$ les scores acoustiques respectifs des arcs $A \to B$ et $A \to C$. $D_{A\to B}$ et $D_{A\to C}$ représentent leur durée.

3. Normalisation des scores acoustiques : étant donné que la distribution des scores acoustiques des treillis combinés n'est pas homogène entre les systèmes (différents modèles acoustiques), les scores acoustiques sont normalisés avant leur combinaison.

Après la combinaison des treillis des différents systèmes, un décodage de type *Viterbi* permet d'obtenir le résultat de la combinaison. Cette méthode permet l'exploitation des sorties plus riches et offre, par conséquent, une réduction plus importante du WER par rapport aux méthodes de combinaison *a posteriori* classique.

3.4 Combinaison durant le décodage

3.4.1 Espace de recherche intégré

[Chen 2006] propose une méthode de combinaison qui prend en compte l'ensemble de l'espace de recherche de tous les systèmes mis en concurrence. La méthode est basée sur la fusion de l'espace de recherche de plusieurs SRAP. La combinaison, présentée dans la figure 3.6, est décomposée en deux opérations : la fusion des différents espaces de recherche dans un seul graphe et la ré-évaluation de ce nouveau graphe.

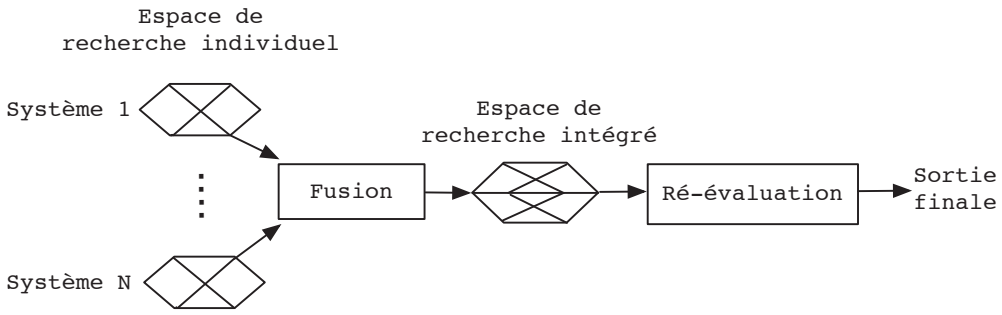


FIGURE 3.6 – Combinaison de systèmes par intégration des espaces de recherche.

La formalisation de l'opération de fusion des espace de recherche est la suivante : soient $G_1, G_2, ..., G_N$ les graphes issus des N SRAP. On considère premièrement deux graphes G_1 et G_2, avec q_1 et q_2 représentant un arc entre deux mots dans les treillis respectifs G_1 et G_2. Les arcs q_1 et q_2 peuvent s'écrire de la façon suivante : $q_1 = [W_i; t_{start}; t_{end}]$ pour le mot W_i produit par le système $S1$ entre t_{start} et t_{end}, et $q_2 = [W_j; \tau_{start}; \tau_{end}]$ pour le mot W_j produit par le système $S2$ entre τ_{start} et τ_{end}. De plus, un score $s(q)$ est associé à chaque arc. L'égalité entre deux arcs issus de systèmes différents est définie par :

$$q_1 = q_2 \quad ssi \quad \begin{cases} w_i & = & w_j \\ t_{start} & = & \tau_{start} \\ t_{end} & = & \tau_{end} \end{cases} \tag{3.8}$$

En cas d'égalité, les deux arcs sont fusionnés et le score de l'arc résultant est calculé en fonction des scores individuels :

$$score(q = q1 + q2) = combiner(score(q_1), score(q_2)) \tag{3.9}$$

À partir de ces équations, la combinaison de deux graphes G_1 et G_2 est alors définie par :

$$G_1 + G_2 = \{q = q_1 + q_2 | q_1 = q_2\} \cup \{q_1 | q_1 \notin G_2\} \cup \{q_2 | q_2 \notin G_1\} \tag{3.10}$$

Le graphe résultat de la fusion contient les arcs additionnés ou combinés s'ils sont égaux. La généralisation à N systèmes est définie par :

$$G = G_1 + G_2 + ... + G_N = \sum_{i=1}^{N} G_i \tag{3.11}$$

Après la fusion des espaces de recherche, quatre approches de ré-évaluation des scores (fonction *combiner* de l'équation 3.9) du graphe résultat de la fusion ont été proposées.

La première est basée sur une combinaison par consensus au niveau lexical. La deuxième est basée sur une granularité au niveau phonétique. La troisième est une combinaison de deux premières méthodes. La dernière méthode est inspirée du fWER (section 3.4.2). Dans la suite, nous détaillons la première approche. Le lecteur pourra se référer à [Chen 2006] pour une description détaillée de chacune des autres approches.

En ce qui concerne la première approche, les scores de chaque arc q, avant la fusion, correspondent à la probabilité *a posteriori* $P_i(q)$ du système i calculée avec l'algorithme *forward-backward*. Après le calcul des probabilités *a posteriori*, les scores des arcs dans l'espace de recherche résultat de la fusion sont calculés par :

$score(q = q_1 + q_2) = score(q_1) + score(q_2)$ si $q_1 = q_2$.
$score(q) = P_i(q)$ si q est généré uniquement par le système i.
$score(q) = \sum_{i=1}^{K}$ si q est généré par K systèmes.

Après la génération du nouveau graphe et la normalisation des scores, l'équation utilisée pour trouver le meilleur chemin dans ce graphe est la suivante :

$$w^* = (q^1 \ . \ q^2 ... q^M) = \arg\max_{w \in W, q^k \in w} \prod_{k=1}^{M} score(q^k) \tag{3.12}$$

3.4.2 Combinaison par fWER

Dans l'objectif de combiner des systèmes qui utilisent une segmentation différente et qui produisent des graphes avec une structure différente, [Hoffmeister 2006] propose une méthode de combinaison basée sur le décodage par fWER (*frame Word Error Rate*). Avec cette combinaison, la règle de décision ne prend pas en compte le contexte, car elle se base uniquement sur les probabilités *a posteriori*. Le décodage par min-fWER (Minimum fWER) a été aussi généralisé à G graphes issus de N systèmes. Néanmoins les résultats obtenus sont très similaires à ceux obtenus avec *ROVER* ou une combinaison de réseaux de confusion.

3.4.3 Décodage guidé

Le décodage guidé est une méthode de combinaison basée sur l'intégration d'une source d'information additionnelle dans le processus de décodage. Cette technique à été utilisée au départ pour exploiter des transcriptions approchées et améliorer la qualité de transcription [Lecouteux 2006]. Dans [Lecouteux 2007], cette technique a été étendue pour combiner plusieurs systèmes en remplaçant les transcriptions approchées par les sorties des systèmes de transcription auxiliaires. La combinaison des systèmes par décodage guidé est détaillée dans le chapitre suivant (chapitre 4).

3.5 Conclusion

La combinaison des systèmes de reconnaissance de la parole offre une solution permettant l'amélioration de la performance de la transcription. Dans ce chapitre, nous avons présenté, dans un premier temps, le notion de complémentarité entre systèmes ainsi que les méthodes de génération des systèmes complémentaires.

Nous avons survolé également les différentes approches de combinaison des SRAP. Les combinaisons peuvent s'effectuer à tous les niveaux (modèles acoustiques et linguistiques, paramétrisation acoustique, algorithme de recherche et sorties de systèmes). Nous avons divisé les méthodes de combinaison sur trois classes : les méthodes appliquées avant, après ou durant le processus de décodage.

Les méthodes de combinaison *a posteriori* restent, aujourd'hui, les méthodes les plus utilisées. Ces méthodes exploitent les sorties des différents systèmes sous leurs différents formats (*one-best*, réseau de confusion et treillis).

Deuxième partie

Mes contributions

CHAPITRE 4

Décodage guidé par sacs de n-grammes

Sommaire

4.1 Introduction

Afin d'améliorer la qualité de la transcription dans de multiples situations de test et grâce à l'augmentation de la puissance de calcul et de la quantité de mémoire disponible, les SRAP peuvent utiliser, aujourd'hui, une très grande quantité d'information lors de l'apprentissage. Cependant, les données d'apprentissage ne représentent qu'un échantillon restreint de l'ensemble des variabilités possibles du signal de la parole.

Face à ce problème, de nouvelles voies de recherche ont été explorées et des solutions alternatives ont été proposées pour chaque composante d'un SRAP. Le développement de systèmes hétérogènes, ainsi que les méthodes de combinaison abordées dans le chapitre précédent, provient de ces solutions alternatives. La combinaison de plusieurs SRAP hétérogènes à différents niveaux semble constituer une approche intéressante pour obtenir une meilleure transcription avec des systèmes plus robustes. Néanmoins, les méthodes de combinaison pré- et post-décodage restent, aujourd'hui, les méthodes les plus étudiées et les plus utilisées.

Dans ce chapitre, nous nous intéressons aux méthodes de combinaison intervenant durant le processus de décodage. Nous étudions dans un premier temps, la combinaison de systèmes par décodage guidé (DDA : *Driven Decoding Algorithm*). Nous nous focaliserons ensuite sur la robustesse de cette méthode. Enfin, nous proposons une amélioration du DDA permettant une généralisation efficace et performante.

4.2 Combinaison de systèmes par décodage guidé (DDA)

L'algorithme de décodage guidé (DDA) a été initialement proposé comme une technique d'exploitation des informations contenues dans des transcriptions imparfaites afin d'améliorer les performances d'un SRAP [Lecouteux 2006]. Des transcriptions approchées sont intégrées comme sources d'informations supplémentaires *via* une procédure d'alignement suivie d'une ré-évaluation des scores linguistiques du SRAP. Le DDA a été utilisé par la suite comme méthode de combinaison de systèmes en remplaçant les transcriptions approchées par les sorties d'un autre SRAP dit auxiliaire.

4.2.1 Principe de la combinaison utilisant DDA

Le décodage guidé modifie dynamiquement l'exploration de l'espace de recherche [Lecouteux 2007]. Il procède par la recherche de points de synchronisation entre les hypothèses d'un système primaire et celles d'un système auxiliaire. Cette recherche est réalisée en utilisant un alignement dynamique (DTW : *Dynamic Time Warping*) entre les sorties du système auxiliaire et les résultats partiels de décodage du système primaire. Un score de correspondance est calculé ensuite en fonction du nombre de mots correctement alignés. Le score de correspondance est utilisé pour modifier la probabilité linguistique des hypothèses du système primaire en utilisant la formule suivante :

$$L(w_i|w_{i-2}, w_{i-1}) = P(w_i|w_{i-2}, w_{i-1})^{1-\alpha(w_i)} \tag{4.1}$$

avec $P(w_i|w_{i-2}, w_{i-1})$ la probabilité initiale du trigramme (w_i, w_{i-2}, w_{i-1}) et $\alpha(w_i)$ le score de correspondance calculé *via* une mesure de similarité entre hypothèses du système primaires hw_i et celles du système auxiliaire w_i. Ce score de correspondance est donné par :

$$\alpha(w_i) = \begin{cases} \frac{\phi(w_i)+\phi(w_{i-1})+\phi(w_{i-2})}{3} & \text{si } (hw_i, hw_{i-1}, hw_{i-2}) = (w_i, w_{i-1}, w_{i-2}) \\ \frac{\phi(w_i)+\phi(w_{i-1})}{2} & \text{si } (hw_i, hw_{i-1}) = (w_i, w_{i-1}) \\ \phi(w_i) - \gamma & \text{si } (hw_i) = (w_i) \text{ et } \phi(w_i) \geq \gamma \\ 0 & \text{si } (hw_i) \neq (w_i) \text{ ou } \phi(w_i) < \gamma \end{cases} \tag{4.2}$$

avec $\phi(w_i)$ la mesure de confiance du mot w_i et γ un seuil fixé empiriquement.

La combinaison par décodage guidé a été testée sur trois heures extraites du corpus de développement de la campagne ESTER 1 [Galliano 2006]. Les résultats obtenus montrent une amélioration significative avec la combinaison des systèmes par rapport aux systèmes seuls [Lecouteux 2007].

4.2.2 Généralisation de la combinaison DDA

Afin d'exploiter des sorties de systèmes auxiliaires plus riches qu'une simple *one-best*, une généralisation de la combinaison par décodage guidé a été proposée dans [Lecouteux 2008b]. Cette généralisation consiste à réaliser un décodage guidé par réseaux de confusion. Dans ce cas de figure, l'alignement est réalisé en minimisant la distance d'édition entre l'hypothèse et le réseau de confusion. Cet alignement permet l'extraction de la meilleure projection de l'hypothèse partielle sur le réseau. La ré-évaluation linguistique est similaire à celle utilisée dans le décodage guidé par la meilleure sortie (équation 4.2).

Les résultats obtenus avec le décodage guidé par réseaux de confusion sont limités à ceux de la combinaison avec la *one-best*. En effet, les réseaux de confusion ne permettent pas seulement d'intégrer davantage d'information dans le processus de la combinaison, mais ils intègrent aussi du bruit, qui limite l'apport de la richesse d'information intégré *via* les réseaux de confusion.

4.2.3 Discussion sur la combinaison DDA

Contrairement aux méthodes de combinaison classiques, où chaque système développe indépendamment ses hypothèses avant la combinaison, le DDA s'apparente plutôt à une méthode de collaboration de systèmes. En effet, tous les systèmes *auxiliaires* participent à la construction des hypothèses de transcription du système *primaire*. Les travaux présentés dans [Lecouteux 2008a], ont montré que la combinaison DDA est plus performante que la combinaison ROVER. Dans la suite, nous nous intéressons à la combinaison DDA pour étudier :

- la robustesse de la combinaison par décodage guidé vis-à-vis de la qualité de la transcription auxiliaire et l'écart de performance entre les systèmes combinés ;
- la généralisation de la combinaison DDA pour une combinaison efficace avec plusieurs systèmes auxiliaires.

4.3 Robustesse de la combinaison DDA

Afin de tester la robustesse de la combinaison par décodage guidé, nous reprenons les expériences menées dans [Lecouteux 2008b]. Les SRAP (LIUM, SPEERAL (système du LIA) et IRÈNE (système de l'IRISA)) ont été évalués sur les trois mêmes heures issues du corpus de développement d'ESTER 1 : une heure de France Inter, une heure de France Info et une heure de la Radio France International. La différence par rapport aux expériences antérieures est que nous utilisons le système le plus performant comme système primaire.

4.3.1 Cadre expérimental

L'ensemble de nos expériences a été effectué avec trois systèmes de reconnaissance différents : le système du LIUM est utilisé comme système *primaire* et ceux du LIA et de l'IRISA comme systèmes *auxiliaires*. Dans cette section nous présentons brièvement les différents systèmes utilisés.

4.3.1.1 Système de reconnaissance du LIUM

Le système de reconnaissance du LIUM utilisé durant ces expériences est celui mis en place lors de la campagne d'évaluation ESTER 1. Le système est basé sur le décodeur du laboratoire CMU (Sphinx 3.3) avec une modélisation acoustique continue basée sur des MMC [Deléglise 2005]. Le processus de décodage et composé de trois passes de décodage : la première passe utilise

un modèle de langage tri-grammes et des modèles acoustiques dépendant du genre et de la bande. La meilleure sortie de la première passe est utilisée par la suite durant la deuxième passe pour l'adaptation CMLLR avec le même modèle de langage et des modèles acoustiques SAT. La deuxième passe fournit un treillis d'hypothèses réévaluées durant la troisième passe avec un modèle de langage quadri-grammes.

4.3.1.2 Système de reconnaissance du LIA : SPEERAL

SPEERAL est un système de reconnaissance large vocabulaire pour la parole continue. Le processus de décodage est basé sur un algorithme A^*, qui opère sur un treillis de phonèmes, avec une modélisation linguistique type n-grammes et des modèles acoustiques basés sur des Modèles de Markov Cachées contextuels à états partagés. Les paramètres acoustiques utilisés sont composés de 12 coefficients PLP, plus l'énergie, et leurs dérivées première et seconde. Le processus complet de décodage, présenté lors de l'évaluation ESTER 2, se décompose en trois passes : la première utilise des modèles acoustiques dépendant du genre et de la bande de fréquence audio, ainsi qu'un modèle de langage tri-grammes. La sortie de la première passe est utilisée en deuxième passe pour adapter les modèles acoustiques avec la méthode MLLR. La dernière passe est une ré-évaluation de graphes de mots issus de la deuxième passe avec un modèle de langage quadri-grammes. Le système du LIA est présenté en détails dans [Nocera 2004].

4.3.1.3 Système de reconnaissance de l'IRISA : IRÈNE

Le système Irène est organisé de manière modulaire autour d'un décodeur de type *beam search*. Le système fonctionne en quatre passes de décodage : la première passe utilise un modèle de langage tri-grammes et un modèle acoustique non-contextuel pour générer un treillis de mots. Le treillis généré est ensuite réévalué, dans la deuxième passe, en utilisant un modèle de langage quadri-grammes et des modèles acoustiques contextuels pour obtenir la meilleure hypothèse utilisée durant la troisième passe pour une adaptation MLLR des modèles acoustiques. Un décodage final est appliqué sur les 1000 meilleures hypothèses, sorties de la passe précédente, en utilisant des informations morpho-syntaxiques [Huet 2007].

4.3.2 Performance des systèmes

Les trois heures de test sont initialement transcrites en utilisant uniquement les deux premières passes de chaque système. Le taux d'erreur mots de chaque système est reporté dans le tableau 4.1.

Systèmes	F.Inter	F.Info	RFI
LIUM-base	**19,34** %	**17,92** %	**22,59** %
SPEERAL	22,52 %	21,97 %	24,95 %
IRÈNE	21,96 %	21,61 %	26,03 %

TABLE 4.1 – Taux d'erreur mot du système LIUM, du système SPEERAL et du système IRÈNE sur les trois heures issues du corpus de développement d'ESTER 1.

En se basant sur ces résultats, nous avons assigné à chaque système son rôle. Ainsi le système du LIUM a été définit comme *primaire* et les systèmes SPEERAL et IRÈNE comme *auxiliaires*.

Dans l'implémentation initiale de DDA, le processus d'alignement entre les hypothèses du système *primaire* et la transcription *auxiliaire* a été effectué par une DTW. Dans notre implémentation, cet alignement a été remplacé par un alignement direct en utilisant les informations temporelles fournies avec les transcriptions auxiliaires. En effet, les SRAP fournissent, pour chaque mot reconnu, des informations temporelles (début et durée) exploitées pour une recherche rapide des points de synchronisation. La ré-évaluation linguistique est effectué avec la formule 4.1.

4.3.3 Résultats de la combinaison

Afin d'estimer la qualité de la combinaison DDA implémentée avec le système du LIUM, nous avons testé, dans un premier temps, une combinaison avec comme transcriptions auxiliaires, celles du système IRÈNE. Les résultats de la combinaison sont reportés dans le tableau 4.2.

Systèmes	F.Inter	F.Info	RFI
LIUM	19,34%	17,92%	22,59%
IRÈNE	21,96%	21,61%	26,03%
LIUM-DDA-IRÈNE	**18.94**%	**17.59**%	**22.54**%

TABLE 4.2 – Taux d'erreur mot par radio de la combinaison DDA implémentée dans le système du LIUM en utilisant la sortie du système IRÈNE.

Ces résultats montrent la robustesse de la combinaison par décodage guidé : la combinaison permet d'obtenir une réduction statistiquement significative du taux d'erreur mot même lorsque le système primaire est guidé

par des transcriptions issues des systèmes auxiliaires moins performants. En effet, sur la radio France Inter par exemple, la combinaison DDA améliore de 0,4% le taux d'erreur mot d'un système ayant un WER initial de 19,34% en utilisant une transcription auxiliaire moins précise de 2,62% points en absolu (21,96% de WER).

4.3.4 Analyse de la combinaison DDA

Afin d'étudier le comportement de la combinaison DDA, telle qu'elle est implémentée dans le système du LIUM, nous avons effectué une analyse de la sortie de la combinaison DDA ainsi que du résultat du processus d'alignement. Nous avons remarqué que le score linguistique des hypothèses communes entre système primaire et transcription auxiliaire n'est pas toujours modifié. Cela est causé par des erreurs d'alignement dues à la différence de limites des mots entre systèmes combinés.

