Vers une modélisation statistique multi-niveau du langage, application aux langues peu dotées

Sopheap Seng

UNIVERSITÉ DE GRENOBLE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

Docteur de L'Université de Grenoble

Spécialité : **Informatique**

préparée au **Laboratoire Informatique de Grenoble**

dans le cadre de l'École Doctorale **MSTII**

présentée et soutenue publiquement

par

Sopheap SENG

le 01/03/2010

Titre :

Vers une modélisation statistique multi-niveau du langage, application aux langues peu dotées

Directeur de thèse : **Laurent Besacier**

Co-directeur de thèse : **Eric Castelli**

Jury

M. Christian Boitet,	Président du jury
Mme Lori Lamel,	Rapporteur du jury
M. Frédéric Béchet,	Rapporteur du jury
Mme Tanja Schultz,	Membre du jury
M. Vincent Berment,	Membre du jury
Mme Brigitte Bigi,	Membre du jury
M. Laurent Besacier,	Membre du jury
M. Eric Castelli,	Membre du jury

Remerciements

Je tiens tout d'abord à remercier Laurent Besacier et Eric Castelli pour avoir accepté d'encadrer cette thèse. Un grand remerciement à Laurent Besacier, qui m'a guidé tout au long de ces années de thèse, pour ses critiques, ses conseils sur mes travaux de recherche et pour avoir relu, corrigé et commenté ce manuscrit. Je voudrais remercier Brigitte Bigi pour son aide dévouée sur mes travaux de thèse, pour ses conseils très utiles et sa relecture de tout mon manuscrit.

J'adresse mes remerciements à Lori Lamel et Frédéric Béchet pour avoir accepté d'être rapporteurs de ma thèse. Je voudrais remercier aussi Christian Boitet pour avoir accepté d'être le président du jury. Je remercie Tanja Schultz et Vincent Berment pour sa participation au jury de cette thèse.

Je tiens à remercier Pham Thi Ngoc Yen, la directrice du Centre MICA (Hanoi, Vietnam) pour m'avoir accueilli pour un stage de recherche à MICA.

J'adresse mes remerciements à Alex Waibel et Sebastian Stüker pour m'avoir accueilli dans le laboratoire Interactive Systems Labs (Université de Karlsruhe) et pour l'intérêt porté à mes travaux de recherche.

Je tiens à remercier également tous les membres de l'équipe GETALP pour leur accueil et leur sympathie. Un grand remerciement à mes amis à Grenoble avec qui j'ai partagé de grands moments au cours de ma thèse.

Enfin, je voudrais exprimer mes plus profonds remerciements à mes parents, à ma soeur, ma petite amie, pour leurs sentiments, leurs soutiens et leurs encouragements dans tout le temps où j'ai effectué cette thèse.

Résumé

Ce travail de thèse porte sur la reconnaissance automatique de la parole des langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots. La spécificité des langues traitées dans notre contexte d'étude nécessite la segmentation automatique en mots pour rendre la modélisation du langage n-gramme applicable. Alors que le manque de données textuelles a un impact sur la performance des modèles de langage, les erreurs introduites par la segmentation automatique peuvent rendre ces données encore moins exploitables. Pour tenter de pallier les problèmes, nos recherches sont axées principalement sur la modélisation du langage, et en particulier sur le choix des unités lexicales et sous-lexicales, utilisées par les systèmes de reconnaissance. Nous expérimentons l'utilisation des multiples unités au niveau des modèles du langage et au niveau des sorties de systèmes de reconnaissance. Au niveau des modèles de langage, les modèles sont entraînés avec des vocabulaires hybrides créés en utilisant à la fois l'unité lexicale et l'unité sous-lexicale. Au niveau des sorties de systèmes, nous essayons de combiner les sorties de plusieurs systèmes de reconnaissance. Chaque système est fondé sur une unité de modélisation : lexicale ou sous-lexicale. Dans un objectif consistant à mieux exploiter les données textuelles en utilisant différentes vues sur données, nous proposons une méthode qui effectue des segmentations multiples sur le corpus d'apprentissage au lieu d'une segmentation unique classique. Cette méthode de segmentation multiple basée sur des automates d'état finis permet de générer toutes les segmentations possibles à partir d'une séquence de caractères et nous pouvons ensuite en extraire les n-grammes pour apprendre le modèle de

langage. Elle permet de trouver les n-grammes non obtenus par la segmentation unique et d'ajouter de nouveaux n-grammes dans le modèle de langage. Nous validons ces approches de modélisation à base des multiples unités sur les systèmes de reconnaissance pour un groupe de langues peu dotées : le khmer, le vietnamien, le thaï et le laotien.

Mots-clés : reconnaissance automatique de la parole, langue peu dotée, modélisation statistique multi-niveau du langage.