Pour évaluer l'impact des erreurs introduites par ce problème d'alignement nous avons conduit une expérience qui consiste à modifier la contrainte d'alignement temporel. Lorsque la contrainte d'alignement est stricte (cas du DDA) le processus d'alignement cherche un alignement exact : l'hypothèse du système primaire $w_i(pri)$, est comparée uniquement à l'hypothèse $w_j(aux)$ proposée par le système auxiliaire avec un chevauchement temporel. Lorsque le mot $w_i(pri)$ est différent de $w_j(aux)$, et pour une contrainte d'alignement avec relâchement égal à 1, le mot $w_i(pri)$, est comparé aussi avec $w_{j-1}(aux)$ et $w_{j+1}(aux)$. De la même manière, pour un relâchement égal à n, et tant qu'on ne trouve pas un point de synchronisation, $w_i(pri)$ est comparé avec les n mots avant et après $w_j(aux)$. Après la détermination des points de synchronisation entre l'hypothèse courante et la transcription auxiliaire, le score linguistique est modifié selon l'équation 4.1 avec l'exception que le score de correspondance est fixé pour un historique de deux mots (seuls les scores des trigrammes sont modifiés).

La figure 4.1 présente la variation de taux d'erreur en fonction du relâchement de la contrainte d'alignement. La relaxation totale de la contrainte d'alignement correspond à une recherche exhaustive des trigrammes proposés par le système primaire dans la transcription auxiliaire du segment en cours de traitement.

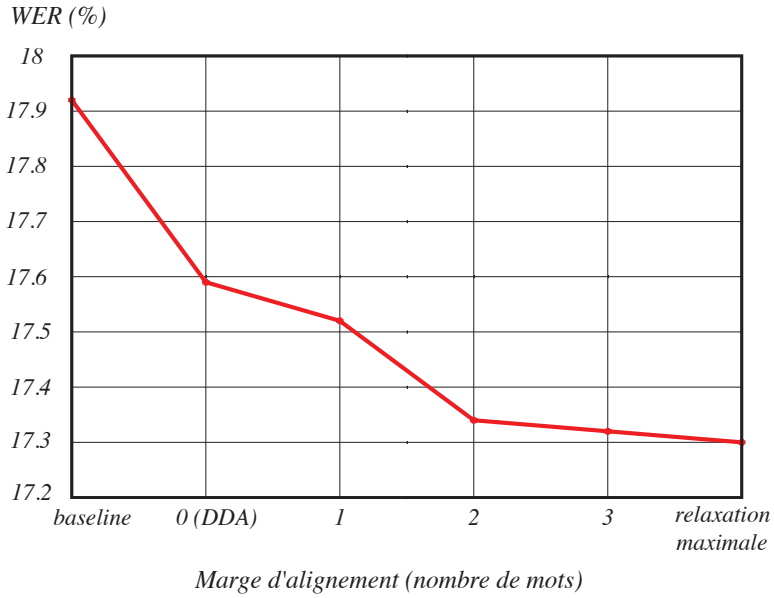


FIGURE 4.1 – Variation du taux d'erreur mot sur la radio France Info en fonction de la marge d'alignement entre les hypothèses du système primaire et celles de la transcription auxiliaire issue du système de l'IRISA.

Le meilleur score est obtenu avec un relâchement total de la contrainte d'alignement dans la limite du segment. Sur ces premiers résultats, il apparaît clairement que le relâchement de cette contrainte permet d'éviter la perte introduite par l'alignement temporel. Nous avons obtenu un gain additionnel de **0,29%** sur la radio France info et par conséquent une réduction totale du WER de **0,62%** point en absolu à partir d'un WER initial de **17,92%**.

4.4 Décodage guidé par sacs de n-grammes (BONG)

Avec le relâchement de la contrainte d'alignement et la ré-évaluation linguistique limitée aux trigrammes, nous avons réussi à obtenir un gain par rapport à la combinaison DDA comme implémentée dans le système du LIUM. Dans cette section, nous détaillerons d'abord cette nouvelle stratégie de combinaison par décodage guidé, baptisé BONG (**B**ag **O**f **NG**ram driven decoding). Nous présenterons par la suite les expériences réalisées en utilisant les systèmes auxiliaires séparément, puis en généralisant la combinaison à plusieurs systèmes auxiliaires.

4.4.1 Principe du BONG

Dans la formulation initiale de DDA, l'hypothèse auxiliaire est considérée comme une séquence de mots. Notre proposition est de relâcher partiellement cette contrainte de séquentialité et de représenter chaque segment de l'hypothèse *auxiliaire* comme un sac de trigrammes. Contrairement à la formulation initiale (équation 4.2), la ré-évaluation linguistique est limitée aux n-grammes communs entre hypothèses primaires et transcriptions auxiliaires. Par ailleurs, l'alignement DTW est remplacé par une simple recherche des n-grammes proposés par le système *primaire* dans le sac de n-grammes correspondants. Cette simplification est raisonnable, car la durée des segments n'excède pas 10 secondes, ce qui correspond, en moyenne, à 20 mots par segments. Ces modifications permettent une accélération du processus de combinaison et rendent l'intégration de plusieurs systèmes auxiliaires simple et efficace [Bougares 2011]. L'architecture du décodage guidé par sac de n-grammes (BONG) est présentée dans la figure 4.2.

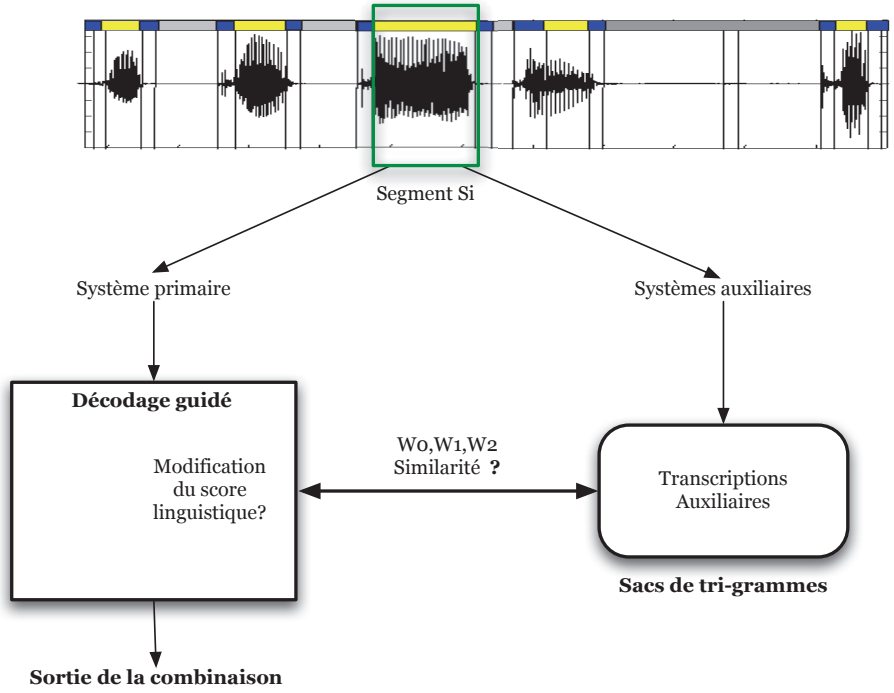


FIGURE 4.2 – Architecture du décodage guidé par sac de trigrammes (BONG).

4.4.2 BONG : utilisation d'un seul système auxiliaire

L'ensemble des expériences qui suivent ont été effectuées en utilisant le décodage guidé par sac de n-grammes (avec $n = 3$). Le système primaire est celui du LIUM et les transcriptions auxiliaires sont fournies par les systèmes IRÈNE (IRISA) et SPEERAL (LIA). Nous utilisons, dans un premier temps, les deux systèmes auxiliaires séparément. Les résultats par radio sont reportés dans le tableau 4.3 pour chaque système combiné avec le système du LIUM, avant et après l'adaptation des modèles acoustiques de chaque système.

Système	F.Info	F.Inter	RFI
LIUM-P1	20,33%	20,93%	25,21%
LIUM-P2	17,92%	19,34%	22,59%
IRÈNE	21,61%	21,96%	26,03%
SPEERAL	21,97%	22,52%	24,95%
BONG-SPEERAL-P1	19,85%	20,17%	23,53%
BONG-SPEERAL-P2	**18,01**%	**19,04**%	**21,17**%
BONG-IRÈNE-P1	19,39%	19,74%	23,16%
BONG-IRÈNE-P2	**17,30**%	**18,47**%	**21,38**%

TABLE 4.3 – Taux d'erreur mot par radio pour la combinaison *BONG* du système du LIUM avec celui du LIA (BONG-SPEERAL) et de l'IRISA (BONG-SPEERAL) avec (P2) et sans (P1) adaptation acoustique.

La combinaison BONG permet une amélioration des résultats par rapport aux systèmes seuls. Le meilleur résultat de combinaison est obtenu avec la transcription du meilleur système auxiliaire IRÈNE (réduction absolue du WER comprise entre 0.62% et 1.21%). Nous présentons également les résultats globaux obtenus sur le même corpus dans le tableau 4.4.

Système	SPEERAL-aux	IRÈNE-aux
SPEERAL	23,06%	–
IRÈNE	–	23,07%
LIUM-P1	22,03%	22,03%
BONG-P1	21,09% (-0,97)	20,66% (-1,37)
LIUM-P2	19,85%	19,85 %
BONG-P2	**19,34%** (-0,51)	**18,96%** (-0,89)

TABLE 4.4 – Taux d'erreur mot global pour la combinaison *BONG* de système du LIUM avec celui du LIA (BONG-SPEERAL) et de l'IRISA (BONG-SPEERAL) avec (P2) et sans (P1) adaptation acoustique.

Le tableau 4.4 montre que la combinaison *BONG* permet d'obtenir une réduction conséquente du taux d'erreur mot, en comparaison avec les résultats de référence du système primaire. La réduction du WER sur l'ensemble des trois heures montre des gains absolus de 0,51% en utilisant la transcription auxiliaire du système SPEERAL et de 0,89% avec celle du système IRÈNE. Il est important de noter aussi que ces résultats sont obtenus en utilisant des transcriptions auxiliaires d'un WER plus élevé que le système primaire (plus de 3% en absolu).

Dans la section suivante, nous proposons d'étendre la combinaison BONG en généralisant la combinaison à plusieurs systèmes auxiliaires.

4.4.3 BONG : généralisation vers n systèmes

La généralisation de la combinaison par décodage guidé est directe ; en cas de présence de plusieurs systèmes, les hypothèses auxiliaires issues de ces systèmes sont groupées dans le même sac de n-grammes à utiliser pendant la ré-évaluation linguistique. Dans le cadre de la combinaison BONG généralisée, nous avons repris la même expérience en fusionnant, pour chaque segment, les transcriptions de deux systèmes dans le même sac de trigrammes. Les résultats obtenus sont reportés dans le tableau 4.5.

Systèmes	WER Global
SPEERAL	23,06 %
IRÈNE	23,07 %
LIUM-P1	22,03%
LIUM-P2	19,85 %
BONG-IRÈNE-SPEERAL-P1	**20,48**% (-1,55)
BONG-IRÈNE-SPEERAL-P1-P2	**18,77**% (-1,08)

TABLE 4.5 – Taux d'erreur mot des systèmes auxiliaires, de la première (LIUM-P1) et de la deuxième passe (LIUM-P2) du système primaire, puis de la combinaison *BONG* avec plusieurs systèmes auxiliaires appliquée en première (BONG-IRÈNE-SPEERAL-P1) et en deuxième passe (BONG-IRÈNE-SPEERAL-P1-P2).

Nous observons une amélioration avec la combinaison BONG généralisée par à rapport à l'utilisation d'un seul système auxiliaire. La réduction est de l'ordre de 0,2% en absolu par rapport au meilleur résultat obtenu avec la combinaison avec un seul système auxiliaire. Au final, la combinaison BONG permet une réduction en absolu de 1,08% de WER par rapport au meilleur système seul.

Afin de compléter notre analyse de la combinaison BONG nous avons comparé les résultats obtenus avec la combinaison ROVER de trois systèmes de base (*ROVER-3*). De plus, nous avons intégré la sortie de la combinaison BONG dans le schéma du ROVER. Les résultats sont reportés dans le tableau 4.6. Nous observons que la combinaison BONG permet une amélioration marginale (-0,14%) par rapport à la combinaison ROVER des trois systèmes. Cependant, l'ajout de la sortie de la combinaison BONG au schéma de la combinaison *ROVER* (*BONG+ROVER*) permet un gain additionnel permettant d'obtenir un WER final de **18,66%**.

Systèmes	WER Globale
ROVER-3	18.91%
BONG-IRÈNE-SPEERAL	**18.77**%
BONG+ROVER	**18,66**%

TABLE 4.6 – Taux d'erreur mot selon le schéma de la combinaison : le ROVER entre les trois systèmes (ROVER-3), la combinaison avec les deux systèmes auxiliaires (BONG-IRÈNE-SPEERAL) et l'intégration de la sortie de la combinaison BONG dans ROVER (BONG+ROVER).

Afin de vérifier si l'amélioration du WER obtenue par rapport à un ROVER de base est significative, nous avons réalisé le test de significativité statistique « *MAPSSWE : Matched Pairs Sentence-Segment Word Error* » fourni par l'institut *NIST* dans l'outil *sc_stats* [Pallett 1990]. Ce test se focalise sur les segments où les systèmes comparés proposent des hypothèses différentes. Ce test indique que l'amélioration obtenue est statistiquement significative avec un niveau $\mathbf{p<0{,}001}$.

4.4.4 BONG : analyse de la combinaison

Étant donné que la combinaison BONG permet d'exploiter les transcriptions auxiliaires au niveau des segments, nous proposons une évaluation plus fine de l'efficacité de la méthode. Nous divisons les segments à décoder en plusieurs classes selon leur taux d'erreur initial, et nous évaluons le gain obtenu avec la combinaison pour chacune des classes. Cette méthode d'évaluation permet de mesurer le comportement de la méthode de combinaison en fonction du taux d'erreur mot du système primaire.

Les segments ont été découpés en 12 classes selon le taux d'erreur initial du système primaire. La première classe (classe «0») contient tous les segments transcrits parfaitement par le système. La classe «0-10» contient tous les segments ayant un WER appartenant à l'intervalle $]0, 10]$, «10-20» pour les segments avec un WER compris entre $]10, 20]$ et ainsi de suite. La dernière classe contient les segments avec un WER supérieur à 100%. Par la suite nous calculons l'impact de la combinaison sur chaque classe en utilisant la formule suivante :

$$\frac{(WER_{baseline} - WER_{combination})\ \#word_c}{\#word_t} \tag{4.3}$$

avec $WER_{baseline}$ le WER initial du système primaire, $WER_{combination}$ le WER résultat de la combinaison, $\#word_c$ le nombre de mots dans la classe et $\#word_t$ le nombre total de mots.

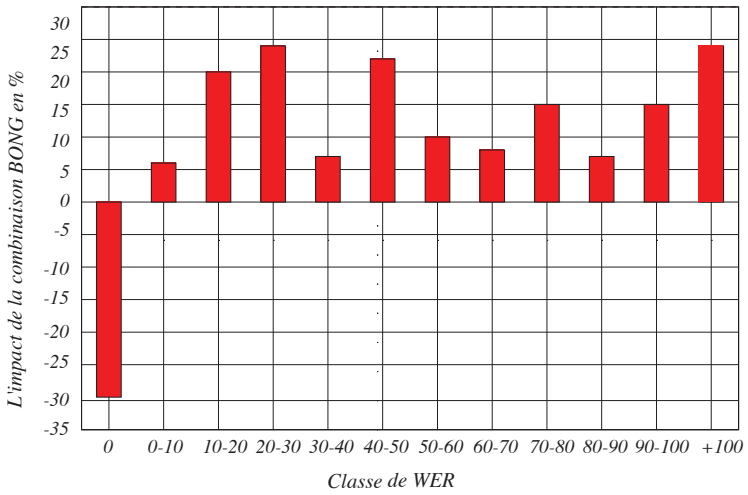


FIGURE 4.3 – Impact de la combinaison BONG par classe de taux d'erreur.

La figure 4.3 montre que l'impact de la combinaison dépend de la performance initiale du système primaire. En effet, la transcription auxiliaire a un effet négatif lorsque la sortie du système primaire est parfaite (classe d'erreur «0»). En revanche, et pour toutes les autres classes, l'impact de la combinaison est positif.

À l'issue de cette étude et dans le but d'éviter les dégradations causées par la combinaison sur certains segments, nous avons cherché à caractériser les segments dans le but d'ajouter un module de décision permettant de

prédire l'effet de la combinaison et de l'appliquer uniquement lorsque cela permet d'améliorer la sortie du système primaire.

Dans le but de distinguer les segments nous avons calculé, pour chaque segment, la moyenne et la variance des mesures de confiance des mots reconnus par le système primaire. Nous avons ensuite classé les segments en deux groupes : les segments dont la transcription est améliorée après la combinaison et les segments où la sortie est dégradée après la combinaison. La figure 4.4 montre l'impact de la combinaison en fonction de la moyenne des mesures de confiance calculée segment par segment.

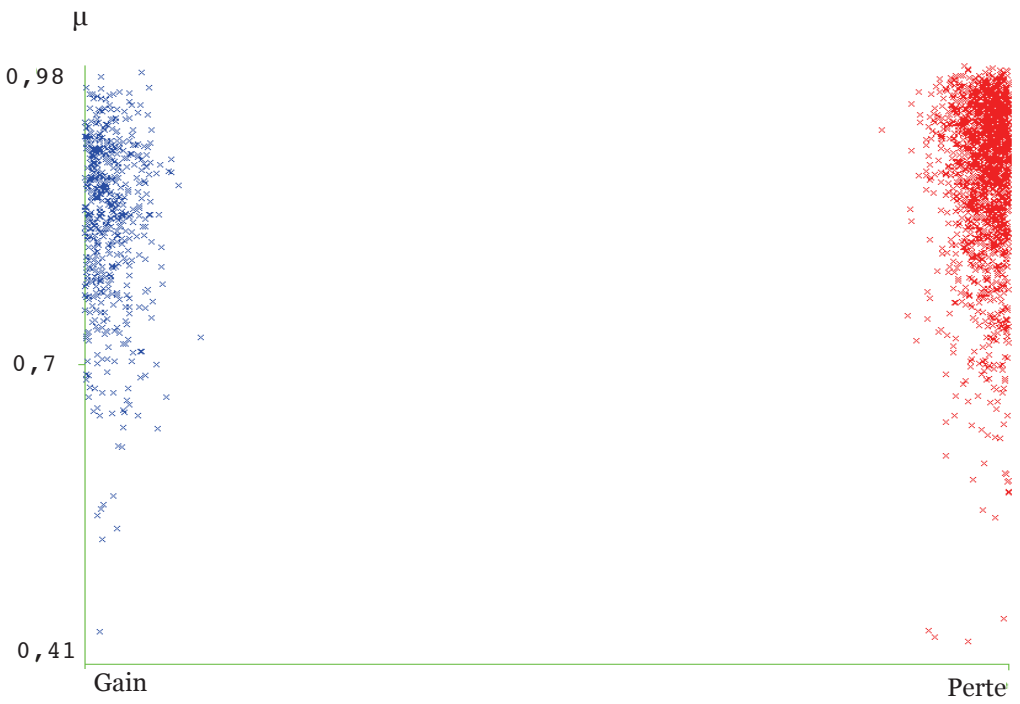


FIGURE 4.4 – Impact de la combinaison BONG en fonction de la moyenne des mesures de confiance des segments.

Les résultats de cette analyse montrent que la moyenne des mesures de confiance ne permet pas de distinguer les segments améliorés par la combinaison (les segments à gauche sur la figure 4.4) des autres segments (à droite sur la figure 4.4). La même analyse a été réalisée en utilisant la variance des mesures de confiance et cela ne permet pas non plus de différencier les segments selon l'impact de la combinaison BONG.