Abstract

This PhD thesis focuses on the problems encountered when developing automatic speech recognition for under-resourced languages with a writing system without explicit separation between words. The specificity of the languages covered in our work requires automatic segmentation of text corpus into words in order to make the n-gram language modeling applicable. While the lack of text data has an impact on the performance of language model, the errors introduced by automatic segmentation can make these data even less usable. To deal with these problems, our research focuses primarily on language modeling, and in particular the choice of lexical and sub-lexical units, used by the recognition systems. We investigate the use of multiple units in speech recognition system. At language models level, the models are trained with hybrid vocabularies created using both the lexical and the sub-lexical unit. At the system output level, we try to combine the outputs of several recognition systems. Each system is based on a different modeling unit : lexical or sub-lexical. To better exploit the textual data using different views on the same data, we propose a method that performs multiple segmentations on the training corpus instead of a conventional single segmentation. This method based on finite state machines allows generating all possible segmentations from a sequence of characters and then we can extract n-grams to train the language model. It allows finding the n-grams not found by unique segmentation method and adding new n-grams in the language model. We validate these modeling approaches based on multiple units in recognition systems for a group of languages : Khmer, Vietnamese, Thai and Laotian.

Keywords : Automatic speech recognition, under-resourced language, multi-level statistical language modeling.

Table des matières

Table des figures

Liste des tableaux

Introduction

La reconnaissance automatique de la parole consiste à extraire, à l'aide d'un ordinateur, l'information lexicale contenue dans un signal de parole. Les applications de cette technologie sont nombreuses. Il existe des logiciels de dictée, des systèmes d'indexation automatique de documents audiovisuels ou des systèmes de dialogue. Les sorties du système de reconnaissance automatique de la parole peuvent également servir d'entrée à d'autres systèmes, par exemple pour la traduction dans le but construire un système de traduction automatique de la parole.

Parmi les 6000 langues parlées dans le monde, seul un tout petit nombre d'entre-elles possède les ressources nécessaires pour implémenter des technologies issues du traitement du langage naturel. Il s'agit des langues des pays développés ou des langues qui présentent un intérêt stratégique ou politique, comme par exemple l'anglais, le français, l'allemand, le mandarin, le japonais, l'arabe. Depuis plus de deux décennies, des recherches intensives dans ce domaine ont été accomplies par de nombreux laboratoires internationaux. Des progrès importants ont été accomplis grâce notamment aux efforts de collecte des données linguistiques nécessaires pour la modélisation statistique de la parole.

En reconnaissance automatique de la parole, il subsiste un certain nombre de verrous, notamment en ce qui concerne la généricité des méthodes utilisées et leur portabilité vers de nouvelles langues. Premièrement, les approches statistiques utilisées dans la modélisation de la parole nécessitent de très grands corpus de données pour construire des modèles performants. Pour les langues parlées dans les pays en voie de développement ou pour les langues qui ne suscitent pas d'in-

térêt économique ou politique, ces ressources sont généralement disponibles en quantité insuffisante pour le développement d'un tel système. Ces langues sont appelées des langues "peu dotées" dans plusieurs études, notamment dans la thèse de V. Berment intitulée "Méthodes pour informatiser des langues et des groupes de langues peu dotées" [Berment, 2004] et dans la thèse de V-B. Le intitulée "Reconnaissance Automatique de la parole des langues peu dotées" [Le, 2006], des travaux réalisés ces dernières années au LIG. Deuxièmement, les méthodes de modélisation qui ont été initialement étudiées pour les langues comme l'anglais ou le français ne sont pas directement applicables sur les autres langues qui possèdent des caractéristiques différentes. Par exemple, pour beaucoup de langues, déterminer la frontière des mots dans le texte est une tâche particulièrement difficile comparativement à une langue comme l'anglais et la méthode de modélisation statistique du langage par n-grammes ne peut pas s'appliquer directement sur le corpus de texte comme dans le cas de l'anglais ou du français.

Ce travail de thèse porte sur la reconnaissance automatique de la parole des langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots. La spécificité des langues traitées dans notre contexte d'étude nécessite la segmentation automatique en mots pour rendre la modélisation du langage n-gramme applicable. Alors que le manque de données textuelles a un impact sur la performance des modèles de langage, les erreurs introduites par la segmentation automatique peuvent rendre ces données encore moins exploitables. Pour tenter de pallier les problèmes, à savoir des taux de mots inconnus élevés et des modèles de langage peu fiables conséquence du manque de données textuelles et des erreurs de segmentation automatique en mots, nos recherches sont axées principalement sur la modélisation du langage, et en particulier sur le choix des unités lexicales et sous-lexicales, utilisées par les systèmes de reconnaissance. Le problème de la faible quantité de données textuelles implique de réfléchir à des techniques de modélisation lexicale et sous-lexicale permettant ainsi de réduire la taille du vocabulaire de l'application, tout en essayant d'exploiter au mieux les données. Nous proposons de traiter ce problème en exploitant plusieurs vues sur les données textuelles dans la modélisation du langage. Nous expérimentons l'utilisation des multiples unités au niveau des modèles du langage et au niveau des sorties de systèmes de reconnaissance. Au niveau des modèles de langage, les

modèles sont entraînés avec des vocabulaires hybrides créés en utilisant à la fois l'unité lexicale et l'unité sous-lexicale. Au niveau des sorties de systèmes, nous essayons de combiner les sorties de plusieurs systèmes de reconnaissance. Chaque système est fondé sur une unité de modélisation : lexicale ou sous-lexicale. Nous cherchons à valider ces approches de modélisation à base des multiples unités sur les systèmes de reconnaissance pour un groupe de langues peu dotées : le khmer, le vietnamien, le thaï et le laotien.