4.4.5 Combinaison BONG et traduction automatique de la parole

La campagne d'évaluation IWSLT (International Workshop on Spoken Language Translation) [Federico 2011] vise l'évaluation des performances des systèmes de traitement automatique des langues. L'évaluation s'articule autour de plusieurs tâches de traitement automatique de différentes langues.

Une nouvelle tâche a été introduite dans la campagne IWSLT de l'année 2011. Cette tâche, nommée SLT (Spoken Langage Translation), consiste à construire une chaîne complète en partant d'un signal audio dans une langue A vers sa transcription traduite dans une langue B (de l'anglais vers le français dans IWSLT 2011). Autrement dit, la tâche nécessite la transcription automatique de la parole dans la langue A et la traduction automatique des sorties du système de transcription vers la langue B.

La tâche de reconnaissance automatique de la parole consiste à transcrire automatiquement un discours en langue anglaise couvrant une variété de sujets diffusés sur le web [1]. Le LIUM à développé un système de reconnaissance pour l'anglais basé sur le projet CMU Sphinx. Le système développé est inspiré du système développé pour la langue française et détaillé dans la section 2.10.

Cinq laboratoires ont participé à la tâche de transcription. Le taux d'erreur mot des systèmes individuels et du ROVER entre les quatre meilleurs systèmes sont présentées dans le tableau 4.7.

Systèmes	Dev2010	tst2010
Système 0 (FBK)	21,2%	19,7%
Système 1 (KIT)	23,7%	22,3%
Système 2 (LIUM)	19,2%	18,2%
Système 3 (NICT)	28,7%	28,0%
Système 4 (MIT)	17,8%	15,8%
Rover-4-2-0-1	16,2%	14,6%

TABLE 4.7 – Performance des systèmes participants à IWSLT 2011 et de la combinaison *ROVER* entre les quatre meilleurs systèmes.

1. http://www.ted.com/

Ces résultats montrent que le système du LIUM a obtenu des bons résultats, malgré la faible quantité de données utilisée pour l'apprentissage des modèles (118 heures d'enregistrement audio).

Dans le cadre de la campagne d'évaluation, les sorties de la transcription automatique sont utilisées comme entrée dans le système de traduction. Par conséquent, les sorties de tous les systèmes ont été rendues accessibles par les organisateurs. Nous avons utilisé ces transcriptions pour appliquer la combinaison BONG avec le système du LIUM. Différents schémas de combinaison ont été testés. Le meilleur résultat est obtenu lorsque le système du LIUM est guidé par les transcriptions auxiliaires issues de deux meilleurs systèmes auxiliaires (systèmes 4 et 0 dans le tableau 4.7). Les résultats de la combinaison sont rapportés dans le tableau 4.8.

Systèmes	Dev2010	tst2010
LIUM	19.2%	18.2%
BONG_{4-0}	18.3%	17.0%
Rover-4-2-0-1	16.2%	14.6%
Rover-(4-BONG-2-0-1)	**15.7**%	**14.1**%

TABLE 4.8 – Taux d'erreur mot du système du LIUM, de la combinaison BONG ainsi qu'un ROVER sans (Rover-4-2-0-1) et avec (Rover-(4-BONG-2-0-1)) la sortie de la combinaison BONG.

Ces résultats montrent une amélioration significative du WER avec la combinaison des systèmes par rapport au système primaire seul (le système du LIUM). La combinaison avec les systèmes 0 et 4 améliore le WER de 0,9 points en absolu sur le dev2010 et de 1,2 points en absolu sur le tst2010, mais les résultats du système MIT (système 5) restent meilleurs que la combinaison BONG. Au final, nous avons intégré la sortie de la combinaison BONG dans le schéma du ROVER initial, permettant ainsi une amélioration du résultat final de 0,5 points en absolu sur le dev2010 et le tst2010 par rapport au ROVER initial.

Finalement, le LIUM a participé à la tâche de traduction automatique des sorties du système de transcription. L'utilisation de la sortie du ROVER avec l'intégration du résultat de la combinaison BONG améliore le résultat du système de traduction. Une description détaillée des différents travaux présentés dans la campagne d'évaluation IWSLT est présentée dans [Rousseau 2011].

4.5 Conclusion

Dans ce chapitre, nous avons étudié la combinaison par décodage guidé. Cette méthode exploite les transcriptions fournies par un système auxiliaire pour guider le processus de décodage d'un système primaire. Cette méthode de combinaison a été d'abord implémentée et testée dans le système du LIUM. Les résultats expérimentaux montrent que cette approche de combinaison intégrée obtient des gains significatifs même lorsque le système primaire est guidé par des transcriptions auxiliaires issues d'un système auxiliaire moins performant.

Nous avons proposé par la suite la combinaison par sacs de n-grammes (BONG), pour améliorer et simplifier la combinaison par décodage guidé. Dans cette amélioration, les transcriptions auxiliaires sont représentées par des sacs des 3-grammes et utilisées par le système primaire pour la ré-évaluation linguistique des hypothèses de transcription durant le décodage. Les résultats expérimentaux montrent que la combinaison BONG apporte des gains significatifs par rapport à la combinaison DDA de base comme implémentée dans le système du LIUM.

Nous avons également présenté la généralisation de la combinaison BONG à n systèmes auxiliaires avec une amélioration par rapport à l'utilisation d'un seul système auxiliaire. De plus, la sortie de la combinaison BONG permet de fournir une nouvelle sortie complémentaire qui, intégrée dans le schéma de la combinaison ROVER, permet une amélioration additionnelle de la transcription finale. L'ensemble de ces travaux a été présenté dans [Bougares 2011].

La méthode de combinaison BONG a été utilisée pendant la campagne d'évaluation IWSLT 2011 pour la reconnaissance des discours en langue anglaise. L'intégration de la sortie de la combinaison ROVER permet un gain en WER absolu de 0,5 point par rapport au ROVER initial et par conséquent, l'amélioration de la qualité de la sortie de système de traduction.

Chapitre 5

Attelage de SRAP hétérogènes à latence réduite

Sommaire

5.1 Introduction

L'efficacité des SRAP actuels a été considérablement améliorée et les performances obtenues, même si elles sont encore loin d'égaler les performances humaines, permettent l'intégration de fonctionnalités de reconnaissance vocale dans des applications commerciales variées.

L'amélioration des performances et de la robustesse de tels systèmes est, entre autres, due à l'utilisation d'architectures multi-passes associées à des méthodes de combinaison adaptées. Cependant, l'utilisation de telles architectures couplées à des méthodes de combinaison de systèmes, appliquées généralement *a posteriori*, augmente substantiellement le temps de traitement nécessaire pour effectuer la reconnaissance.

Afin d'obtenir une transcription avec un temps de réponse acceptable, les systèmes sont généralement limités à une seule passe sans utilisation de méthodes de combinaison. Cette limitation dégrade de façon conséquente la qualité des transcriptions. Dans l'objectif de pallier cette dégradation de performance, tout en gardant une latence réduite, nous proposons un cadre d'attelage définit par une combinaison de systèmes mono-passe hétérogènes durant leurs processus de décodage.

Dans ce chapitre nous présentons, dans un premier temps, notre contexte d'étude de la transcription à latence réduite. Nous introduisons par la suite, la combinaison de systèmes à latence réduite en présentant les différents modèles théoriques de collaboration et les méthodes de combinaison adaptées pour la collaboration. Enfin, nous présentons les premières expériences réalisées et les résultats obtenus.

5.2 Latence des SRAP : définition

La latence d'un système de reconnaissance à *large vocabulaire* est généralement définie par le temps de traitement entre l'acquisition du signal sonore et la production d'une hypothèse de transcription [Seward 2003]. Par conséquent, les systèmes temps-réel ne garantissent pas une transcription à faible latence. En effet, lorsque le système comporte une phase de pré-traitement (la segmentation par exemple) qui fonctionne sur la totalité du signal de la parole, la latence est au moins égale à la durée du signal traité, quelle que soit la rapidité de la transcription.

La latence d'un SRAP n'est pas liée uniquement à la rapidité du processus de décodage, mais englobe aussi le temps de traitement consommé par le processus de segmentation et de paramétrisation, plus les éventuels pré-traitements et post-traitements réalisés. La latence est affectée par des facteurs de différentes natures comme la taille des fenêtres glissantes utilisées pour la paramétrisation acoustique et pour la normalisation de l'ensemble des paramètres obtenus [Saraclar 2002].

La latence est également fonction de la taille du vocabulaire et de la complexité des modèles linguistiques et acoustiques. En effet, lorsque le vocabulaire utilisé est abondant, la construction et l'exploration de l'espace de recherche sont plus coûteuses en temps de traitement. De plus, le coût de l'estimation des vraisemblances acoustiques est directement dépendant du nombre de composantes gaussiennes à évaluer dans les modèles utilisés. En ce qui concerne le temps de calcul des scores linguistiques, le coût dépend principalement de l'ordre du modèle de langage utilisé : un modèle d'ordre supérieur augmente la taille de l'historique à prendre en compte, multiplie le nombre d'hypothèses évaluées durant le processus du décodage et retarde la stabilisation de l'hypothèse de reconnaissance.

La transcription à latence réduite a toujours été étudiée dans un cadre de transcription mono-système [Gu 2008, Chong 2010]. Contrairement aux études précédentes, nous nous intéressons à la latence introduite par l'utilisation de plusieurs systèmes de transcription dans un cadre de combinaison de systèmes. Dans cette étude, nous définissons la latence par le délai écoulé entre le début du processus de la transcription et la proposition d'une hypothèse, qu'elle soit correcte ou incorrecte.

5.3 Attelage des systèmes mono-passe

Bien que les méthodes de combinaison de systèmes permettent d'augmenter les performances des SRAP, elles nécessitent un temps traitement conséquent induit par le temps d'attente nécessaire avant la combinaison (le temps d'attente est généralement égal au temps de traitement du système le moins rapide). Ce temps d'attente requis avant la combinaison rend les méthodes de combinaisons *a posteriori* inadéquates dans un contexte d'utilisation à latence réduite. Dans le cadre de notre travail, nous nous intéressons à la latence induite par les méthodes de combinaison. Nous proposons différents modèles théoriques pour l'attelage de plusieurs systèmes mono-passe. Nous désignons par attelage de systèmes, la collaboration de plusieurs systèmes durant leurs processus de décodage, en utilisant des méthodes de combinaison adaptées, permettant la construction collaborative d'une hypothèse finale améliorée avec un impact limité sur la latence de la transcription.

5.3.1 Modèles théoriques

Lorsque plusieurs systèmes de transcription sont amenés à coopérer, il est important de considérer les éventuels problèmes liés à la communication et la synchronisation entre les différents systèmes. Le déroulement de l'attelage comme nous l'avons défini plus haut (section 5.3), nécessite un échange d'informations durant les processus de décodage de différents systèmes utilisés. Cet échange, permettra de construire, en collaboration, l'hypothèse de reconnaissance finale tout en gardant une latence réduite. Nous présentons dans cette section différents modèles théoriques de l'architecture de collaboration.

5.3.1.1 Architecture de collaboration symétrique

Dans une architecture *symétrique* chaque système est à la fois primaire et auxiliaire. Cette architecture ne privilégie aucun système, permettant ainsi à chacun d'entre eux de profiter librement des informations partagées par l'ensemble des systèmes impliqués. L'architecture de collaboration *symétrique* est présentée dans la figure 5.1.

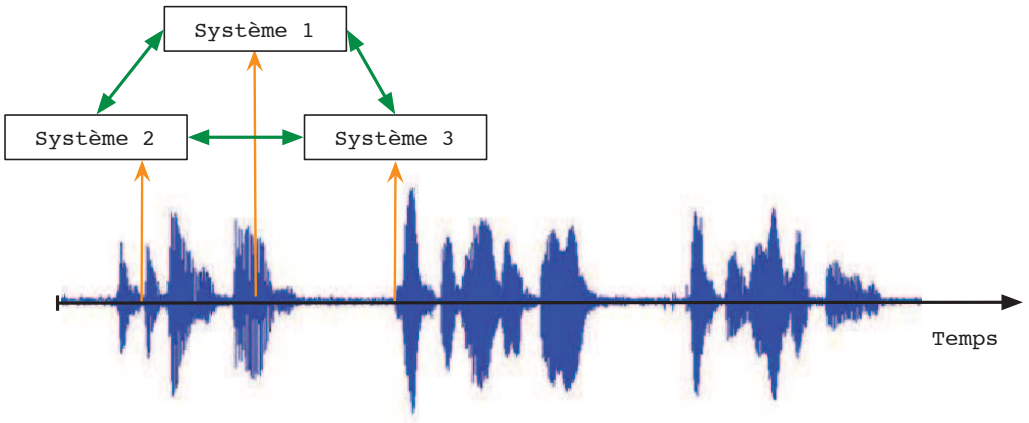


FIGURE 5.1 – Architecture de collaboration *symétrique* entre différents systèmes de transcription mono-passe.

Dans le modèle de collaboration *symétrique* présenté dans la figure 5.1, tous les systèmes commencent simultanément leur processus de décodage. Chaque système partage, à son rythme, ses transcriptions partielles avec les autres systèmes. L'inconvénient majeur de ce type d'architecture provient du fait que la communication n'est pas synchronisée et que la gestion de la collaboration est décentralisée. En effet, chaque système définit indépendamment sa fréquence d'échange d'informations, compliquant la mise en place et la gestion de la communication de l'ensemble.

5.3.1.2 Architecture de collaboration *asymétrique*

Dans une architecture de collaboration *asymétrique*, les systèmes fonctionnent dans une relation *Maître-Esclave* où les rôles sont affectés *a priori*. Par ailleurs, le système *Maître* est choisi à l'avance et la latence est paramétrable selon l'application visée. L'architecture de collaboration *asymétrique* est présentée dans la figure 5.2.

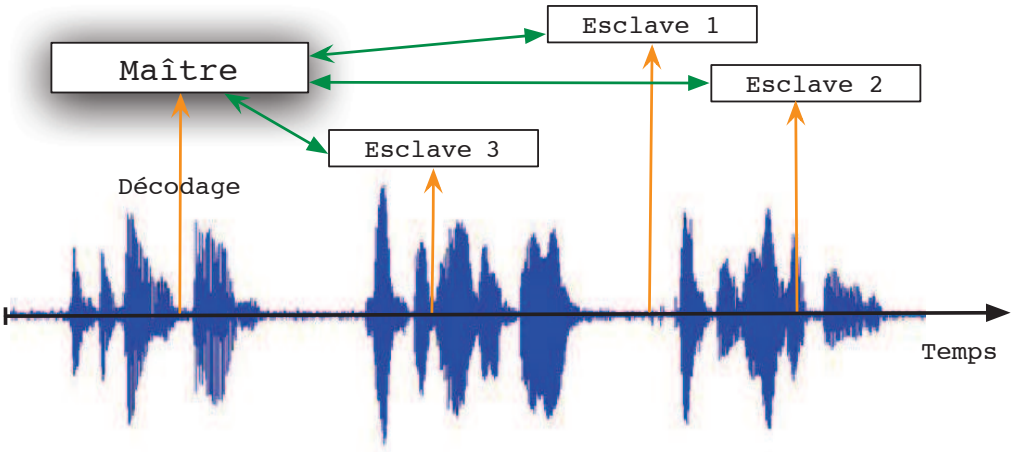


FIGURE 5.2 – Architecture de communication *asymétrique* entre différents systèmes de transcription mono-passe.

Dans cette architecture de collaboration, le système dit *Maître* contrôle, à tout instant, le processus de décodage de l'ensemble des systèmes *esclaves*. Le système *Maître* contrôle l'ensemble des systèmes *esclaves* en demandant, par exemple, qu'un esclave réalise une tâche spécifique ou partage des informations relatives à un segment donné. Il est possible aussi d'augmenter la granularité de contrôle et de doter le système *Maître* de la possibilité de modifier les modèles de connaissance et les jeux de paramètres utilisés par les systèmes *esclaves* (modèles acoustiques et linguistiques, poids du modèle de langage, heurestiques d'élagage...).

Dans une version simplifiée, l'architecture *asymétrique* consiste en une collaboration où le système *Maître* est le seul système qui exploite les informations partagées par l'ensemble des systèmes *esclaves* impliqués, sans avoir besoin de leur donner des ordres ou de modifier leurs paramètres fixés *a priori*.

Dans ce cas de figure, les systèmes auxiliaires démarrent leur processus de transcription en avance de i trames (i étant la latence définie *a priori*) et partagent des informations qui seront utilisées par le système primaire. L'ensemble d'informations partagé est par la suite récupéré par le système primaire (qui commence son décodage avec un retard de i trames) et intégré dans son processus de décodage afin d'appliquer la combinaison.

L'architecture *asymétrique* simplifiée offre un cadre de collaboration contrôlé que nous avons choisi pour implémenter et tester la faisabilité de l'attelage. En effet, la gestion de l'ensemble d'informations, partagées par les systèmes utilisés, est centralisée. L'implémentation de cette architecture est détaillée dans la section suivante.

5.3.2 Exemple d'implémentation

L'attelage de plusieurs SRAP nécessite l'implémentation d'interfaces de communication entre les différents systèmes. L'utilisation d'un intergiciel (*middleware*) semble être adaptée pour implémenter l'architecture de collaboration. En effet, les intergiciels offrent des services de haut niveau liés aux besoins de communication entre applications tout en masquant la complexité des échanges inter-applications.

Parmi les intergiciels les plus employés ces dernières années, on peut citer les intergiciels client/serveur, orientés communication, basés sur la norme CORBA [OMG. 2004]. CORBA est un ensemble de spécifications et de recommandations rédigées par un groupe de travail nommé OMG (Object Management Group) qui permet à des systèmes hétérogènes distribués de communiquer *via* un ensemble d'outils standardisés et un langage IDL (*Interface Definition Language*). Le langage IDL est un langage purement descriptif qui permet de définir les échanges et les interactions entre les clients et les serveurs en faisant abstraction des langages de programmation. Ce langage a été utilisé pour développer chacun des programmes communiquant.

Afin de tester la faisabilité de notre approche, nous avons développé l'architecture de collaboration en utilisant une implémentation gratuite de l'architecture CORBA [1] permettant une intégration simple de différents systèmes de transcriptions dont le code source était à notre disposition.

1. http://omniorb.sourceforge.net/

Comme présenté dans la figure 5.3, nous avons mis en place un serveur permettant d'héberger et de distribuer les transcriptions de différents systèmes.

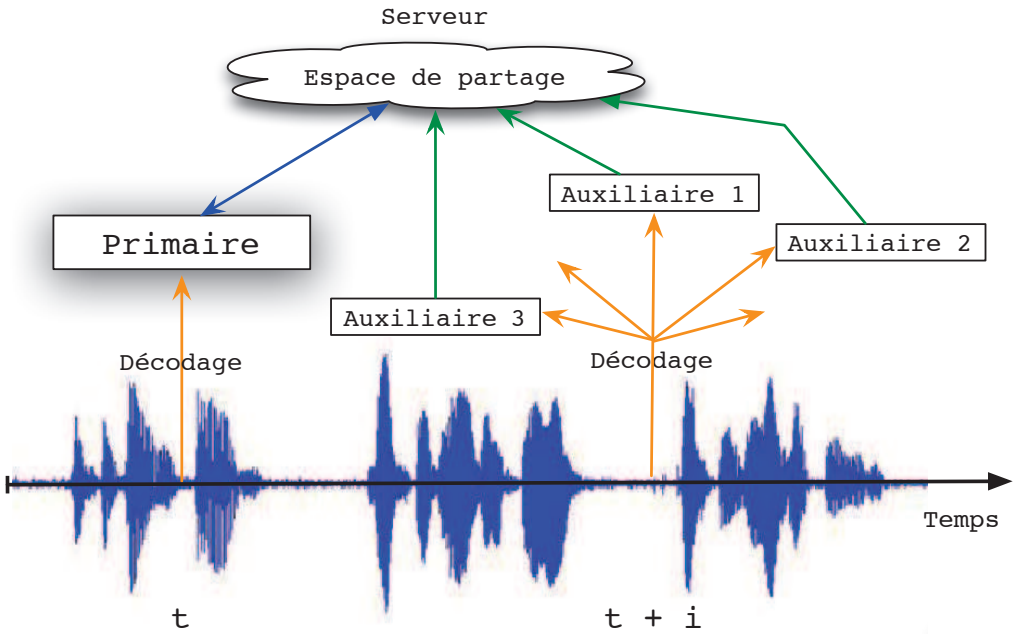


FIGURE 5.3 – Architecture de communication *asymétrique* entre différents systèmes de transcription mono-passe.

Dans la figure 5.3, l'échange d'informations entre les différents systèmes est effectué *via* un processus de communication avec le serveur. Afin de doter les différents systèmes de la capacité de communication avec le serveur nous avons développé des interfaces de communication intégrées à chaque système de transcription utilisé.

La figure 5.4 présente un scénario d'échange d'informations dans une architecture de type *asymétrique* en utilisant l'intergiciel CORBA. Les différents systèmes de transcription impliqués sont disposés autour d'un serveur qui permet de véhiculer les hypothèses de transcription entre systèmes. Une description plus technique de l'architecture CORBA et de l'implémentation de la collaboration est présentée dans l'annexe B.

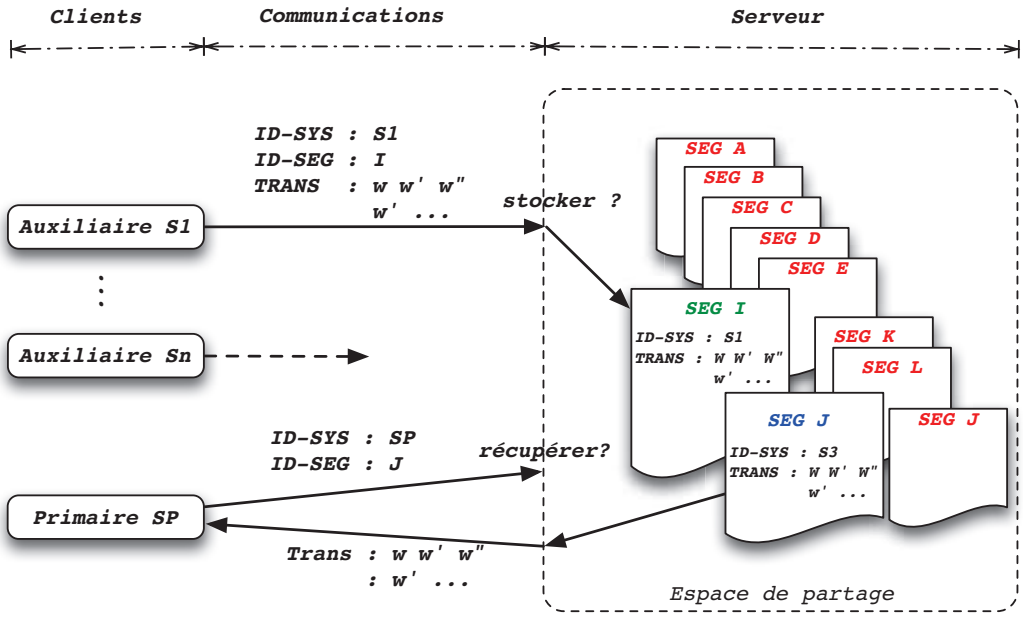


FIGURE 5.4 – Scénario de collaboration entre différents systèmes de transcription mono-passe dans une implémentation client/serveur CORBA.

Dans le scénario présenté, le partage des informations entre les différents systèmes est effectué *via* un processus d'écriture sur un espace partagé et hébergé par le serveur. En l'occurrence, afin de partager des informations sur le segment I, le système auxiliaire $S1$ les envoie au serveur qui s'occupera de leurs stockages au bon endroit (avec l'ensemble d'informations propre au segment I). Pour récupérer l'ensemble des informations concernant un segment donné, le système primaire (système SP dans la figure 5.4) envoie une requête au serveur de partage. La requête est composée des informations concernant l'identifiant du système demandeur de service et de l'identifiant du segment concerné (J en l'occurrence). Le serveur renvoie les transcriptions auxiliaires partagées par les autres systèmes du segment en question. Le serveur peut renvoyer aussi une réponse vide si aucun système n'a partagé de transcriptions pour le segment concerné.

Le fonctionnement de cette formalisation de l'attelage a été testé et vérifié en simulation. Pour la simulation nous avons remplacé les systèmes auxiliaires par des clients *légers* qui envoient les transcriptions auxiliaires mises à leur disposition. Le système primaire récupère et intègre les transcriptions partagées dans son processus de décodage.

5.4 Méthodes de combinaison adaptées pour l'attelage

En plus de la définition de modèles de communication entre systèmes, l'attelage de différents systèmes nécessite la mise en place de méthodes de combinaison adaptées à un contexte de transcription à latence réduite. Ces méthodes doivent permettre l'intégration et l'exploitation de l'ensemble des informations échangées.

5.4.1 BONG : systèmes mono-passe et transcriptions partielles

Les performances des SRAP à l'état de l'art résultent d'une modélisation robuste du signal de la parole et d'algorithmes d'apprentissage et de décodage plus efficaces. Néanmoins, les performances ont aussi été améliorées par l'utilisation d'architectures multi-passes et de méthodes de combinaison. Les techniques de combinaison de systèmes sont apparues avec l'utilisation des architectures de transcription multi-passes [Ostendorf 1991] et continuent à être utilisées avec ce genre d'architecture [Gales 2006, Lamel 2006].

Le temps de latence inhérent au décodage multi-passes rend ce type d'architecture moins attrayant pour répondre à des cadres applicatifs contraignants qui exigent des délais de traitement courts (les systèmes interactifs homme/machine par exemple). Dans ce cas, les architectures multi-passes sont généralement remplacées par des systèmes mono-passe sans méthode de combinaison. Cependant, la limitation à une seule passe de décodage sans combinaison de systèmes fait perdre les améliorations apportées par ces deux techniques. Dans l'objectif d'améliorer la qualité de la transcription et de maintenir une latence acceptable, nous proposons de combiner des systèmes mono-passe. La latence de la combinaison de plusieurs systèmes de transcription mono-passe est égale à la latence du système le plus en retard parmi l'ensemble des systèmes combinés.

Dans l'objectif de limiter la latence induite par les méthodes combinaisons classiques, nous proposons de combiner plusieurs systèmes fonctionnant en parallèle à l'aide d'une méthode de combinaison à latence réduite. La combinaison *BONG*, présentée dans le chapitre 4, est une combinaison qui opère durant le processus de décodage pouvant être adaptée pour une transcription à latence réduite. La combinaison *BONG* de base opère durant le processus de décodage et améliore la sortie du système primaire en utilisant

les transcriptions obtenues par des systèmes auxiliaires. Cette méthode de combinaison, présenté dans la figure 5.5, n'est pas adaptée pour une transcription à latence réduite puisque les transcriptions auxiliaires sont obtenues après un processus de décodage complet d'un ou plusieurs systèmes auxiliaires. Par conséquent, la latence est maximale et égale au temps de décodage du système auxiliaire le plus long (L_aux) dans la figure 5.5).

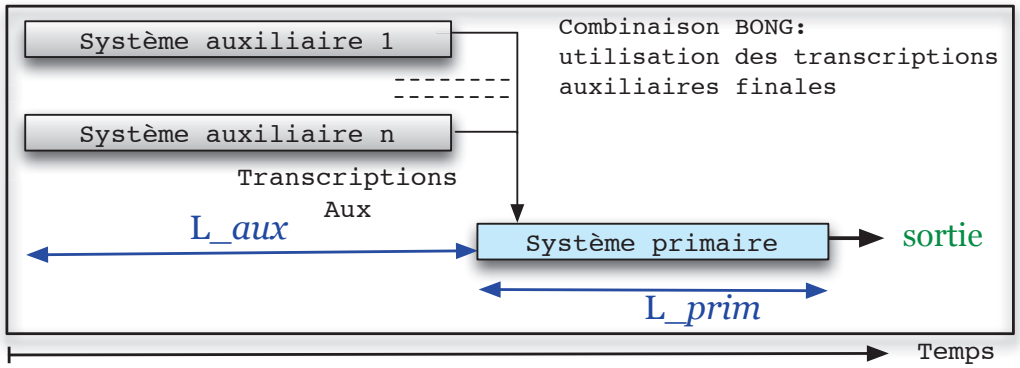


FIGURE 5.5 – Combinaison *BONG* classique des SRAP mono-passe.

Afin de réduire cette latence (L_aux), nous proposons d'utiliser les transcriptions partielles durant les processus de décodage des systèmes auxiliaires [Bougares 2012]. Contrairement aux méthodes classiques où la combinaison est généralement réalisée *a posteriori*, cette méthode consiste à exploiter les transcriptions partielles générées durant le processus de décodage de l'ensemble des systèmes auxiliaires utilisés.

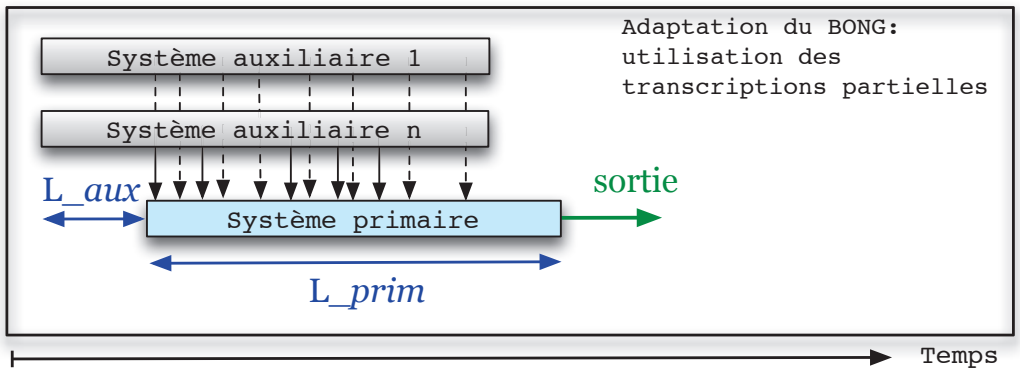


FIGURE 5.6 – Combinaison *BONG* à latence réduite des SRAP mono-passe.

La figure 5.6 présente le schéma de combinaison que nous proposons. L'utilisation des transcriptions partielles permet d'adapter la combinaison *BONG* à une transcription à latence réduite et d'épargner le temps d'attente nécessaire pour obtenir les transcriptions auxiliaires finales avec la combinaison *BONG* de base. Les systèmes auxiliaires fournissent leurs hypothèses partielles durant le processus de décodage, ces transcriptions auxiliaires partielles sont ensuite utilisées pour alimenter les sacs de n-grammes utilisés dans l'approche *BONG*.

5.4.2 Combinaison *ROVER* modifiée (LoROV)

La combinaison *ROVER* a été utilisée dans de précédentes expériences comme méthode de combinaison complémentaire à la combinaison *BONG* [Bougares 2011] : la sortie de la combinaison *BONG* a été intégrée avec les sorties de systèmes auxiliaires pour effectuer une combinaison *ROVER* comme présenté dans la figure 5.7.

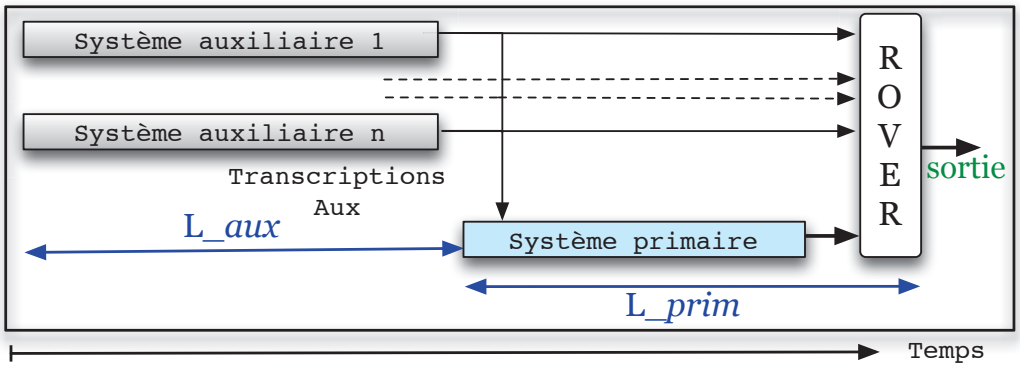


FIGURE 5.7 – Combinaisons *BONG* et *ROVER* utilisés conjointement.

L'utilisation conjointe de deux méthodes de combinaison (*BONG* et *ROVER*) a permis d'obtenir une amélioration additionnelle de la qualité de transcription. Cependant, la combinaison *ROVER* classique introduit une latence non négligeable, non souhaitée dans le cadre de la transcription à latence réduite. Dans l'objectif de bénéficier de l'amélioration obtenue avec *ROVER* tout en réduisant la latence induite par le processus de décodage du système primaire et la combinaison *ROVER* (la latence *L_prim*), nous proposons d'effectuer la combinaison *ROVER* durant le processus de décodage du système primaire comme présenté dans la figure 5.8.

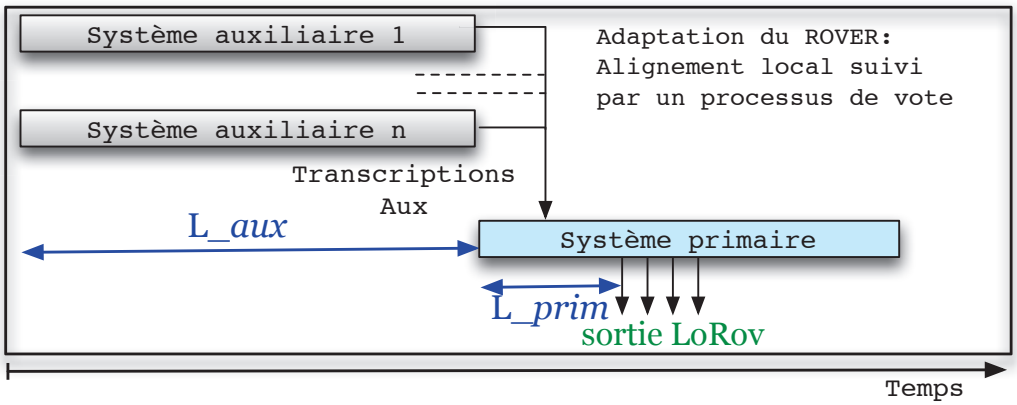


FIGURE 5.8 – Combinaison *ROVER* modifiée appliquée durant le décodage.

Contrairement au *ROVER* classique, où le processus d'alignement est réalisé sur la sortie finale de l'ensemble des systèmes combinés, nous proposons d'utiliser un *ROVER* local (LoROV pour Local *ROVER*). Le processus d'alignement du ROVER classique a été remplacé par un alignement temporel (temps de début et de fin de chaque mot) par rapport à la meilleure hypothèse partielle locale obtenue durant le chaînage arrière (*Backtracking*) du système primaire. Le processus de vote est ensuite réalisé et les hypothèses finales sont sélectionnées en se basant sur le critère de fréquence d'apparition des mots.

5.5 Cadre expérimental

Dans l'ensemble de nos expérimentations, nous utilisons quatre systèmes de reconnaissance de la parole mono-passe basés sur trois moteurs de reconnaissance différents. Le système principal est celui du LIUM, décrit en détail dans 4.3.1.1. En plus des transcriptions auxiliaires en provenance du système SPEERAL (voir section 4.3.1.2), nous avons utilisé deux versions du système de transcription RWTH, décrit dans la section suivante.

5.5.1 Système de reconnaissance du RWTH : RASR

Le système de transcription automatique de la parole RASR (RWTH ASR) a été développé par le groupe RWTH à l'université de Aachen (Allemagne). RASR est gratuitement téléchargeable[2] sous une licence dérivée de la Licence Publique Q (QPL). Le système est basé sur un

2. http://www-i6.informatik.rwth-aachen.de/rwth-asr/

décodeur *Beam search*, une modélisation n-gramme du langage et des modèles de Markov cachés contextuels (triphone inter- et intra-mots) à états partagés avec une paramétrisation $MFCC$ à 15 coefficients, auxquels s'ajoutent, l'énergie et leurs dérivées premières. Contrairement aux autres systèmes, les modèles acoustiques sont indépendants du genre et de la bande. Une description plus détaillée du système est présente dans [Lööf 2007]. Le lexique et le modèle de langage sont ceux utilisés dans le système du LIUM.

Nous utilisons deux variantes du système RASR en modifiant la paramétrisation acoustique : d'abord, nous utilisons directement les 15 coefficients $MFCC$, ensuite nous concaténons les coefficients de 9 trames consécutives pour capturer l'information sur une fenêtre temporelle à plus long terme. Nous appliquons ensuite une LDA [Haeb-Umbach 1992] afin d'obtenir un vecteur acoustique de 45 coefficients.

5.5.2 Données expérimentales

La combinaison de systèmes donne souvent des meilleurs résultats lorsque les systèmes utilisés ont des performances comparables [Hoffmeister 2006]. De ce fait nous avons choisi d'évaluer notre méthode de combinaison uniquement sur la partie où les systèmes utilisés ont des performances proches. Étant donné que le système RASR, contrairement aux autres systèmes, utilise un seul modèle acoustique indépendant de la bande et du genre appris sur des données d'émissions radiophoniques majoritairement de type studio, l'évaluation est faite sur la partie *STUDIO* du corpus de développement de la campagne d'évaluation ESTER 2. Cette partie est plus proche du corpus d'apprentissage et son utilisation pour l'évaluation réduit l'écart entre le système RASR et les autres systèmes. Le corpus de développement contient initialement 6 heures d'émission radiophonique et la partie *STUDIO*, utilisée pour nos expériences, représente 5 heures.

5.5.3 Performance des systèmes

Les données expérimentales sont initialement découpées en segments de 10 secondes en utilisant le système de segmentation et regroupement en locuteur du LIUM. Ces segments sont par la suite transcrits par les quatre systèmes utilisés. Le taux d'erreur mot de chaque système est reporté dans le tableau 5.1.

Systèmes	Taux d'erreur mot
LIUM	**32,3** %
SPEERAL	32,8 %
RASR$_{LDA}$	34,1 %
RASR	34,4 %

TABLE 5.1 – Taux d'erreur mot du système primaire (LIUM) et des systèmes auxiliaires (SPEERAL, RASR et RASR$_{LDA}$) sur la partie STUDIO du corpus dev ESTER 2

En nous basant sur ces résultats, nous avons assigné à chaque système son rôle : le système le plus performant (système du LIUM en l'occurrence) sera le système primaire et les autres seront utilisés comme systèmes auxiliaires.

5.5.4 Résultats

Les transcriptions fournies par les systèmes auxiliaires ont été utilisées, dans un premier temps, pour guider la première passe du système de transcription du LIUM dans une combinaison *BONG*. La transcription obtenue est par la suite intégrée dans le schéma de combinaison *ROVER*. Les résultats sont présentés dans le tableau 5.2.

Systèmes	Taux d'erreur mot
LIUM	32,3 %
BONG$_{ALL}$: LIUM{SPEERAL + RASR$_{LDA}$ + RASR}	28,6 %
ROVER : LIUM-SPEERAL-RASR$_{LDA}$-RASR	28,3 %
ROVER$_{ALL}$: BONG$_{ALL}$-SPEERAL-RASR$_{LDA}$-RASR	**27,4** %

TABLE 5.2 – Taux d'erreur mot du système primaire (LIUM), de la combinaison BONG intégrée dans ce système en utilisant les sorties des systèmes auxiliaires (BONG$_{ALL}$) et d'un ROVER de quatre systèmes (LIUM, SPEERAL, RASR et RASR$_{LDA}$). Le ROVER$_{ALL}$ représente une combinaison *ROVER* en remplaçant le système du LIUM par la sortie du BONG dans le schéma ROVER de base.