D'un point de vue plus opérationnel, nous développons dans le cadre de ce travail de thèse, les systèmes de reconnaissance automatique de la parole pour deux langues peu dotées parlées en Asie du sud-est : le khmer et le laotien. Nous développons donc un système de reconnaissance automatique de la parole de l'état de l'art pour ces deux langues (broadcast news) à partir des ressources collectées. Ce travail permet ainsi de revisiter les méthodes et les outils de l'état de l'art proposées pour la collecte rapide de données et le développement rapide d'un système de reconnaissance pour une nouvelle langue peu dotée.

Ce mémoire se compose de 4 chapitres :

- dans le chapitre 1, après une brève présentation de la motivation et du contexte de ces travaux de thèse, nous présentons le principe général de la reconnaissance automatique de la parole par modèles statistiques et citons les problèmes spécifiques aux langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots. L'accent est mis sur les problèmes posés par le manque de ressources numériques et les erreurs dues à la segmentation automatique en mots ;
- le chapitre 2 décrit les différentes étapes de développement d'un système de reconnaissance de la parole pour une langue peu dotée, le khmer, langue officielle du Cambodge. Nous décrivons tout d'abord notre méthode de collecte de données linguistiques pour le développement rapide d'un nouveau système de reconnaissance pour une langue peu dotée. Le problème du manque de données textuelles et de la présence des erreurs lors de la segmentation en mots en fonction du taux des mots hors vocabulaire est abordé. Plusieurs unités de modélisations sont proposées dans la modélisation statistique du langage pour le khmer. Nous utilisons en plus de l'unité classique "mot",

les sous-unités “syllabe” et “groupe de caractères” pour modéliser le khmer. Pour la modélisation acoustique, nous présentons et comparons des méthodes de génération automatique de dictionnaires de prononciation à base de graphèmes et à base de règles de conversion graphèmes-phonèmes pour le khmer. Enfin, des expérimentations sont menées pour tester et comparer les approches proposées ;

– dans le chapitre 3, nous souhaitons analyser comment les différentes unités lexicales et sous-lexicales peuvent être exploitées au mieux dans la reconnaissance automatique de la parole des langues peu dotées et non-segmentées. Nous essayons de traiter le problème en exploitant plusieurs vues sur les données textuelles dans la modélisation. Nous travaillons au niveau du modèle de langage en créant des modèles à partir de vocabulaires hybrides qui utilisent à la fois des unités lexicales et sous-lexicales. Au niveau du système, nous proposons de combiner des sorties de différents systèmes fondés sur ces différentes unités pour décoder une meilleure hypothèse. Nous appliquons ces deux méthodes à la reconnaissance automatique de la parole de deux langues peu dotées, le vietnamien et le khmer.
– dans un objectif consistant à mieux exploiter les données textuelles en utilisant différentes vues sur les mêmes données, le chapitre 4 propose une méthode qui effectue des segmentations multiples sur le corpus d'apprentissage au lieu d'une segmentation unique classique. Cette méthode de segmentation multiple basée sur des automates d'état finis permet de générer toutes les segmentations possibles à partir d'une séquence de caractères et nous pouvons ensuite en extraire les n-grammes pour apprendre le modèle de langage. Elle permet de retrouver les n-grammes non trouvés par la segmentation unique et d'ajouter de nouveaux n-grammes dans le modèle de langage. Ce dernier peut être vu comme une sorte de sur-génération des n-grammes à partir d'un corpus de texte. Cette approche par segmentation multiple est comparée avec la méthode classique de segmentation unique dans l'apprentissage des modèles de langage pour les systèmes de reconnaissance automatique de la parole en langue khmère, laotienne, thaïe et vietnamienne.

Chapitre 1

Contexte d'étude et état de l'art

1.1 Contexte

1.1.1 Motivations

Idéalement, informatiser une langue consiste à mettre à la disposition de l'utilisateur humain tous les moyens dont il a besoin dans sa langue, qu'elle soit écrite ou non : dialogue avec la machine, outils pour écrire ou lire un texte, reconaissance automatique de la parole, synthèse vocale, traduction informatisée dans une autre langue, etc. L'absence des outils informatiques élémentaires dans la langue d'un pays rend l'accès aux informations difficile voir impossible et cela renforce la fracture numérique entre les pays. La fracture numérique peut être définie comme une inégalité face aux possibilités d'accéder et de contribuer à l'information, à la connaissance, ainsi que de bénéficier des capacités majeures de développement offertes par les nouvelles technologies de information et de la communication (NTIC).