Les résultats rapportés dans le tableau 5.2 montrent une amélioration significative du taux d'erreur mot avec une réduction absolue de 4 points avec la combinaison *ROVER* et 3,7 points avec BONG (BONG$_{ALL}$ dans le tableau 5.2). La sortie de la combinaison *BONG* est obtenue en utilisant les transcriptions finales fournies par les systèmes auxiliaires segment par segment. Par conséquent, la latence est égale à la taille des segments traités (nous utilisons des segments de 10s issu d'un processus de segmentation automatique). Cette

sortie est par la suite, intégrée dans une combinaison *ROVER* pour permettre une amélioration finale de 4,9 points dans l'absolu. Nous obtenons finalement un taux d'erreur mot de **27,4%** en utilisant uniquement des systèmes à passe unique.

5.5.4.1 Combinaison *BONG* à latence réduite

La combinaison *BONG* a également été testée en utilisant les hypothèses partielles issues des systèmes auxiliaires. La figure 5.9 présente la variation du WER obtenue avec la combinaison *BONG* en fonction d'une latence maximale. Dans ces expériences, la latence d'un système auxiliaire est définie comme le nombre de trames existant entre la trame en cours de lecture et la dernière trame du dernier mot de l'hypothèse partielle que ce système est capable d'émettre. Par nature, les hypothèses auxiliaires partielles sont de moins bonne qualité que les hypothèses finales et c'est l'impact de cette détérioration due à la contrainte de latence que nous souhaitons mesurer. Il s'agit ici d'expériences qui se situent dans le cas extrême où la latence varie entre 1 trame et la latence maximale. Par exemple, pour une latence maximale égale à 100 trames (une seconde), lorsque le système primaire décode une trame comprise entre la trame 100 et 199 incluses, nous utilisons uniquement les transcriptions auxiliaires fournies de la trame 0 à la trame 200.

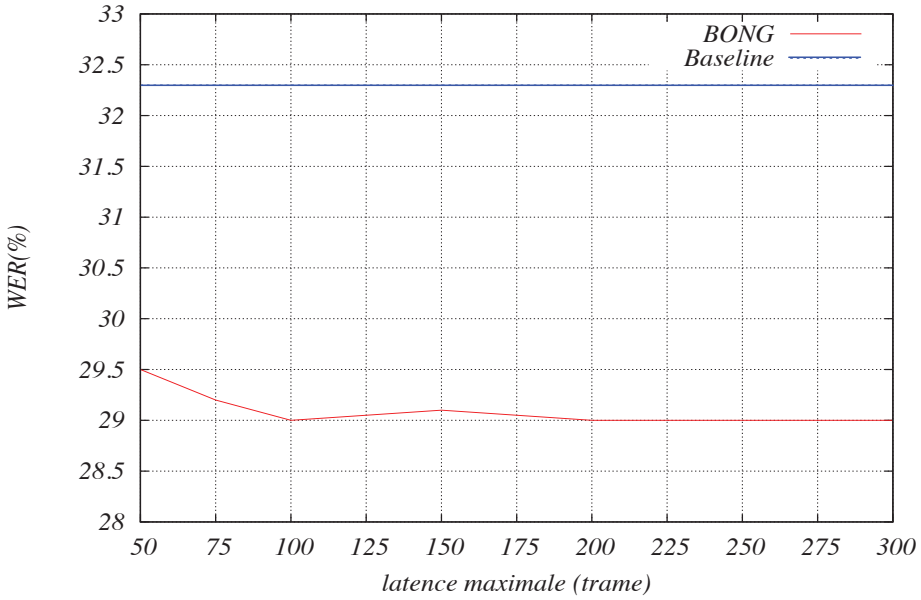


FIGURE 5.9 – WER de la combinaison *BONG* à latence réduite en utilisant les transcriptions partielles fournies par les systèmes auxiliaires RASR, RASR$_{LDA}$ et SPEERAL.

La figure 5.9 montre une amélioration significative du WER de l'ordre de 3 points du WER en comparaison avec les résultats de référence du système primaire (le système baseline sur la figure 5.9). Cette amélioration est obtenue en utilisant des transcriptions auxiliaires avec une latence maximale égale à une seconde. Ces résultats représentent une première expérience qui montre que la combinaison *BONG* est aussi utilisable dans un cadre d'attelage à latence contrainte (une seconde au lieu de 10 secondes en moyenne avec la combinaison *BONG* sans latence réduite). Cela dit, les résultats pourraient être améliorés en remplaçant la latence maximale par une latence fixe, ce qui correspondrait à l'utilisation des 260 premières trames dans l'exemple présenté précédemment (100 trames à compter de la trame en cours de décodage).

5.5.4.2 Combinaison *ROVER* à latence réduite

Différentes expériences ont été menées afin de tester le schéma de combinaison ROVER à latence réduite présenté dans la figure 5.8. Le tableau 5.3 présente les résultats de la combinaison *LoROV* où l'alignement est effectué localement avec une fréquence d'alignement égale à 3 secondes. Le choix de l'hypothèse de transcription finale est réalisé avec un vote local dans l'objectif de réduire la latence introduite par la combinaison *ROVER* classique.

Systèmes	Taux d'erreur mot
LoROV(3s) : LIUM-SPEERAL-RASR$_{LDA}$-RASR	30,8 %
BONG$_{ALL}$: LIUM{SPEERAL + RASR$_{LDA}$ + RASR}	28,6 %
LoROV(3s)-BONG$_{ALL}$	**27,8** %

TABLE 5.3 – Taux d'erreur mot de la combinaison *LoROV(3s)* avec une latence de 3 secondes entre les 4 systèmes de transcription et de la combinaison *BONG* sans (BONG$_{ALL}$) et avec (LoROV(3s)-BONG$_{ALL}$) la combinaison *LoROV(3s)*.

Nous avons testé, dans un premier temps, la combinaison *LoROV (LoROV : LIUM-SPEERAL-RASR$_{LDA}$-RASR dans le tableau 5.3)* en utilisant l'ensemble des systèmes de transcription construits. La combinaison *LoROV* est réalisée en utilisant un processus d'alignement sous optimal basé sur les informations temporelles. Par conséquent, l'amélioration obtenue est moins importante que celle obtenue avec le *ROVER* classique (tableau 5.2) où l'alignement est effectuée en calculant la distance minimale d'édition entre les hypothèses finales de différents système de transcription. Toutefois, la combinaison *LoROV* réduit le taux d'erreur mot initial de l'ordre de 2 points (le WER est réduit de 4 points dans l'absolu avec le *ROVER* classique).

La combinaison *LoROV* a été testée par la suite conjointement avec la combinaison *BONG* (LoROV-BONG$_{ALL}$ dans le tableau 5.3). Le résultat obtenu montre une amélioration par rapport à la combinaison *BONG* seule. L'amélioration absolue est de l'ordre de 0,8% point de WER (27,8% de WER avec LoROV-BONG$_{ALL}$ par rapport à 28,6% avec BONG$_{ALL}$). Les résultats présentés dans les tableaux 5.2 et 5.3 montrent une dégradation de 0,4 point de taux d'erreur mot lorsque la combinaison BONG est couplée avec le *LoROV* à la place de l'utilisation d'un *ROVER* classique. Toutefois l'utilisation du *LoROV* avec la combinaison *BONG* permet d'obtenir une amélioration de l'ordre de 0,5% point dans l'absolu par rapport à l'utilisation du *ROVER* classique seul, tout en étant plus performant en terme de latence.

5.6 Évaluation des systèmes de reconnaissance

Afin de promouvoir le développement des technologies vocales en langue française, l'Association Francophone de la Communication Parlée (AFCP) organise fréquemment des campagnes d'évaluation des systèmes du traitement automatique de la parole de langue française. L'objectif premier est de faire progresser les technologies de traitement automatique de la parole par la production de ressources et l'évaluation de systèmes.

Les campagnes permettent d'établir une référence sur le niveau de performance actuel de chacune des composantes d'un système de transcription. Dans cette partie, nous décrivons notre participation à la dernière campagne d'évaluation organisée en 2011 : ETAPE.

5.6.1 Campagne ETAPE

La campagne d'évaluation ETAPE[3] s'inscrit dans la continuité des campagnes ESTER tout en élargissant les enjeux scientifiques, en particulier, à la parole spontanée, la parole superposée et à la diversité des contenus. ETAPE reprend en grande partie les tâches existantes lors de la campagne d'évaluation ESTER 2. Ces tâches sont organisées autour de trois thèmes, à savoir la segmentation (S), la transcription (T) et l'extraction d'information (E). Pour chaque tâche, les participants peuvent soumettre plusieurs systèmes et identifier le système principal qui servira à établir le classement officiel des systèmes en compétition.

3. Évaluations en Traitement Automatique de la Parole

Le LIUM a participé, entre autres, à la tâche Transcription en utilisant deux systèmes : un système principal multi-passes, détaillé dans l'annexe A, et un système contraste mono-passe détaillé dans cette section.

5.6.2 Systèmes de transcription

Le système contraste, développé durant la campagne ETAPE, est une combinaison de plusieurs systèmes mono-passe : le système du LIUM, le système SPEERAL du LIA et la version LDA du système RASR de RWTH. Pendant la campagne ETAPE, nous avons focalisé nos efforts sur la création d'un système multi-passes RASR$_{LDA}$ (détaillé dans l'annexe A). C'est la raison pour laquelle nous n'avons pas développé la variante sans LDA.

5.6.3 Données expérimentales

Les données d'évaluation utilisées pendant la campagne d'évaluation ETAPE sont constituées de deux corpus d'émissions radiophoniques. Les corpus de développement et de test contiennent respectivement 8h25 et 9h30 d'enregistrement audio. Le tableau 5.4 présente une description générale des corpus utilisés, classés par source.

Source	dev	test
BFM	1h00	1h00
LCP	3h20	3h20
TV8 Mont Blanc	1h05	2h10
France Inter	3h00	3h00
Total	8h25	9h30

Table 5.4 – Données d'évaluation ETAPE.

5.6.4 Résultats

Initialement, nous avons utilisé séparément les trois systèmes pour transcrire les données expérimentales. Les résultats obtenus sont rapportés dans le tableau 5.5.

Systèmes	Dev	Test
LIUM	31,9 %	33,74 %
SPEERAL	39,4 %	39,28 %
RASR$_{LDA}$	30,4 %	33,08 %

Table 5.5 – Taux d'erreur mot de la première passe du système du LIUM, du système SPEERAL et du système RASR$_{LDA}$ sur le Dev et le Test d'ETAPE.

Les systèmes sont par la suite combinés en utilisant différents schémas de combinaison. Le tableau 5.6 montre les résultats obtenus en utilisant plusieurs systèmes mono-passe.

Systèmes	Dev	Test
ROVER-3	29,1%	30,86 %
BONG	29,8%	32,57 %
ROVER-BONG	**27,9**%	**30,38** %

TABLE 5.6 – Taux d'erreur mot du *ROVER* entre les 3 systèmes de base (ROVER-3), de la combinaison *BONG* avec SPEERAL et RASR$_{LDA}$ comme auxiliaires (*BONG*) et d'un *ROVER* entre le résultat de la combinaison BONG et les systèmes auxiliaires (ROVER-BONG).

Les systèmes ont été d'abord combinés avec un *ROVER* basé sur un processus de vote majoritaire (la fréquence d'apparition du mot). La combinaison *BONG* est par la suite effectuée en intégrant les transcriptions des systèmes SPEERAL et RASR$_{LDA}$ dans le processus de décodage du système du LIUM.

Les meilleurs résultats sont obtenus lorsque la sortie de la combinaison BONG est intégrée dans une combinaison *ROVER* avec l'ensemble de systèmes auxiliaires : une amélioration du WER de l'ordre de **2,5** points dans l'absolu sur le corpus du Dev et **2,7** points dans l'absolu sur le Test. La combinaison *LoROV* a été mise de coté dans ces expériences puisque la latence n'est pas évaluée dans la campagne ETAPE.

5.7 Conclusion et perspectives sur l'attelage

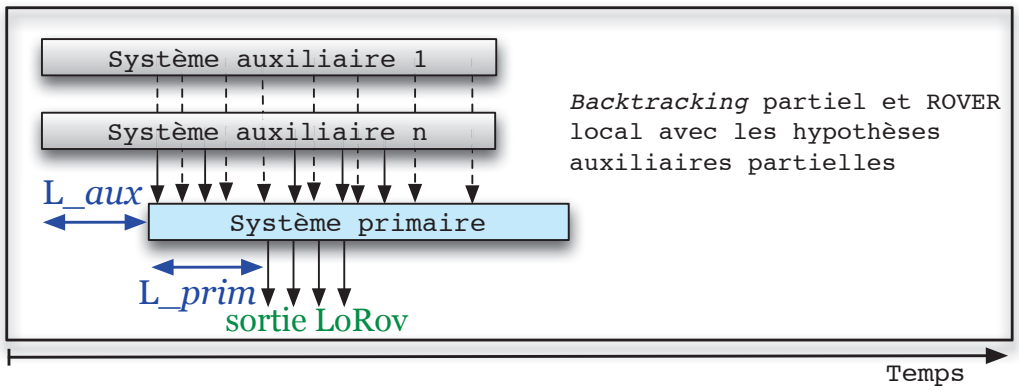
Dans ce chapitre, nous avons proposé une architecture d'attelage de systèmes à latence réduite. Dans l'objectif de réduire la latence du processus de reconnaissance, nous remplaçons l'architecture multi-passes par une combinaison de plusieurs systèmes mono-passe fonctionnant en parallèle. Nous avons présenté, dans un premier temps, une étude de différents modèles théoriques de l'attelage de systèmes hétérogènes suivie par un exemple d'implémentation d'un modèle théorique simplifié basé sur l'architecture client/serveur *CORBA*. Cette architecture permet de mettre en place un serveur de gestion de partage d'informations entre les différents systèmes.

Nous avons proposé par la suite une adaptation de la combinaison *BONG* dans l'objectif de réduire la latence de la transcription induite par les systèmes auxiliaires. L'adaptation est réalisée en utilisant les transcriptions partielles issues de plusieurs systèmes auxiliaires. Nous avons également proposé une technique permettant d'adapter la combinaison *ROVER* classique. Afin de réduire la latence due au processus de décodage du système primaire plus la combinaison *ROVER*, nous utilisons un processus d'alignement local, entre les hypothèses partielles fournies par les systèmes auxiliaires et l'hypothèse partielle du système primaire, suivi par un processus vote basé sur la fréquence de mots.

Les deux méthodes de combinaison adaptées ont été par la suite testées séparément dans un cadre expérimental bien défini. L'ensemble des expériences réalisées valide dans un premier temps l'amélioration de la qualité de la transcription obtenue après la combinaison *BONG* de plusieurs systèmes mono-passe. Les résultats obtenus montrent aussi que la combinaison *BONG* est utilisable dans un cadre d'attelage à latence contrainte avec des transcriptions auxiliaires partielles de moins bonne qualité. En effet, l'utilisation des transcriptions partielles a permis la réduction de la latence induite par le temps de décodage des systèmes auxiliaires par un facteur de 10 (une seconde d'attente au lieu de 10 secondes).

La combinaison *ROVER* adapté à la latence réduite a été aussi expérimentée et l'utilisation d'un ROVER local à une fréquence de trois secondes permet d'améliorer la qualité de la transcription finale.

Enfin, les deux méthodes de combinaison à latence réduite proposées sont complémentaires et pourront sans doute être combinées ensemble comme présenté dans la figure suivante.



Chapitre 6

Conclusion et perspectives

Sommaire

6.1 Conclusion

Le travail de thèse présenté dans ce manuscrit s'inscrit dans le cadre du projet ASH (Attelage de systèmes Hétérogènes), mettant l'accent sur les méthodes de combinaison des systèmes de reconnaissance automatique de la parole. Les SRAP s'appuient généralement sur cinq modules : la paramétrisation acoustique, le lexique, les modèles acoustiques et linguistiques et le décodeur. La diversité des techniques de mise en œuvre de chaque module a été exploitée pour construire différents systèmes. La disponibilité et la diversité des SRAP actuels est à l'origine du développement de plusieurs méthodes de combinaison, appliquées majoritairement *a posteriori*, pour profiter des points forts de chaque système, dans le but d'améliorer la qualité de la transcription finale.

Ce travail de thèse est au cœur des contributions du projet ASH. Ce projet de recherche s'intéresse en particulier au développement d'un cadre d'attelage permettant la construction collaborative de l'hypothèse de reconnaissance finale. L'idée générale consiste à faire collaborer différents systèmes de transcription durant leur processus de décodage. La collaboration est réalisée *via* un processus d'échange d'informations afin d'améliorer la qualité de la transcription et de réduire la latence induite par les méthodes de combinaison traditionnelles.

Dans la première partie de ce manuscrit, nous nous sommes intéressés aux méthodes de combinaison appliquées durant le processus de décodage. Lors de nos travaux, nous avons étudié puis amélioré la méthode de combinaison par décodage guidé (*DDA : Driven Decoding Algorithm*). Dans la deuxième partie de cette thèse, nous avons adapté et intégré notre nouvelle méthode de combinaison dans le cadre de l'attelage de systèmes que nous avons proposé.

Combinaison de systèmes durant le décodage

La combinaison de systèmes par décodage guidé permet de guider l'exploration du graphe de recherche d'un système primaire *via* une réévaluation à la volée des scores linguistiques. Le guidage est réalisé en utilisant des transcriptions issues de plusieurs systèmes auxiliaires.

Lors de notre étude, cette méthode de combinaison a tout d'abord été intégrée dans le système de transcription du LIUM, puis expérimentée et analysée en utilisant un puis plusieurs systèmes auxiliaires. Enfin, nous avons proposé une amélioration de la combinaison par décodage guidé. L'amélioration proposée, baptisée combinaison *BONG*, permet l'accélération de la combinaison, l'amélioration de la qualité de transcription et l'augmentation de la robustesse du système lorsqu'un seul système auxiliaire est utilisé. La combinaison *BONG* est facile à mettre en œuvre et directement généralisable : l'utilisation de plusieurs systèmes auxiliaires permet l'intégration d'un ensemble d'informations multiples et variées et améliore la qualité de la transcription du système primaire même lorsque les transcriptions auxiliaires sont fournies par des SRAP moins performants.

La combinaison *BONG* produit également une sortie complémentaire, que nous avons intégrée dans une combinaison *ROVER* avec l'ensemble des systèmes auxiliaires : une réduction complémentaire du WER a été obtenue. La combinaison *BONG* a également été testée dans le cadre de la traduction automatique de la parole durant la campagne d'évaluation IWSLT 2011. La combinaison *BONG* a permis, entre autres, l'amélioration de la qualité de la traduction automatique statistique de la parole.

Attelage de systèmes mono-passe à latence réduite

Nous nous sommes intéressés, dans la seconde partie de nos travaux, à la mise en place d'une méthode de combinaison de SRAP qui autorise une combinaison à latence réduite, au contraire des méthodes existantes jusqu'alors. Nous avons proposé, dans un premier temps, différents modèles théoriques permettant l'attelage de plusieurs SRAP mono-passe. Nous avons ensuite présenté un exemple d'implémentation fondé sur l'utilisation d'une architecture de communication *client/serveur* permettant le partage d'informations entre les différents SRAP impliqués dans l'attelage. Le partage d'informations est réalisé en ajoutant des interfaces de communication pour échanger avec un espace partagé hébergé sur un serveur. Celui-ci offre différents services, permettant ainsi le stockage des informations partagées

par les systèmes auxiliaires, et la récupération de ces informations par le système primaire.

Après avoir déterminé l'architecture de l'attelage, nous nous sommes intéressés aux méthodes de combinaison exploitables dans le cadre d'un attelage de systèmes à latence réduite. Deux méthodes de combinaison complémentaires ont été présentées :

- la première méthode est une adaptation de la combinaison *BONG*. L'adaptation consiste à utiliser les hypothèses partielles fournies par les systèmes auxiliaires durant leur processus de décodage. Cette adaptation permet de réduire la latence induite par la combinaison *BONG* initiale, qui est effectuée à partir des transcriptions finales proposées par les systèmes auxiliaires ;

- la deuxième méthode est une modification de la combinaison *ROVER*. Nous avons remplacé le processus d'alignement de la combinaison *ROVER* classique par un alignement local intégré au processus de décodage du système primaire. L'alignement est réalisé en utilisant les informations temporelles afin de projeter les transcriptions auxiliaires sur la meilleure hypothèse partielle du système primaire.