Pour entrer dans le monde numérique d'aujourd'hui sans renier sa culture, une nation doit le faire en utilisant des logiciels dans sa propre langue. Les logiciels en langue étrangère exacerbent la fracture numérique, rendent les formations de base en informatique difficiles et coûteuses, appauvrit la culture, et bloque la plupart des traitements informatiques de base pour la gouvernance du pays.

Parmi les 6000 langues parlées dans le monde, seul un tout petit nombre d'entre-elles possède les ressources nécessaires pour implémenter des technologies issues du traitement du langage naturel. Pour ces langues dites bien dotées, un certain nombre de ressources est disponible en grande quantité, à savoir : une orthographe stable dans un système d'écriture donné, des ouvrages de référence (grammaires, dictionnaires), des œuvres de diffusion massive (presse écrite et audiovisuelle, films, chansons et musique), des ouvrages techniques et d'apprentissage (publications techniques et scientifiques, ouvrages didactiques) et un nombre abondant d'applications informatiques dans cette langue. D'un autre côté, un très grand nombre de langues dites peu dotées, parlées généralement dans les pays en voie de développement, ne dispose pas suffisamment, voire pas du tout, des ressources dont sont généralement dotées les grandes langues. Une langue peut être majoritaire, écrite, enseignée à l'école, mais manquer cruellement de ressources informatiques ou même de ressources linguistiques en quantité et en qualité suffisantes. Les langues dites peu dotées peuvent être en effet des langues en grand danger de disparition ou bien des langues émergentes qui possèdent déjà une bonne partie de ces ressources mais en nombre estimé insuffisant et incomplet.

D'une manière générale, pour les langues peu dotées, les technologies vocales ne sont peut-être pas la première lacune à combler, les outils de traitement informatique de bases comme la saisie, l'affichage, l'impression et le tri lexicographique sont des applications plus critiques et plus demandées. Mais la recherche et le développement sur ce thème, génère des outils et des corpus qui peuvent servir à d'autres tâches et d'autres applications. L'intérêt des technologies vocales est mis en évidence dans le contexte du projet Spoken-Web [Kumar *et al.*, 2007] initié par IBM Research qui vise à imiter le Web en proposant l'accès aux informations vocales aux habitants dans les villages en Inde via le téléphone. Le Web est une révolution et représente une source d'informations très importante mais seulement 17% de la population mondiale bénéficie d'un accès à ces ressources[1]. Il y plusieurs raisons qui empêchent les autres 83% de la population de bénéficier de cette nouvelle technologie. Une première cause est le coût très élevé des ordinateurs par rapport au niveau de vie local et le manque d'infrastructure : l'électricité, le réseaux Internet. Deuxièmement, une grande partie de la population mondiale est

1. Source Internet usage statistic : http ://www.internetworldstats.com/stats.htm

encore illettrée et ne sait pas utiliser un ordinateur. Troisièmement, les contenus disponibles sur le Web sont généralement dans une langue étrangère dominante comme l'anglais et ne sont pas adaptés aux besoins quotidiens de ce groupe de population. En revanche, le développement du réseau téléphonique n'a pas rencontré le même handicap que le réseau Internet. Le coût du téléphone, les frais de communication et la complexité d'utilisation sont plus faibles que ceux de l'Internet, ce qui fait que le taux de pénétration du téléphone portable est très élevé dans beaucoup de pays. La vision du projet Spoken-Web est de créer un réseau similaire au Web mais avec des sites vocaux accessibles par le téléphone en utilisant la voix humaine comme vecteur de communication. La mise en place de ce concept a besoin intensivement de technologies vocales très avancées, en particulier la reconnaissance automatique de la parole.

Ce travail de thèse s'inscrit dans les efforts d'informatisation de la langue khmère, langue officielle du Cambodge. Classée comme une langue peu dotée [Berment, 2004], la disponibilité des ressources et des outils de traitement automatique de base pour la langue khmère reste encore très limitée. Pendant que les outils informatiques de base comme le traitement de texte, la saisie simple, l'affichage et l'impression du texte en unicode viennent d'être mis en service par les organisations comme KhmerOS [2] et PAN Localization [3], les outils plus avancés comme le traitement de l'oral, la reconnaissance vocale, la synthèse vocale ne sont pas encore disponibles. Ainsi, dans une perspective de développement informatique plus avancé, la qualité d'un outil informatique doit tenir compte de nouveaux critères qui définissent son utilisabilité : interaction entre la machine et l'homme, facilité et rapidité d'apprentissage. La maîtrise des technologies vocales est nécessaire pour développer des technologies de communication et les télécommunications : la parole reste le premier médium de communication entre les hommes et de ce fait est le signal d'information le plus communément transmis. Une bonne maîtrise des technologies vocales dans la langue du pays est indispensable pour mettre en place des moyens de télécommunications performants et adaptés au pays et à ses ressortissants.

2. www.khmeros.info
3. http ://www.pancambodia.info/

1.1.2 Projet en collaboration

L'Institut de Technologie du Cambodge (ITC) est une école d'ingénieurs de haut niveau qui forme les cadres techniques nécessaires au développement du pays. Dans sa stratégie de développement, l'ITC souhaite initier des activités de recherche, et dès que les conditions le permettront, ouvrir un troisième cycle destiné à la formation par la recherche des spécialistes dans le domaine des sciences de l'ingénieur. Le Département Génie Informatique et Communication a été créé en 1999. La première promotion d'ingénieurs est sortie en 2002. De gros efforts ont été faits pour former un personnel enseignant qualifié : chaque année les meilleurs étudiants de dernière année bénéficient de bourses d'études en Europe pour ensuite devenir enseignants dans le département. Des bourses de Master et de thèse sont attribuées aux jeunes enseignants pour se perfectionner et s'initier à la recherche.

Ce travail de thèse s'incrit dans cette stratégie de développement du Département Génie Informatique et Communication de l'ITC. Grâce à une collaboration avec le laboratoire LIG/GETALP (Grenoble, France) et le centre MICA (Hanoï, Vietnam), le projet de recherche TALK "Traitement Automatique de la Langue Khmère" a démarré en 2004 soutenu par l'AUF (Agence Universitaire pour la Francophonie). L'objectif du projet TALK était de mettre en place au sein du Département Génie Informatique et Communication, un groupe de recherche spécialisé dans le domaine du traitement automatique de la parole en langue khmère, pour assurer le transfert des technologies et pour concevoir des applications d'interaction et de communication parlée.

1.2 Reconnaissance automatique de la parole

1.2.1 Historique

La reconnaissance automatique de la parole (RAP) consiste à extraire, à l'aide d'un ordinateur, l'information lexicale contenue dans un signal de parole. Depuis plus de trois décennies, des recherches intensives dans ce domaine ont été accomplies par de nombreux laboratoires internationaux. Des progrès importants ont

été réalisés grâce au développement d'algorithmes puissants et grâce aux avancées en traitement du signal.

Les fondements de la technologie récente en reconnaissance de la parole ont été élaborés par F. Jelinek et son équipe à IBM dans les années 70 [Jelinek, 1970]. Les premiers travaux (années 80) se sont intéressés aux mots, et ce, pour des applications à vocabulaire réduit. Au début des années 90, les systèmes de reconnaissance automatique de la parole continue gande vocabulaire et indépendants du locuteur ont vu le jour. La technologie s'est développée rapidement et déjà vers le milieu des années 90, une précision raisonnable est atteinte pour une tâche de dictée vocale . Une partie de ce développement a été réalisée dans le cadre de programmes d'évaluation de la DARPA (Defense Advanced Research Projects Agency).

Différents systèmes de reconnaissance de la parole ont été développés, couvrant des domaines variés : reconnaissance de quelques mots clés sur lignes téléphoniques, systèmes de dictée vocale, systèmes de commande et contrôle sur PC, systèmes de compréhension en langage naturel.

1.2.2 Formulation statistique du problème de reconnaissance

Les premiers travaux de reconnaissance de la parole ont essayé d'appliquer des connaissances expertes en production et en perception. De nos jours, les techniques de modélisation statistique apportent les meilleures performances.

La formulation statistique du problème de reconnaissance suppose que la parole est représentée par une séquence de vecteurs acoustiques $O = o_1...o_T$ et que cette séquence encode la suite de mots : $M = m_1...m_K$.

La transcription orthographique de la parole se ramène alors à un problème de décodage où on cherche à trouver la séquence de mots M' tel que :

$$M' = argmax P(M|O) = argmax P(O|M)P(M) \tag{1.1}$$

$P(O/M)$ est déterminée par un modèle acoustique et $P(M)$ par un modèle de langage. L'architecture globale d'un Système de Reconnaissance Automatique de la Parole peut être représentée comme dans le figure 1.1.

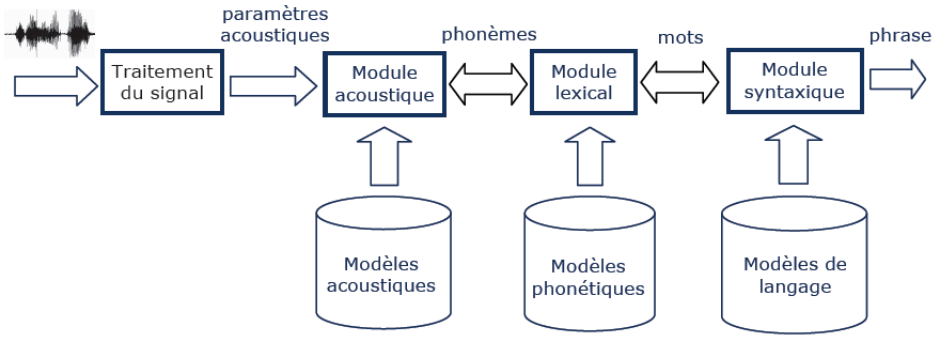


FIGURE 1.1 – *Architecture globale d'un Système de Reconnaissance*

1.2.3 Modélisation du langage

Pour la reconnaissance de la parole continue, la seule information acoustique ne suffit pas pour transcrire correctement les suites de mots. Les modèles de langage représentent une composante majeure du système de reconnaissance automatique de la parole. Ils introduisent les contraintes linguistiques dans le système. Le modèle de langage modélise les contraintes liées à une langue, afin d'estimer la probabilité d'une suite de mots :

$$P(W_1^k) = P(w_1) \prod_{i=2}^{k} P(w_i|h_i) \tag{1.2}$$

où h_i correspond à l'historique du mot w_i.