Les deux méthodes de combinaison à latence réduite proposées sont complémentaires. En effet, leur utilisation conjointe permet une réduction du WER de l'ordre de **4,7%** dans l'absolu (27,8% avec les deux méthodes de combinaison par rapport à 32,3% sans combinaison *cf.* tableau 5.3). Enfin, à notre connaissance ce travail de recherche représente la première étude sur la réduction de la latence des méthodes de combinaison.

6.2 Perspectives

Le cadre de collaboration des SRAP présenté dans cette thèse permet l'amélioration des transcriptions finales en utilisant plusieurs systèmes mono-passe avec une latence réduite. À partir de ce cadre de collaboration, plusieurs perspectives d'amélioration sont envisagées. D'abord, la méthode de combinaison *BONG* peut bénéficier de plusieurs améliorations. Dans un premier temps, il serait possible de créer des systèmes auxiliaires conçus pour être complémentaires. Cela permettrait de construire des sacs de n-grammes contenant plus probablement les bonnes hypothèses.

La souplesse de la combinaison *BONG* peut être exploitée pour augmenter le nombre de systèmes auxiliaires dans un cadre de décodage massif. De plus, les n-grammes partagés pourraient être pondérés en fonction du système qui les a émis par exemple.

Une autre extension possible de la combinaison *BONG* serait d'utiliser d'autres types d'informations. La combinaison, basée pour le moment sur la mise à jour de score linguistique, pourrait s'étendre à d'autres niveaux : par exemple en favorisant les phonèmes proposés par plusieurs systèmes auxiliaires. Les informations phonétiques proposées par un systèmes auxiliaire pourraient être également exploitées pour guider le processus d'élagage. En effet, cela permettrait d'accélérer le processus de décodage du système primaire en réduisant le nombre d'évaluations acoustiques. Ceci étant dit, l'utilisation des informations phonétiques nécessite une réflexion sur le processus de phonétisation de l'ensemble des systèmes impliqués.

L'utilisation d'anti-systèmes, proposant des transcriptions peu probables, mais représentatives des erreurs les plus fréquentes dans le contexte d'énonciation, est aussi envisageable. Elle permettrait de pénaliser certaines hypothèses proposées par ces anti-systèmes. L'utilisation conjointe des systèmes et des anti-systèmes auxiliaires dans la combinaison *BONG* permettrait d'avoir deux classes d'hypothèses auxiliaires : une classe des hypothèses qu'il faudra favoriser et une autre pour celles qu'il faudra pénaliser.

À plus long terme, la combinaison *BONG* pourrait être étendue dans le cadre de la traduction statistique.

Les travaux sur la combinaison de systèmes à latence réduite doivent être aussi approfondis. Jusqu'à présent, les systèmes partagent uniquement leurs meilleures hypothèses. Il serait possible d'élargir ce partage en utilisant des parties de l'espace de recherche de chaque système auxiliaire.

Enfin, il serait intéressant de comparer l'impact des méthodes de combinaison à latence réduite d'un point de vue applicatif : interprétation sémantique dans un contexte de dialogue oral homme/machine, entités nommées, questions/réponses etc.

ANNEXES

Annexe A

Participation du LIUM à la campagne d'évaluation ETAPE

Comme nous l'avons précisé dans la section 5.6.1, le système primaire du LIUM, présenté durant la campagne d'évaluation ETAPE, est une combinaison de plusieurs systèmes multi-passes. Dans cet annexe, nous présentons, tout d'abord, les systèmes et les méthodes de combinaison utilisées. Nous exposons ensuite les résultats obtenus ainsi que le classement final de l'ensemble des participants à la tâche de transcription automatique de la parole.

A.1 Systèmes de reconnaissance

Deux systèmes de transcription ont été utilisés pour construire le système primaire du LIUM durant la campagne ETAPE : le système du LIUM et le système RASR du RWTH.

A.1.1 Système du LIUM

Le système du LIUM est une amélioration du système détaillé dans la section 2.10. Il a été utilisé durant la campagne d'évaluation ESTER 2. Les données d'apprentissage des modèles acoustiques ont été augmentées par les données distribuées par les organisateurs d'ETAPE (17 heures 30 d'enregistrement audio), par des transcriptions rapides de «podcasts» d'émissions de débat de l'année 2011 (99 heures), plus d'autres enregistrements radiophoniques de l'année 2007-2008 (227 heures) pour un total de 571 heures d'enregistrement. Le processus de transcription a également évolué avec l'ajout d'une passe de réévaluation linguistique en utilisant un modèle de langage 5-grammes. Cette réévaluation linguistique a été insérée entre la quatrième passe (la réévaluation linguistique avec un modèle 4-grammes) et la cinquième passe (décodage de réseaux de confusion) du processus de transcription du système ESTER 2.

A.1.2 Système RASR

Le deuxième système de reconnaissance utilisé pendant ETAPE a été construit en utilisant RASR, la boite à outils distribuée par le laboratoire RWTH. Le système utilise une paramétrisation *MFCC* à 15 coefficients, plus l'énergie et leurs dérivées première, une modélisation linguistique de type n-grammes et une modélisation acoustique contextuelle indépendante du genre et de la bande. Le processus de transcription est composé de 4 passes de décodage séquentielles :

1. La première passe de décodage est réalisée avec un modèle de langage 3-grammes et un modèle acoustique indépendant du genre et de la bande ;
2. la deuxième passe utilise la meilleure sortie de la première passe pour effectuer une adaptation CMLLR ;
3. une adaptation MLLR est appliquée durant la troisième passe en utilisant la meilleure sortie de la passe précédente ;
4. les treillis de la troisième passe sont transformés en réseaux de confusion. Ces réseaux sont, par la suite, décodés pour obtenir la sortie finale.

Une deuxième variante du système RASR a été testée en ajoutant une passe de réévaluation linguistique avec un modèle 4-grammes des graphes obtenus après l'adaptation MLLR. La dernière passe de décodage en consensus est maintenue après la réévaluation linguistique.

A.1.3 Performances des systèmes

Initialement, le corpus de développement de la campagne ETAPE, décrit dans la section 5.6.3, a été transcrit par les deux systèmes : les résultats sont présentés dans les tableaux A.1 (système du LIUM), A.2 (système RASR sans réévaluation 4-grammes) et A.3 (système RASR avec réévaluation 4-grammes).

Les résultats présentés dans les tableaux précédents sont obtenus en se fondant sur les outils de normalisation et d'évaluation distribués par NIST et utilisés durant la campagne ESTER 2. Ces résultats montrent une différence de performance assez importante entre les deux systèmes. La transcription de système du LIUM est bien meilleure que celle obtenue avec RASR (une différence de 5,6 points dans l'absolu).

La différence de performance est liée à plusieurs raisons : la première raison réside sans doute dans la différence de modélisation acoustique utilisée dans

Systèmes	Dev
LIUM-Passe 1	31,9 %
LIUM-adaptation CMLLR	27,2 %
LIUM-réévaluation acoustique	23,6 %
LIUM-réévaluation linguistique 4-G	22,3 %
LIUM-réévaluation linguistique 5-G	22,0 %
LIUM-réseaux de confusion	21,1 %

TABLE A.1 – Taux d'erreur mots du système du LIUM sur le corpus de DEV ETAPE.

Systèmes	Dev
RASR-Passe 1	30,4 %
RASR-adaptation CMLLR	28,6 %
RASR-adaptation MLLR	28,3 %
RASR-réseaux de confusion	27,7 %

TABLE A.2 – Taux d'erreur mots du système RASR sur le corpus de DEV ETAPE.

Systèmes	Dev
RASR-Passe 1	30,4 %
RASR-adaptation CMLLR	28,6 %
RASR-adaptation MLLR	28,3 %
RASR-réévaluation 4-G	27.8 %
RASR-réseaux de confusion	26,7 %

TABLE A.3 – Taux d'erreur mots du système RASR sur le corpus de DEV ETAPE avec la réévaluation linguistique quadri-grammes en quatrième passe.

les deux systèmes. En effet, le système du LIUM, contrairement au système RASR, utilise plusieurs modèles acoustiques dépendant du genre de locuteur (homme/femme) et de la largeur de la bande passante (Studio/Téléphone). La différence de performance peut être aussi expliquée par la différence de méthode de paramétrisation du signal utilisée dans les deux systèmes : la paramétrisation *PLP*, utilisée par le système du LIUM est plus robuste que la paramétrisation *MFCC* appliquée dans RASR [Psutka 2001]. De plus, le système du LIUM, contrairement au système RASR, comporte une passe de réévaluation acoustique (troisième passe du système du LIUM dans le tableau A.1) et une passe de réévaluation linguistique avec un modèle de langage 5-grammes (cinquième passe dans le tableau A.1) qui permettent de réduire le WER de 3,9 points dans l'absolu (3,6 points avec la réévaluation acoustique

et 0,3 points avec la réévaluation linguistique).

A.2 Combinaison de systèmes

La construction de deux systèmes de transcription pour la campagne ETAPE a permis de tester et d'évaluer différentes méthodes de combinaison de systèmes. En effet, les systèmes construits ont été utilisés pour expérimenter certaines méthodes de combinaison et pour vérifier leurs performances. Nous avons testé, entre autres, l'adaptation par transformation linéaire croisée et la réévaluation acoustique croisée :

- l'adaptation par transformation linéaire croisée est réalisée en utilisant la meilleure sortie de la quatrième passe du système du LIUM pour effectuer l'adaptation CMLLR et MLLR du système RASR ;

- la réévaluation acoustique croisée est effectuée par une réévaluation des scores acoustiques des treillis, sortis de la passe 3 du système RASR, en utilisant les modèles acoustiques dépendant du genre et de la bande du système du LIUM. Les résultats obtenus sont présentés dans les tableaux A.4 et A.5.

Systèmes	Dev
LIUM-Passe 4	22,3 %
RASR-adaptation CMLLR	27,9 %
RASR-adaptation MLLR	27,7 %
RASR-réévaluation 4-G	27,3 %
RASR-réseaux de confusion	26,2 %

Table A.4 – Taux d'erreur mots du système RASR en utilisant la sortie de la passe 4 du système du LIUM pour l'adaptation CMLLR et MLLR sur le corpus de DEV ETAPE.

Les résultats de l'adaptation par transformation linéaire croisée sont présentés dans le tableau A.4 : l'adaptation CMLLR, en utilisant la sortie de la passe 4 du LIUM (28,3% du WER), permet une amélioration du WER de l'ordre de **0,7** points dans l'absolu par rapport à l'adaptation CMLLR sur la sortie de la première passe du système RASR (du 28,6 % à 27,9 % du WER). L'adaptation MLLR sur la même sortie (passe 4 du LIUM), permet un gain de **0,6** points dans l'absolu du WER (du 28,3 % à 27,7% du WER)

par rapport à l'adaptation MLLR présentée dans le tableau A.3.

Systèmes	Dev
LIUM-Passe 4	28,3 %
RASR-adaptation CMLLR	27,9 %
RASR-adaptation MLLR	27,7 %
RASR-réévaluation acoustique	25,3 %
RASR-réévaluation 4-G	24,6 %
RASR-réseaux de confusion	24,1 %

TABLE A.5 – Taux d'erreur mots du système RASR en utilisant la sortie de la passe 4 du système du LIUM pour l'adaptation CMLLR et MLLR suivie par une réévaluation acoustique en utilisant les modèles du système du LIUM sur le corpus de DEV ETAPE.

Les résultats de la réévaluation acoustique croisée, appliquée sur les treillis provenant de l'adaptation croisée MLLR, sont présentés dans le tableau A.5. Cette réévaluation permet d'améliorer le WER de l'ordre de **2,1** points dans l'absolu (du 27,7 % à 24,6 % dans le tableau A.5).

Au final, les deux techniques utilisées permettent une amélioration du WER de **2,6** points dans l'absolu pour obtenir une sortie à **24,1** % du WER (la sortie du système RASR est à 26,7% sans la combinaison).

A.3 ETAPE : résultats semi-officiels

Nous présentons dans cette section les résultats des participants dans la campagne d'évaluation ETAPE. Les résultats présentés dans le tableau A.6 ont été distribués par les organisateurs avant la phase d'adjudication sur le corpus de Test d'ETAPE.

Systèmes	Test
LIUM	23.60 %
LIA	36.85 %
LORIA	25.87 %
CRIM	26.03 %

TABLE A.6 – Taux d'erreur mots des participants à la campagne d'évaluation ETAPE sur le corpus de Test ETAPE.

Annexe B

Architecture et implémentation de l'attelage des SRAP

Cette annexe détaille l'implémentation de l'architecture d'attelage présentée auparavant dans le chapitre 5, section 5.3.2. Nous examinons dans un premier temps, d'un point de vue plus technique, l'architecture de construction d'application distribuée CORBA. Nous présentons par la suite les raisons qui ont motivé le choix de cette architecture pour l'implémentation de l'attelage des SRAP. Enfin, nous détaillons l'implémentation de l'interface de communication entre les différents systèmes ainsi que les structures de données utilisées pour l'échange d'informations.

B.1 L'architecture CORBA

CORBA est une architecture client/serveur qui constitue la première initiative majeure dans le domaine des objets distribués. Le cœur de CORBA est constitué par le bus ORB (*Object Request Broker*) qui assure l'automatisation des tâches de communication, mais aussi des tâches de localisation et d'activation d'objets ainsi que la traduction des messages échangés entre systèmes hétérogènes.

L'architecture CORBA, présentée dans la figure B.1, permet à chaque application d'exporter certaines de ses fonctionnalités (appelées *services*) sous la forme d'objets CORBA. Les communications sont fondées sur un mécanisme d'invocation de procédures distantes et requièrent la création d'amorces qui se branchent au bus. Celles-ci permettent l'émission et la réception des messages entre les clients et les serveurs.

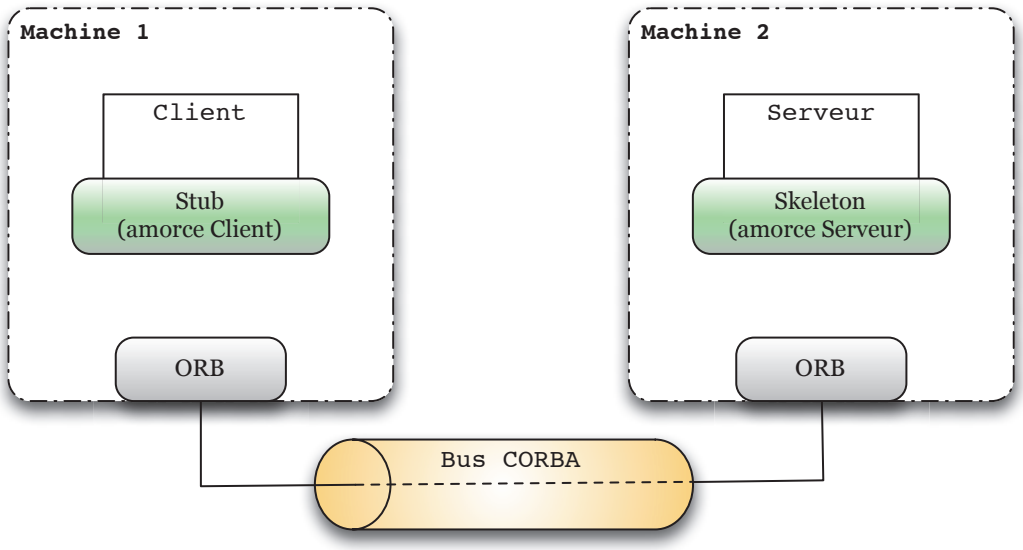


FIGURE B.1 – Modèle de communication Client/Serveur CORBA.

Des nombreuses implémentations du bus CORBA existent à ce jour et beaucoup sont commerciales. Cependant, quelques produits gratuits ont vu le jour et certains sont utilisables, au même titre que des bus commerciaux, pour la conception d'applications robustes et fiables.

B.2 Le langage *IDL*

Le rôle d'un serveur est de mettre un ensemble d'objets à la disposition des clients. Pour pouvoir accéder à ces objets, les clients doivent connaître l'ensemble des méthodes qu'ils peuvent invoquer sur ces objets. Ceci est possible par l'intermédiaire d'un «*contrat*» défini avec le langage de définition d'interface *IDL* (*Interface Definition Language*). L'interface définie par l'*IDL* est constituée d'un ensemble de prototypes de méthodes, permettant de définir une vue fonctionnelle sur un objet ; qui sera exposé par le serveur pour que le client puisse interagir avec lui.

La notion d'interface est similaire à celle de classe utilisée en Programmation Orientée Objet. Une interface met en oeuvre des méthodes et des attributs dont il est nécessaire de définir le type. CORBA étant destiné à créer des applications interopérables, les types de base utilisés sont spécifiques au langage *IDL* (il ne s'agit ni de types Java ni de types C++). Ces types sont ensuite projetés vers les langages dans lesquels sont réalisées les implémentations. Les types *IDL* de base ainsi que les types Java correspondants sont présentés dans le tableau B.1.

Type IDL	Type Java
boolean	boolean
char	char
wchar	char
octet	byte
string	java.lang.String
wstring	java.lang.String
short	short
unsigned short	short
long	int
long long	long
unsigned long	int
float	float
double	double

TABLE B.1 – Types IDL de base.

En plus des types simples (présenté dans le tableau B.1), *IDL* permet d'utiliser des types complexes comme les énumérations, les tableaux, les structures et les séquences. Le type *sequence* permet de déclarer des séquences d'objet quelconque. Les séquences peuvent être bornées ou non-bornées :

- bornée : *sequence<char, 50> nom* ;
- non-bornée : *sequence<char> adresse*.

La définition des méthodes est réalisée en utilisant une syntaxe des spécifications semblable à celle de C++. *IDL* propose 3 types de passages de paramètres pour la description de fonctions prenant un ou plusieurs arguments :

1. **in :** indique que le paramètre est passé au serveur ;
2. **out :** indique que le paramètre est retourné au client ;
3. **inout :** indique que le paramètre est passé au serveur où il peut être modifié et ensuite retourné au client.

Une description détaillée du langage de spécification *IDL* et de correspondance avec les différents langages de programmation impératifs est présentée sur le site Web de l'OMG [1].

1. http://www.omg.org

B.3 CORBA et l'attelage des SRAP

Dans le cadre de collaboration des systèmes de transcription, l'architecture CORBA apporte une souplesse de développement grâce à son approche orientée objet et son indépendance vis-à-vis de l'environnement d'exécution et les langages de programmation. Dans le cadre de notre travail, nous utilisons différents systèmes de transcription appartenant à différents laboratoires de recherche et développés dans différents langages de programmation. L'architecture CORBA permet de faire communiquer ces différents systèmes d'une manière transparente. Elle garantit aussi le développement d'une application distribuée dont les composants collaborent avec :

- Efficacité ;
- fiabilité ;
- transparence ;
- scalability, *i.e.*, une capacité d'évolution importante.

En outre, l'architecture CORBA fournit un cadre de collaboration assez générique permettant l'intégration des nouveaux systèmes de transcription à moindre coût (sans pour autant modifier les composantes déjà développées).

Enfin, la spécification CORBA est toujours en évolution, des services continuent à être ajoutés ou améliorés. Cela est très important pour tenir compte des besoins qui ne cessent de changer de forme en raison des exigences des nouvelles applications.

Le développement d'une application CORBA respecte toujours les mêmes étapes suivantes :

1. La description du contrat de communication avec le langage *IDL* ;
2. l'implantation des divers objets ;
3. l'implantation du serveur ;
4. l'implantation des clients.

Nous présentons dans la section suivante le contrat *IDL*, l'implémentation du serveur de partage et des interfaces de communication. Les interfaces de communication permettent aux systèmes de communiquer avec le serveur pour stocker leurs transcriptions sur l'espace partagé (les systèmes auxiliaires) et pour récupérer les transcriptions partagées (le système primaire).

B.4 Attelage des SRAP : Implémentation

Nous présentons dans cette section l'architecture que nous avons mis en place pour l'attelage de deux systèmes de transcriptions : le système du LIUM basé sur le moteur de reconnaissance sphinx 3 et développé en langage *C* et le système Speeral développé par le laboratoire d'Informatique d'Avignon en langage *C++*. Nous utilisons la même segmentation en locuteur pour les deux systèmes.

Les implémentations de la spécification CORBA gratuites sont de plus en plus performantes. Dans notre implémentation de l'architecture de collaboration nous avons choisi deux implémentations gratuites en fonction de la compatibilité avec le langage de programmation de chaque SRAP ; OmniOrb [2] a été utilisée pour implanter le serveur et l'interface de communication avec le système SPEERAL et Orbit [3] a été retenue pour la communication avec le système du LIUM.

B.4.1 Le contrat *IDL*

Le contrat entre le serveur et les systèmes de transcription s'exprime sous la forme d'un ensemble d'interfaces spécifiées à l'aide du langage *IDL*. Dans le cadre de notre application, le serveur prend en charge la gestion des transcriptions auxiliaires et leur transmission au système primaire.

Durant le processus de la collaboration, les hypothèses transmises par les systèmes auxiliaires correspondent à des n-grammes ($N = 3$) stockés dans une séquence non-bornée de la structure *whyp_t* suivante :

```
struct whyp_t{
        long wrd;  // premier mot de 3-grammes
        long wrd1; // deuxième mot de 3-grammes
        long wrd2; // troisième mot de 3-grammes
        int sys_id; // le système source  (utile pour réaliser des analyses)
        float start_w,start_w1,start_w2;  // trame du début de chaque mot
        float end_w,end_w1,end_w2;   // trame de fin de chaque mot
        float score_w,score_w1,score_w2; // des éventuels scores de confiance
        };

typedef sequence<whyp_t> seqw_t; // séquence non-bornée de tri-grammes
```

2. http://omniorb.sourceforge.net/
3. http://projects.gnome.org/ORBit2/

Lorsqu'un système transmet ses hypothèses, le serveur prend en charge l'enregistrement et l'affectation de ces transcriptions au segment correspondant. Chaque segment est une structure d'éléments contenant :

- L'identifiant du segment ;
- le temps de début du segment ;
- le temps de la fin du segment ;
- la séquence de n-grammes déjà envoyés par des systèmes auxiliaires ;
- un champs additionnel pour indiquer l'état du segment actuel.

Les segments sont stockés dans une séquence non-bornée appelée *seqs_ t*. La structure segment comme définie en langage *IDL* est la suivante :

```
struct segm_t{
     string show; // Identifiant du segment
     float begin;  // Temps de début du segment
     float end;   //  Temps de fin du segment
     seqw_t aux_seq; // Liste de n-grammes partagés pour ce segment
     long state;  // indicateur d'état du segment (vide : 0 ; non vide :1)
     };

typedef sequence <segm_t> seqs_t; // séquence non-bornée de segments
```

Dans la suite nous définissons l'interface *IDL «ash_ collaboration»* où on présente les principales méthodes qui permettent, d'une part, aux systèmes auxiliaires d'envoyer leurs hypothèses et, d'autre part, au système primaire de récupérer les transcriptions mises à disposition par les systèmes auxiliaires.

```
interface ash_collaboration{

void stackhyp (in string show, in float begin,in float end, in whyp_t hyp,
               in long sys_id);

seqw_t get_seg (in string show, in float begin,in float end, in long sys_id,
                in long nb_trig);
   };
```

B.4.2 Code serveur

Le contrat *IDL* est compilé afin de générer les amorces (souche et squelette) requises pour l'établissement de la communication interprocessus. La compilation est réalisée en utilisant *omniidl*, le compilateur *IDL* d'omniORB, en procédant comme suit :

omniidl -bcxx -Wbexample <contrat.idl>

Cette commande traduit les définitions *IDL* dans le langage cible (ici le langage c++) et génère les souches de communication (contratSK.cc et contrat.hh). Avec l'option *-Wbexample* le compilateur génère un exemple d'implémentation permettant d'accélérer la mise en œuvre du serveur. Dans la suite, nous présentons la déclaration de la classe du serveur, la méthode main et la gestion des exceptions système.

- **La classe du serveur :**

```
class ash_collaboration_i: public POA_ash_collaboration {
private:
     seqs_t aux; // séquence de segments auxiliaires

public:
     ash_collaboration_i();
     virtual ~ash_collaboration_i();
     void stackhyp(const char* show, ::CORBA::Float begin, ::CORBA::Float end,
                    const whyp_t& hyp, ::CORBA::Long sys_id);

     seqw_t* get_seg(const char* show, ::CORBA::Float begin, ::CORBA::Float end,
                      ::CORBA::Long  sys_id, ::CORBA::Long nb_trig);

};
```

Les deux méthodes *stackhyp* et *get_ seg* sont les méthodes utilisées par les clients pour stocker et récupérer les transcriptions auxiliaires.

- **La fonction main :** la fonction main s'occupe de la création de l'objet ORB et de la gestion de cet objet.

```
int main(int argc, char** argv)
{
  try {
        // (1)  Initialiser  l'ORB
    CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);
    CORBA::Object_var obj = orb->resolve_initial_references("RootPOA");
        // (2)  Initialiser l'adaptateur d'objets
    PortableServer::POA_var poa = PortableServer::POA::_narrow(obj);
        // (3) Créer les implantations d'objets
    asyn_ash_combination_i* myash_collaboration_i = new ash_collaboration_i();

   PortableServer::ObjectId_var myash_collaboration_iid =
     poa->activate_object(myasyn_ash_combination_i);
       {
         CORBA::Object_var ref = myash_collaboration_i->_this();
         CORBA::String_var sior(orb->object_to_string(ref));
         // (4) Diffuser les références
         // (la chaîne codifiant l'IOR : Interoperable Object Reference)
         // et (5)  Attendre des requêtes venant du bus
         std::cout << "IDL object ash_collaboration_i IOR = '" << (char*)sior
                   << "'" << std::endl;
         ofstream f("ior.txt",ios::out);
         f<<(char*)sior;
         f.close();
        }
    PortableServer::POAManager_var pman = poa->the_POAManager();
    pman->activate();
    orb->run();
    orb->destroy();
  }
  catch(CORBA::TRANSIENT&) {
    cerr << "Caught system exception TRANSIENT -- unable to contact the "
         << "server." << endl;
  }
  catch(CORBA::SystemException& ex) {
    cerr << "Caught a CORBA::" << ex._name() << endl;
  }

  return 0;
}
```

Le serveur suit le scénario suivant :

1. Initialiser le bus CORBA (obtenir l'objet ORB) ;
2. initialiser l'adaptateur d'objets (obtenir POA) ;
3. créer les implantations d'objets ;
4. enregistrer les implantations par l'adaptateur (implicite $C++$) ;
5. diffuser leur référence (afficher une chaîne codifiant l'IOR) ;
6. attendre des requêtes venant du bus.

B.4.3 Code clients

La projection du contrat *IDL* est réalisée avec le même compilateur utilisé précédemment avec le serveur (*omniidl*). Dans le cadre de la collaboration le système Speeral doit être doté de la capacité de communiquer avec le serveur et d'y déposer ses transcriptions.

Nous présentons ensuite la partie utilisée pour établir la communication avec le serveur et les étapes nécessaires pour l'invocation de la méthode *stackhyp*, définie dans le contrat *IDL* et implantée par le serveur.

```
CORBA::ORB_var orb;
whyp_t w;
char ior[1024],segment[512];
orb = CORBA::ORB_init(0,NULL,"omniORB4");
            try{
                ash_collaboration_ptr rstack;
                rstack=ash_collaboration::_narrow(orb->string_to_object(ior));
                rstack->stackhyp(segment,deb,fin,w,sys_id);
                orb->destroy();
                }catch(CORBA::SystemException& f1) {
                        cerr << "Caught CORBA::SystemException." <<endl;}
                 catch(CORBA::Exception&) {
                        cerr << "Caught CORBA::Exception." << endl;}
```

Le client commence par l'initialisation du bus (fonction *ORB_init*) et la création des souches des objets à utiliser (*rstack* dans le code ci-dessus). Le client récupère ensuite les références d'objets (l'IOR distribué par le serveur). Enfin, le client invoque la méthode *stackhyp* qui prend en charge le stockage de ses transcriptions, au bon endroit, dans l'espace de partage hébergé par le serveur.

En ce qui concerne système primaire (le système du LIUM en l'occurrence), l'implémentation consiste à intégrer une fonction permettant l'accès aux transcriptions partagées sur le serveur. Le scénario de communication est identique à celui utilisé avec le système Speeral à l'exception de la méthode invoquée (le système primaire invoque la méthode *get_seg* implantée côté serveur).

La figure B.2 présente l'architecture générale et les sens des communications entre les différents systèmes utilisés.

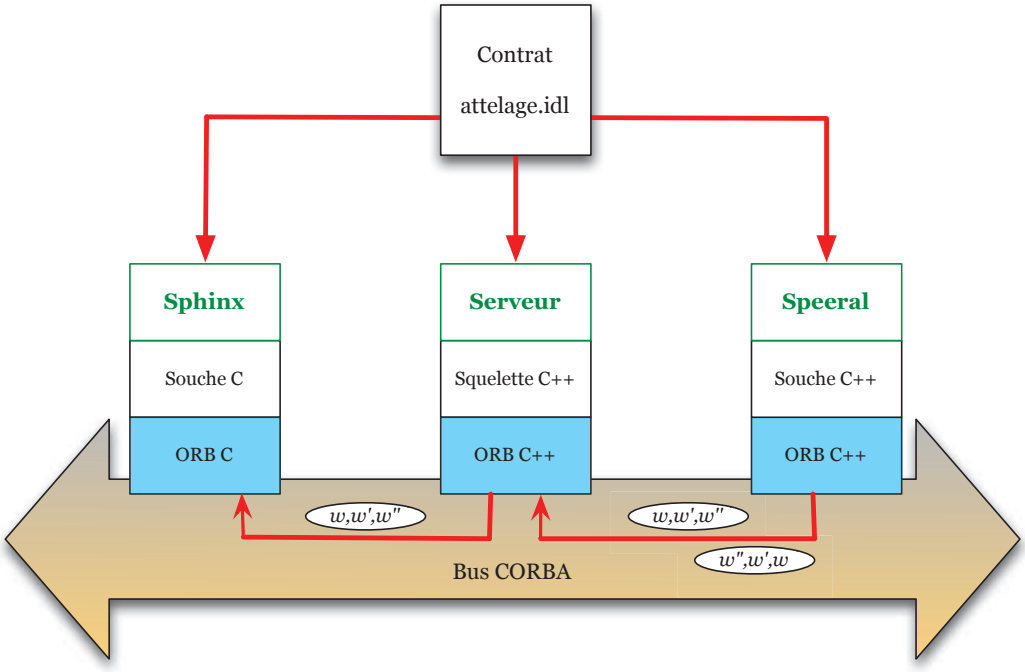


FIGURE B.2 – Collaboration des systèmes de reconnaissance avec l'architecture CORBA.

Table des figures

Liste des tableaux

Acronymes

AFP	Agence France Presse
BAYCOM	BAYesien COMbination
BONG	Bag-Of-NGram driven decoding
CMU	Carnegie Mellon University
CMLLR	Constrained Maximum LikelihoodLinear Regression
CORBA	Common Object Request Broker Architecture
DDA	Driven Decoding Algorithm
EPAC	Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques
ETAPE	Évaluations en Traitement Automatique de la Parole
ESTER	Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques
E.M	Expectation Maximisation
fMLLR	feature-space Maximum Likelihood Linear Regression
fWER	frame Word Error Rate
GMM	Gaussian Mixture Model, modèle de mixtures de gaussiennes.
IDL	Interface Definition Language
IWSLT	International Workshop on Spoken Language Translation
IOR	Interoperable Object Reference
LDA	Linear Discriminant Analysis
LPCC	Linear Prediction Cepstral Cœfficients
MAP	Maximum *A Posteriori*
MLLR	Maximum Likelihood Linear Regression
MLE	Maximum Likelihood Estimation

MFCC	Mel-Frequency Cepstral Cœfficients
ML	Modèle de Langage
MMC	Modèle de Markov Caché
MMIE	Maximum Mutual Information Estimation
NCE	Normalized Cross Entropy
NIST	National Institute of Standards and Technology
ORB	Object Request Broker
PPL	Perplexité
POA	Portable Object Adapter
ROVER	Recognition Output Voting Error Reduction
SRAP	Système de Reconnaissance Automatique de la Parole
SAT	Speaker Adaptive Training
RASR	RWTH Automatic Speech Recognizer
SPEERAL	Système de reconnaissance de la parole du LIA
WFST	Weighted Finite-State Transducer
WER	Word Error Rate

Bibliographie personelle

Conférences d'audience internationale

[Bougares 2012] **Fethi Bougares**, Yannick Estève, Paul Deléglise et Georges Linarès : Low latency combination of parallelized single-pass LVCSR. Interspeech , Portland(USA), 09-14 Spetembre 2012.

[Bougares 2012a] **Fethi Bougares**, Yannick Estève, Paul Deléglise, Michael Rouvier, George Linarès : Avancées dans le domaine de la transcription automatique par décodage guidé. JEP , 04-09 Juin Grenoble(France) 2012.

[Bougares 2011] **Fethi Bougares**, Yannick Estève, Paul Deléglise, Géorges Linarès : Bag of n-gram driven decoding for LVCSR system harnessing. ASRU , Hawaï(USA), 11-15 December 2011.

[Rousseau 2011] Anthony Rousseau, **Fethi Bougares**, Paul Deléglise, Holger Schwenk, Yannick Estève : LIUM's systems for the IWSLT 2011 Speech Translation Tasks. IWSLT, San Francisco(USA), 8-9 Septembre 2011.

[Estève 2010] Yannick Estève, Paul Deléglise, Sylvain Meignier, Simon Petitrenaud, Holger Schwenk, Loïc Barrault, **Fethi Bougares**, Richard Dufour, Vincent Jousse, Antoine Laurent, Anthony Rousseau : Some recent research work at LIUM based on the use of CMU Sphinx. CMU SPUD Workshop, Dallas(Texas), Mars 13, 2010.

[Dufour 2010] Richard Dufour, **Fethi Bougares**, Yannnick Estève, Paul Deléglise : Unsupervised model adaptation on targeted speech segments for LVCSR system combination. Interspeech 2010, Makuhari(Japan), 26-30 september 2010.

[Bougares 2009a] **Fethi Bougares**, Laurent Besacier, Hervé Blanchon : LIG approach for IWSLT09 : Using Multiple Morphological Segmenters for Spoken Language Translation of Arabic. IWSLT, Tokyo, 1-2 December 2009.

Conférences d'audience nationale

[Bougares 2011a] **Fethi Bougares**. Amélioration de la combinaison de systèmes de reconnaissance de la parole par décodage guidé. Rencontre de Jeune Chercheur en Parole, Grenoble(France), 25-27 Mai 2011.

[Bougares 2009a] **Fethi Bougares**. Traduction automatique de la parole arabe/anglais par segmentations multiples. RJCP, Avignon(France), 17-19 Novembre 2009.

Prix

Le prix du meilleur système de traduction de la parole à IWSLT 2011 (San Francisco - USA)

Le prix de la deuxième meilleure présentation orale aux journées des doctorants. JDOC 2011 (Nantes -France)

Bibliographie

[Allauzen 2004] Alexandre Allauzen et Jean-Luc Gauvain. *Construction automatique du vocabulaire d'un système de transcription*. In JEP, Fez, Maroc, April 2004. (Cité en page 39.)

[Anastasakos 1996] Tasos Anastasakos, John McDonough, Richard Schwartz et John Makhoul. *A compact model for speaker-adaptive training*. In In Proceedings ICSLP, volume 2, pages 1137–1140, Philadelphia, PA., 1996. (Cité en page 20.)

[Aubert 2002] Xavier L. Aubert. *An overview of decoding techniques for large vocabulary continuous speech recognition*. Computer Speech & Language, vol. 16, no. 1, pages 89–114, 2002. (Cité en page 26.)

[Baum 1966] Leonard E. Baum et Ted Petrie. *Statistical inference for probabilistic functions of finite state Markov chains*. Annals of Mathematical Statistics, vol. 37, pages 1554–1563, 1966. (Cité en page 17.)

[Bougares 2011] Fethi Bougares, Yannick Estève, Paul Deléglise et George Linarès. *Bag Of N-Gram driven decoding for LVCSR system harnessing*. In ASRU, 11-15 December 2011. (Cité en pages 73, 81 et 93.)

[Bougares 2012] Fethi Bougares, Mickael Rouvier, Yannick Estève et George Linarès. *Low latency combination of parallelized single-pass LVCSR systems*. In Interspeech, Portland, OR, USA, Septembre 2012. (Cité en page 92.)

[Breslin 2006] Catherine Breslin et Mark J. F. Gales. *Generating complementary systems for speech recognition*. In Interspeech, 2006. (Cité en page 46.)

[Breslin 2009] Catherine Breslin et Mark J. F. Gales. *Directed decision trees for generating complementary systems*. Speech Commun., vol. 51, pages 284–295, March 2009. (Cité en page 47.)

[Brown 1992] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra et Jenifer C. Lai. *Class-Based n-gram Models of Natural Language*. Computational Linguistics, vol. 18, pages 467–479, 1992. (Cité en pages 22 et 23.)

[Burget 2004] Lukas Burget. *Measurement of Complementarity of Recognition Systems*. In Proceedings of the 5th International Conference on Text, Speech and Dialogue, pages 283–290, 2004. (Cité en pages 48 et 49.)

[Béchet 2001] Frédéric Béchet. *LIA_PHON, un système complet de phonétisation de texte*. Traitement Automatique des Langues, vol. 42, 2001. (Cité en page 39.)

[Chen 1999] Stanley F. Chen et Joshua Goodman. *An empirical study of smoothing techniques for language modeling*. In Computer Speech and Language, pages 359–394, 1999. (Cité en pages 22 et 41.)

[Chen 2006] I-Fan Chen et Lin-Shan Lee. *A new framework for system combination based on integrated hypothesis space*. In Interspeech, 2006. (Cité en pages 60 et 61.)

[Chong 2010] Jike Chong, Gerald Friedland, Adam Janin, Nelson Morgan et Chris Oei. *Opportunities and challenges of parallelizing speech recognition*. In Proceedings of the 2nd USENIX conference on Hot topics in parallelism, Berkeley, CA, USA, 2010. (Cité en page 85.)

[Davis 1990] Steven B. Davis et Paul Mermelstein. *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. In Alex Waibel et Kai-Fu Lee, editeurs, Readings in speech recognition, pages 65–74. San Francisco, CA, USA, 1990. (Cité en page 13.)

[Deléglise 2005] Paul Deléglise, Yannick Estève, Sylvain Meignier et Téva Merlin. *The LIUM Speech Transcription System : a CMU Sphinx III-based System for French Broadcast News*. In Interspeech 2005, pages 1653–1656, Lisbon, Portugal, September 2005. (Cité en pages 36 et 68.)

[Deléglise 2009] Paul Deléglise, Yannick Estève, Sylvain Meignier et Téva Merlin. *Improvements to the LIUM French ASR system based on CMU sphinx : what helps to significantly reduce the word error rate ?* In Interspeech, pages 2123–2126, Brighton UK, 2009. (Cité en page 36.)

[Dempster 1977] A.P. Dempster, N.M. Laird et D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, vol. 39, pages 1–38, 1977. (Cité en page 19.)

[Digalakis 1995] Vassilios Digalakis et Leonardo Neumeyer. *Speaker adaptation using combined transformation and Bayesian methods*. In ICASSP, pages 680–683, Detroit, Michigan, USA, May 1995. (Cité en page 20.)