De nombreux systèmes de seconnaissance automatique de la parole utilisent des modèles de langage n-grammes. Les modèles n-grammes correspondent à une modélisation stochastique du langage où l'historique d'un mot est représentée par les $n-1$ mots qui le précèdent :

$$P(W_1^k) = P(w_1) \prod_{i=2}^{n-1} P(w_i|w_1, ..., w_{i-1}) \prod_{i=n}^{k} P(w_i|w_{i-n+1}, ..., w_{i-1}) \tag{1.3}$$

Les modèles de langage n-grammes sont assez souples car ils permettent de modéliser des phrases grammaticalement incorrectes mais ils n'interdisent pas

non plus de produire des phrases totalement incohérentes. Les modèles les plus couramment utilisés en RAP sont les modèles d'ordre 3 à 5. Dans le cas d'un modèle tri-gramme, l'équation précédente s'écrit :

$$P(W_1^k) = P(w_1)P(w_2|w_1)\prod_{i=3}^{k} P(w_i|w_{i-2}, ..., w_{i-1}) \tag{1.4}$$

Estimation des modèles de langage

L'estimation des paramètres d'un modèle de langage n-grammes s'effectue en deux opérations : une opération de décompte et une opération de redistribution des probabilités. La méthode d'estimation effectue un décompte des suites de mots observés afin d'en extraire une probabilité d'apparition. Le principe est d'estimer toutes probabilités issues d'événements observés, puis de les redistribuer à des événements non vus. Cette seconde étape, qui correspond au lissage, permet d'associer une probabilité non nulle à des événements jamais observés sur le corpus d'apprentissage. Les méthodes de lissage classiques calculent une probabilité non nulle en réduisant la fenêtre d'observation.

Les modèles n-grammes sont donc très dépendants du corpus d'apprentissage, et ont un champ de vision limité à la taille du n-gramme (qui est comprise entre 3 et 5 généralement). Même pour les langues bien dotées, les quantités disponibles de textes pour estimer les probabilités des n-grammes ne sont pas suffisantes pour les n-grammes d'ordre plus élevé. De nombreuses techniques de lissage ont été proposées pour pallier ce problème. Le lissage consiste à prendre une partie de la masse de probabilité des n-grammes observés, pour donner une valeur non-nulle aux probabilités des n-grammes non-observés ou peu observés. L'une des techniques de lissage les plus utilisées est la technique dite de Kneser-Ney [Kneser et Ney, 1995]. Avec cette technique, les probabilités des n-grammes peu observés sont estimées comme avec les autres techniques de lissage, en faisant un repliement (backoff) sur un historique d'ordre moins grand. Pour un trigramme par exemple, le bigramme puis l'unigramme si nécessaire sont utilisés. L'originalité de la technique Kneser-Ney modifiée est de ne pas prendre la même distribution de probabilités pour les ordres plus petits que n. Au lieu de prendre la fréquence de

l'historique d'ordre $n-1$ à savoir h_{i-n+1}^{i-1}, c'est le nombre de contextes différents dans lesquels se produit h_{i-n+1}^{i-1} qui est consulté. L'idée est que si ce nombre est faible alors la probabilité accordée au modèle d'ordre $(n-1)$ doit être petite et ce, même si h_{i-n+1}^{i-1} est fréquent. Ainsi le biais potentiel introduit par la fréquence de l'historique est évité.

Les modèles n-grammes sont extrêmement simples, mais ont prouvé leur efficacité et leur souplesse. Ils se sont imposés dans les systèmes état de l'art bien que diverses alternatives efficaces aient été proposées dans la littérature [Schwenk et Gauvain, 2002] et [Schwenk, 2007], ils continuent d'être quasi systématiquement intégrés aux systèmes de RAP état de l'art.

En pratique, pour construire les modèles de langage, nous avons utilisé la librairie SRILM [Stolcke, 2002]. Il existe cependant d'autres boîtes à outils, comme par exemple CMU SLM, pour Carnegie Mellon Statistical Language Modeling-Toolkit.

Évaluation des modèles de langage

La qualité d'un modèle de langage dépend de sa capacité à influencer le système de reconnaissance automatique de la parole afin d'en augmenter la performance. Une question primordiale est de savoir comment deux modèles de langage peuvent être comparés en termes de performances dans un système de reconnaissance. La façon correcte de procéder consiste à incorporer chaque modèle dans un système complet et d'évaluer quelle est la meilleure transcription en sortie du système. Cette méthode permet d'évaluer concrètement la performance d'un modèle de langage mais nécessite de disposer d'un système complet.

La mesure la plus couramment utilisée consiste à estimer la perplexité de chacun des modèles. La perplexité d'un modèle de langage correspond à sa capacité de prédiction. Plus la valeur de perplexité est petite, plus le modèle de langage possède des capacités de prédiction. La perplexité s'estime sur le corpus d'apprentissage pour définir si les modèles choisis modélisent correctement le corpus. Elle est calculée sur un corpus de test ou de développement, pour estimer le degré de généralisation du modèle. Cependant, bien que la perplexité permette d'estimer

la capacité de représentation d'un modèle de langage, elle n'est pas systématiquement corrélée avec la qualité du décodage. Pour des modèles n-grammes, la perplexité se définit ainsi :

$$PP = 2^{-\frac{1}{N}\sum_{t=1}^{n}\log_2 P(w_t|h)} \tag{1.5}$$

où $P(w_t|h)$ est la probabilité associée au n-gramme $(w_t|h)$.

Deux remarques importantes sont à prendre en considération lorsque l'on compare des modèles de langage :

- une réduction de perplexité n'implique pas toujours un gain de performances d'un système de reconnaissance,
- en général, la perplexité de deux modèles n'est comparable que s'ils utilisent le même vocabulaire. Sinon, il faut utiliser une perplexité normalisée qui simule un nombre de mots identique.

Bien que des modèles de langage avec des mesures de perplexité qui diminuent tendent à améliorer les performances d'un système de reconnaissance, il existe dans la littérature des études qui reportent des diminutions importantes de perplexité n'ayant peu ou pas apporté de gain de performance [S.C. Martin et Ney, 1997] et [R. Iyer et Meteer., 1997].

1.2.4 Modélisation acoustique

Vecteurs acoustiques

Le signal de parole ne peut être exploité directement. En effet, le signal contient de nombreux autres éléments que le message linguistique : des informations liées au locuteur, aux conditions d'enregistrement, etc. Toutes ces informations ne sont pas nécessaires lors du décodage de parole et rajoutent même du bruit. De plus, la variabilité et la redondance du signal de la parole le rendent difficilement exploitable tel quel. Il est donc nécessaire d'en extraire uniquement les paramètres qui seront dépendants du message linguistique.

Généralement, ces paramètres sont estimés via des fenêtres glissantes sur le signal. Cette analyse par fenêtrage permet d'estimer le signal sur une portion ju-

gée stationnaire : généralement 10 à 30 ms en limitant les effets de bord et les discontinuités du signal via une fenêtre de Hamming. La majorité des paramètres représentent le spectre fréquentiel et son évolution sur une fenêtre de taille donnée. Les techniques de paramétrisation les plus utilisées sont : PLP (Perceptual Linear Prediction : domaine spectral) [Hermansky et Cox, 1991], LPCC (Linear Prediction Cepstral Coefficients : domaine temporel) [Markel et JR., 1976] et MFCC (Mel Frequency Cepstral Coefficients : domaine cepstral) .

Modèles de Markov Cachés (MMC)

Le signal acoustique de parole est modélisable par un ensemble réduit d'unités acoustiques, qui peuvent être considérées comme des sons élémentaires de la langue.

Classiquement, l'unité choisie est le phonème : un mot étant formé par la concaténation de phonèmes. Des unités plus précises peuvent être employées comme les syllabes, les di-syllabes, les phonèmes en contexte, permettant ainsi de rendre la modélisation plus fine, mais cette amélioration théorique est limitée dans la pratique par la complexité induite et les problèmes d'estimation. Un compromis souvent employé est l'utilisation de phonèmes contextuels avec partage d'états. Le signal de parole peut être assimilé à une succession d'unités. Dans le cadre des systèmes de RAP Markoviens, les unités acoustiques sont modélisées par des Modèles de Markov Cachés (MMC), typiquement des MMC gauche-droite à trois états.

A chaque état du modèle de Markov est associée une distribution de probabilité modélisant la génération des vecteurs acoustiques via cet état. Un MMC est caractérisé par plusieurs paramètres :

- son nombre d'états N,
- l'ensemble des états du modèle $e = (e_i)_{(1 \leq i \leq N)}$,
- une matrice de transition entre les états : $A = (a_{ij})_{(1 \leq i,j \leq N)}$ de taille $N \times N$,
- la probabilité d'occupation d'un état à l'instant initial : $(\pi_i)_{(1 \leq i \leq N)}$: $\pi_i = P(e_1 = e_i)$,
- la densité de probabilité d'observation associée à l'état e_i : b_i.

Un MMC est donc représenté par un ensemble de paramètres :

$$MMC = (N, A, \{\pi\}, \{b\}) \tag{1.6}$$

Les paramètres du MMC sont estimés empiriquement sur de grands corpus de parole annotés.

Apprentissage des modèles

Les paramètres des MMC qui comprennent les probabilités de transition entre états, les moyennes, les variances et les poids des mélanges de gaussiennes, sont estimés sur des alignements de transcriptions de données audio d'apprentissage. Au cours de l'opération appelée alignement, le signal audio est découpé en tronçons, associés chacun à une seule unité acoustique (phone par exemple).