[Ellis 2000] Daniel P. W. Ellis. *Stream combination before and/or after the acoustic model*. In ICASSP, pages 1635–1638, 2000. (Cité en page 50.)

[Estève 2002] Yannick Estève. *Intégration de sources de connaissances pour la modélisation stochastique du langage appliquée a la parole continue dans un contexte de dialogue oral homme-machine*. PhD thesis, Université d'avignon, 2002. (Cité en page 24.)

[Estève 2004] Yannick Estève, Paul Deléglise et Bruno Jacob. *Systèmes de transcription automatique de la parole et logiciels libres*. Traitement Automatique des Langues, vol. 37, 2004. (Cité en page 36.)

[Estève 2009] Yannick Estève. *Traitement automatique de la parole : contributions*. In Habilitation à Diriger des Recherches (HDR), LIUM, Université du Maine, 2009. (Cité en page 36.)

[Estève 2010] Yannick Estève, Thierry Bazillon, Jean-Yves Antoine, Frédéric Béchet et Jérôme Farinas. *The EPAC corpus : manual and automatic annotations of conversational speech in French broadcast news*. In LREC 2010, Malta, 17-23 may 2010. (Cité en page 38.)

[Evermann 2000] G. Evermann et P.C. Woodland. *Posterior Probability Decoding, Confidence Estimation And System Combination*. In Proceedings NIST Speech Transcription Workshop, 2000. (Cité en page 58.)

[Federico 2011] Marcello Federico, Luisa Bentivogli, Michael Paul et Sebastian Stüker. *Overview of the IWSLT 2011 Evaluation Campaign*. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT), San Francisco (CA), December 2011. (Cité en page 79.)

[Fiscus 1997] J. Fiscus. *A post-prcessing system to yield reduced word error rates : Recogniser Output Voting Error Reduction (ROVER)*. In ASRU, pages 347–354, 1997. (Cité en pages 55 et 56.)

[Freund 1995] Yoav Freund et Robert E. Schapire. *A decision-theoretic generalization of on-linelearning and an application to boosting*. European Conference on Computational Learning Theory (EUROCOLT), Mars 1995. (Cité en page 47.)

[Gales 1996a] M.J F. Gales. *The Generation And Use Of Regression Class Trees For Mllr Adaptation*. Rapport technique, Cambridge University Engineering Department, 1996. (Cité en page 19.)

[Gales 1996b] M.J.F. Gales et P.C. Woodland. *Mean and Variance Adaptation within the MLLR Framework*. Computer Speech and Language, vol. 10, pages 249–264, 1996. (Cité en page 19.)

[Gales 1998] M.J.F. Gales. *Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition*. Computer Speech and Language, vol. 12, pages 75–98, 1998. (Cité en page 20.)

[Gales 2006] M. J. F. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha et S. E. Tranter. *Progress in the CU-HTK broadcast news transcription system*. In IEEE transactions speech and audio processing, 2006. (Cité en page 91.)

[Galliano 2006] S. Galliano, E. Geoffrois, G. Gravier, J.F. Bonastre, D. Mostefa et K. Choukri. *Corpus description of the ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News*. In Proceedings of the 5th Intl. Conf. on Language Resources and Evaluations, LREC 2006, Genoa, Italie, May 2006. (Cité en page 67.)

[Gauvain 1994] Jean-luc Gauvain et Lee Chin-hui. *Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains*. IEEE Transactions on Speech and Audio Processing, vol. 2, pages 291–298, 1994. (Cité en page 17.)

[Gu 2008] Liang Gu, Jian Xue, Xiaodong Cui et Yuqing Gao. *High-performance low-latency speech recognition via multi-layered feature streaming and fast Gaussian computation*. In Interspeech, pages 2098–2101, 2008. (Cité en page 85.)

[Haeb-Umbach 1992] R. Haeb-Umbach et H. Ney. *Linear discriminant analysis for improved large vocabulary continuous speech recognition*. In IEEE ICASSP, pages 13–16, 1992. (Cité en page 95.)

[Hart 1968] Peter Hart, Nils Nilsson et Bertram Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics (SSC4), vol. 2, pages 100–107, 1968. (Cité en page 30.)

[Hermansky 1991] Hynek Hermansky et Louis Anthony Cox Jr. *Perceptual linear predictive (PLP) analysis-resynthesis technique.* In Eurospeech, pages 037– 038, 1991. (Cité en page 13.)

[Hillard 2007] D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schlüter et H. Ney. *iROVER : Improving System Combination with Classification*. In The Conference of the North American Chapter of the Association for Computational Linguistics, pages 65–68, Rochester, New York, avril 2007. (Cité en page 57.)

[Hoffmeister 2006] Björn Hoffmeister, Tobias Klein, Ralf Schlüter et Hermann Ney. *Frame based system combination and a comparison with weighted ROVER and CNC.* In Interspeech, 2006. (Cité en pages 62 et 95.)

[Huang 2001] Xuedong Huang, Alex Acero et Hsiao-Wuen Hon. Spoken language processing : a guide to theory, algorithm and system development. Prentice-Hall Inc, 2001. (Cité en page 13.)

[Huet 2007] Stéphane Huet, Guillaume Gravier et Pascale Sébillot. *Morphosyntactic Processing of N-Best Lists for Improved Recognition and Confidence Measure Computation*. In Eurospeech'07, pages 1741–1744, Anvers, Belgique, 2007. (Cité en page 69.)

[Iyer 1996] R. Iyer et M. Ostendorf. *Modeling Long Distance Dependence in Language : Topic Mixtures vs. Dynamic Cache Models*. In IEEE Transactions on Speech and Audio Processing, pages 236–239, 1996. (Cité en page 25.)

[Jelinek 1977a] F. Jelinek, R. L. Mercer, L. R. Bahl et J. K. Baker. *Perplexity – a measure of the difficulty of speech recognition tasks*. Journal of

the Acoustical Society of America, vol. 62, page S63, November 1977. (Cité en page 25.)

[Jelinek 1977b] Fred Jelinek. *Continuous speech recognition*. SIGART Bull., pages 33–34, February 1977. (Cité en page 12.)

[Jelinek 1980] Fred Jelinek et Robert L. Mercer. *Interpolated estimation of Markov source parameters from sparse data*. In Proceedings, Workshop on Pattern Recognition in Practice. Amsterdam, 1980. (Cité en page 52.)

[Kanthak 2002] Stephan Kanthak, Hermann Ney, Michael Riley et Mehryar Mohri. *A comparison of two LVR search optimization techniques*. In Interspeech, 2002. (Cité en page 27.)

[Kirchhoff 1998] Katrin Kirchhoff. *Combining Articulatory And Acoustic Information For Speech Recognition In Noisy And Reverberant Environments*. In Proceedings of ICSLP, pages 891–4, 1998. (Cité en page 52.)

[Kneser 1993] Reinhard Kneser et Hermann Ney. *Improved clustering techniques for class-based statistical language modeling*. In In Proceedings of the European Conference on Speech Communication and Technology (Eurospeech), 1993. (Cité en pages 22 et 23.)

[Kneser 1995] R. Kneser et H. Ney. *Improved backing-off for n-gram language modeling*. In ICASSP, 1995. (Cité en pages 22 et 41.)

[Knill 1996] K.M. Knill, M.J.F. Gales et S.J. Young. *Use Of Gaussian Selection In Large Vocabulary Continuous Speech Recognition Using HMMs*, 1996. (Cité en page 35.)

[Kuhn 1990] R. Kuhn et R. De Mori. *A Cache-Based Natural Language Model for Speech Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 6, 1990. (Cité en page 53.)

[Lacouture 1995] Roxane Lacouture. *Au sujet des algorithmes de recherche des systèmes de reconnaissance de la parole à grands vocabulaires*. Ph.d. dissertation, School of Computer Science Université McGill,, Montréal, 1995. (Cité en page 27.)

[Lamel 2006] Lori Lamel, Jean-Luc Gauvain, Gilles Adda, Claude Barras, Eric Bilinski, Olivier Galibert, Agusti Pujol, Holger Schwenk et Xuan Zhu. *The LIMSI 2006 TC-STAR Transcription Systems.* In TC-STAR Workshop on Speech-to-Speech Translation, pages 123–128, Juin 2006. (Cité en page 91.)

[Lau 1993] R. Lau, R. Rosenfeld et S. Roukos. *Trigger-based language models : a maximum entropy approach*. ICASSP, vol. 2, pages 45–48, 1993. (Cité en page 53.)

[Lecouteux 2006] Benjamin Lecouteux, Georges Linarès, Pascal Nocera et Jean-François Bonastre. *Imperfect transcript driven speech recognition*. In ICSLP /Interspeech, Pittsburgh, Pennsylvania, USA, 2006. (Cité en pages 62 et 66.)

[Lecouteux 2007] Benjamin Lecouteux, Georges Linarès, Yannick Estève et Julie Mauclair. *System Combination by Driven Decoding*. In ICASSP, 2007. (Cité en pages 8, 62, 66 et 67.)

[Lecouteux 2008a] Benjamin Lecouteux. *Reconnaissance automatique de la parole guidée par des transcriptions a priori*. PhD thesis, Université d'avignon, 2008. (Cité en page 68.)

[Lecouteux 2008b] Benjamin Lecouteux, Georges Linarès, Yannick Estève et Guillaume Gravier. *Generalized driven decoding for speech recognition system combination*. In ICASSP, Las Vegas, Nevada, USA, 2008. (Cité en pages 67 et 68.)

[Leggetter 1995] C. J. Leggetter et P. C. Woodland. *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*. Computer Speech & Language, vol. 9, no. 2, pages 171–185, april 1995. (Cité en page 18.)

[Li 2002] Xiang Li, Rita Singh et Richard M. Stern. *Combining search spaces of heterogeneous recognizers for improved speech recognition*. In Proc. ICSLP, 2002. (Cité en page 59.)

[Liu 2009] Xunying Liu, Mark J. F. Gales et Philip C. Woodland. *Use of contexts in language model interpolation and adaptation*. In Interspeech, pages 360–363, 2009. (Cité en page 55.)

[Liu 2010] Xunying Liu, Mark J. F. Gales et Philip C. Woodland. *Language model cross adaptation for LVCSR system combination*. In Interspeech, pages 342–345, 2010. (Cité en page 55.)

[Liu 2011] Xunying Liu, Mark J. F. Gales et Philip C. Woodland. *Improving LVCSR System Combination Using Neural Network Language Model Cross Adaptation*. In Interspeech, pages 2857–2860, 2011. (Cité en page 55.)

[Lööf 2007] J. Lööf, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter et H. Ney. *The RWTH 2007 TC-STAR Evaluation System for European English and Spanish*. In Interspeech, Antwerp, Belgium, 2007. (Cité en page 95.)

[Mangu 1999] Lidia Mangu, Eric Brill et Andreas Stolcke. *Finding Consensus Among Words : Lattice-Based Word Error Minimization*. In in Proc. Eurospeech, pages 495–498, 1999. (Cité en page 32.)

[Markel 1982] John E. Markel et A. H. Gray. Linear prediction of speech. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982. (Cité en page 13.)

[Mauclair 2006] Julie Mauclair. *Mesures de confiances en traitement automatique de la parole et applications*. PhD thesis, LIUM, Université du Maine Le Mans France, 2006. (Cité en page 33.)

[Meignier 2010] Sylvain Meignier et Téva Merlin. *LIUM SpkDiarization : an open source toolkit for diarization*. In CMU SPUD Workshop, Dallas, Texas, USA, 2010. (Cité en page 42.)

[Mohri 2002] Mehryar Mohri, Fernando Pereira et Michael Riley. *Weighted finite-state transducers in speech recognition*. Computer Speech & Language, vol. 16, no. 1, pages 69–88, 2002. (Cité en page 27.)

[Ney 1999] Hermann Ney et Stefan Ortmanns. *Dynamic Programming Search for Continuous Speech Recognition*. IEEE Signal Processing Magazine, vol. 16, no. 5, pages 64–83, sep 1999. (Cité en page 26.)

[Nocera 2004] Pascal Nocera, Corinne Fredouille, Georges Linarès, Driss Matrouf, Sylvain Meignier, Jean-François Bonastre, Dominique Massonié et Fréderic Béchet. *The LIA's French broadcast news transcription system.* In SWIM : Lectures by Masters in Speech Processing, Maui, Hawaii, 2004. (Cité en page 69.)

[OMG. 2004] OMG. *The Common Object Request Broker Architecture : Core Specification, V. 3.0.2. Object Management Group*. Rapport technique, 2004. (Cité en page 88.)

[Ortmanns 1997] S. Ortmanns, A. Eiden, H. Ney et N. Coenen. *Look-Ahead Techniques for Fast Beam Search*. In ICASSP, 1997. (Cité en page 35.)

[Ortmanns 1998] S. Ortmanns, H. Ney et A. Eiden. *Language-Model Look-Ahead for Large Vocabulary Speech Recognition.* In International Conference on Spoken Language Processing, pages 2095–2098, Sydney, Australia, oct 1998. (Cité en page 35.)

[Ostendorf 1991] M. Ostendorf, A. Kannan, S. Auagin et O. Kimball. *Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses.* In Proceedings DARPA Speech and Natural Language Processing Workshop, pages 83–87, 1991. (Cité en page 91.)

[Pallett 1990] D. Pallett, W. Fisher et J. Fiscus. *Tools for the Analysis of Benchmark Speech Recognition Tests*. In ICASSP, pages 97–100, Alburquerque, (Nouveau-Mexique) États-Unis, Avril 1990. (Cité en page 76.)

[Perennou 1987] Guy Perennou et Martine de Calmès. *BDLEX lexical data and knowledge base of spoken and written French*. In European Conference on Speech Technology, 1987. (Cité en page 39.)

[Phillips 1999] Steven Phillips et Anne Rogers. *Parallel Speech Recognition*. International Journal of Parallel Programming, vol. 27, no. 4, pages 257–288, 1999. (Cité en page 36.)

[Povey 2011] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer et Karel Vesely. *The Kaldi Speech Recognition Toolkit*. In IEEE 2011 ASRU, Décembre 2011. (Cité en page 27.)

[Psutka 2001] Josef Psutka, Ludek Müller et Josef V. Psutka. *Comparison of MFCC and PLP parameterizations in the speaker independent continuous speech recognition task*. In Interspeech, pages 1813–1816, 2001. (Cité en page 113.)

[Rabiner 1990] Lawrence R. Rabiner. *Readings in speech recognition*. In A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. San Francisco, CA, USA, 1990. (Cité en page 14.)

[Rousseau 2011] Anthony Rousseau, Fethi Bougares, Paul Deléglise, Holger Schwenk et Yannick Estève. *Overview of the IWSLT 2011 Evaluation Campaign*. In LIUM's systems for the IWSLT 2011 Speech Translation Tasks, San Francisco (CA), December 2011. (Cité en page 80.)

[Sankar 2005] Ananth Sankar. *Bayesian Model Combination (BAYCOM) for Improved Recognition*. In ICASSP, volume 1, pages 845–848, 2005. (Cité en page 58.)

[Saraclar 2002] Murat Saraclar, Michael Riley, Enrico Bocchieri et Vincent Goffin. *Towards automatic closed captioning : low latency real time broadcast news transcription.* In Interspeech. ISCA, 2002. (Cité en page 84.)

[Schapire 2003] R. E. Schapire. *The boosting approach to machine learning : An overview*. Nonlinear Estimation and Classification, 2003. (Cité en page 47.)

[Schwenk 2000] Holger Schwenk et Jean-Luc Gauvain. *Combining multiple speech recognizers using voting and language model information*. In Interspeech, pages 915–918, 2000. (Cité en page 56.)

[Seward 2003] Alexander Seward. *Low-latency incremental speech transcription in the synface project*. In Interspeech, 2003. (Cité en page 84.)

[Singh-miller 2007] Natasha Singh-miller. *Trigger-Based Language Modeling Using a Loss-Sensitive Perceptron Algorithm.* IEEE ICASSP, 2007. (Cité en page 53.)

[Singh 2001] Rita Singh, Michael L. Seltzer, Bhiksha Raj et Richard M. Stern. *Speech in Noisy Environments : robust automatic segmentation, feature extraction, and hypothesis combination.* IEEE ICASSP, vol. 1, pages 273–276, 2001. (Cité en page 57.)

[Siohan 2005] Olivier Siohan, Bhuvana Ramabhadran et Brian Kingsbury. *Constructing ensembles of ASR systems using randomized decision trees.* In IEEE ICASSP, pages 197–200, Philadelphia, USA, May 2005. (Cité en page 47.)

[Stemmer 2002] Georg Stemmer, Stefan Steidl, Elmar Nöth, Heinrich Niemann et Anton Batliner. *Comparison and Combination of Confidence Measures.* In Proceedings of the 5th International Conference on Text, Speech and Dialogue, pages 181–188, 2002. (Cité en page 33.)

[Stolcke 2002] Andreas Stolcke. *SRILM-An extensible language modeling toolkit.* In ICSLP 2002, volume 2, pages 901–904, Denver, Colorado, USA, 2002. (Cité en page 41.)

[Stüker 2006] S. Stüker, C Fügen, S. Burger et M. Wölfel. *Cross-System Adaptation and Combination for Continuous Speech Recognition : The Influence of Phoneme Set and Acoustic Front-End ?* In Interspeech, 2006. (Cité en page 53.)

[Suh 2007] Youngjoo Suh, Sungtak Kim et Hoirin Kim. *Compensating acoustic mismatch using class-based histogram equalization for robust speech recognition.* EURASIP J. Appl. Signal Process., vol. 2007, no. 1, Janvier 2007. (Cité en page 47.)

[Utsuro 2005] Takehito Utsuro, Yasuhiro Kodama, Tomohiro Watanabe, Hiromitsu Nishizaki et Seiichi Nakagawa. *Combining outputs of multiple LVCSR models by machine learning.* Syst. Comput. Japan, vol. 36, no. 10, pages 9–15, Septembre 2005. (Cité en page 57.)

[Valtchev 1997] V. Valtchev, J. J. Odell, P. C. Woodland et S. J. Young. *MMIE training of large vocabulary recognition systems.* Speech Communication, vol. 22, pages 303–314, September 1997. (Cité en page 17.)

[Venkataramani 2005] Veera Venkataramani et William Byrne. *Lattice Segmentation and Support Vector Machines for Large Vocabulary Continuous Speech Recognition.* In ICASSP, volume 1, pages 817–820, 2005. (Cité en page 48.)

[Viterbi 1967] A. Viterbi. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. IEEE Transactions on Information Theory, vol. 13, no. 2, April 1967. (Cité en page 28.)

[Vogelgesang 2011] Matthias Vogelgesang et Florian Metze. *parallelization strategies for a dynamic lexical tree decoder*. In Technical report cmulti-010, Language Technologies Institute, Carnegie Mellon University, 2011. (Cité en page 36.)

[Wessel 2001] Frank Wessel, Ralf Schlüter, Klaus Macherey et Hermann Ney. *Confidence Measures for Large Vocabulary Continuous Speech Recognition*. IEEE Transactions on Speech and Audio Processing, vol. 9, pages 288–298, 2001. (Cité en page 33.)

[Yongxin 2002] Li. Yongxin, H. Erdogan, Y. Gao et E. Marcheret. *Incremental on-line feature space MLLR adaptation for telephony speech recognition*. In Interspeech, 2002. (Cité en page 20.)

[Zolnay 2005] András Zolnay, Ralf Schlüter et Hermann Ney. *Acoustic Feature Combination for Robust Speech Recognition*. In IEEE ICASSP, pages 457–460, 2005. (Cité en page 52.)

[Zouari 2007] Leila Zouari. *Vers le temps réel en transcription automatique de la parole grand vocabulaire*. These, Télécom ParisTech, 2007. (Cité en page 35.)

[Zweig 2000] Geoffrey Zweig et Mukund Padmanabhan. *Boosting Gaussian mixtures in an LVCSR system*. In ICASSP, pages 1527–30, 2000. (Cité en page 47.)