Pour réaliser les premiers alignements, plusieurs techniques sont utilisées, soit des techniques de *flat start*, soit des techniques de *bootstrap* (amorçage) qui utilisent des modèles pré-existants d'une ou plusieurs autres langues.

Approches flat start

L'approche communément appelée *flat start*, est la technique la plus simple pour initialiser les paramètres des MMC. Elle consiste à mettre à zéro les probabilités de transition que l'on veut interdire, par exemple les transitions d'un état vers un état antérieur (modèles gauche-droite). Toutes les autres probabilités de transition entre états sont considérées comme équiprobables. Pour les probabilités d'observation, les moyennes et les variances des gaussiennes sont toutes initialisées aux mêmes valeurs, à savoir la moyenne et la variance estimées sur toutes les données d'apprentissage.

Approches bootstrap

Deux approches principales dites de *bootstrap* (amorçage) existent pour initialiser les modèles acoustiques [Schultz et Waibel, 2001]. La première approche consiste à choisir des modèles acoustiques de systèmes de reconnaissance existants pour segmenter les données transcrites manuellement dans la langue cible. Cette méthode est fréquemment utilisée mais est rarement explicitement men-

tionnée dans la littérature. La deuxième approche consiste à prendre un jeu de modèles acoustiques multilingues génériques qui couvrent un grand nombre de phonèmes [Schultz, 2002]. Cette dernière technique peut être utile pour réaliser un système pour une langue avec très peu de données, typiquement moins de 10h de transcriptions audio. Chaque segment de parole est aligné soit de manière itérative avec l'algorithme de Baum-Welch [Baum *et al.*, 1970], qui prend en compte tous les chemins qui passent par un état, soit uniquement avec la meilleure séquence d'états possible (alignement de type Viterbi). Après l'alignement, les paramètres des MMC sont estimés à l'aide d'une procédure EM (Expectation/-Maximization) en partant d'une seule gaussienne par état, qui est divisée jusqu'à obtenir le nombre maximal de gaussiennes voulu, pris typiquement entre 8 et 128 gaussiennes [JL. Gauvain et Adda, 2002]. Cette technique d'amorçage a été utilisée avec succès dans [Le, 2006] pour le développement d'un système de RAP pour la langue vietnamienne.

1.2.5 Dictionnaire de prononciation

Le dictionnaire de prononciation fournit le lien entre les séquences des unités acoustiques et les mots représentés dans le modèle de langage. Alors que les corpus de texte et de parole peuvent être collectés, le dictionnaire de prononciation n'est généralement pas directement disponible. Bien qu'un dictionnaire de prononciation créé manuellement donne une bonne performance, la tâche est très lourde à réaliser et demande des connaissances approfondies sur la langue en question. La littérature propose des approches qui permettent de générer automatiquement le dictionnaire de prononciation. L'approche, simple et totalement automatique, qui utilise des graphèmes comme unités de modélisation a été validée dans [Billa et al, 2002] et [Bisani et Ney, 2003].

Une autre approche de génération automatique de dictionnaire de prononciation consiste à utiliser des règles de conversion graphème-phonème. Cette construction nécessite une bonne connaissance de la langue et de ses règles de phonétisation, qui par ailleurs ne doivent pas contenir trop d'exceptions. Cependant, ce type d'approche est assez coûteux en temps (écriture d'un analyseur phonétique), mais donnera des dictionnaires de prononciation de qualité très correcte pouvant

ensuite être révisés manuellement relativement rapidement. Il existe également certaines approches utilisant un système de reconnaissance phonémique appliqué sur des enregistrements des mots à phonétiser, permettant un premier étiquetage automatique en phonèmes d'une liste de mots, qui peut être alors révisé par un opérateur humain.

Il est important de noter que les performances du système de reconnaissance sont directement liées au taux de mots hors vocabulaire. La taille et la qualité (couverture) de dictionnaire joue ainsi un rôle très important dans les système de reconnaissance de la parole.

1.2.6 Décodage

L'objectif du décodage est de trouver la séquence de mots la plus probable sachant le dictionnaire et les modèles acoustiques et de langage. En pratique, il s'agit de trouver la suite d'états la plus probable dans un treillis de mots (espace de recherche) où chaque nœud représente un état de phone donné à un temps t. Pour ce faire, deux algorithmes sont fréquemment utilisés : l'algorithme de Viterbi et l'algorithme A^* qui est asynchrone.

Vue la taille de l'espace de recherche, la détermination du meilleur chemin peut devenir compliquée. Une approche multi-passes peut être utilisée pour réduire la complexité du décodage. Par exemple, pour la première passe on peut utiliser un bigramme et des modèles acoustiques simples et dans la seconde un trigramme et des modèles acoustiques plus fins. L'information entre les passes est transmise via un treillis de mots ou les N meilleures hypothèses. Le treillis est un graphe où les nœuds correspondent à des instants et les arcs correspondent aux hypothèses de mots. Les N meilleures hypothèses correspondent à une liste des meilleures séquences de mots et de leurs scores respectifs.

1.2.7 Evaluation

Les systèmes de reconnaissance de la parole sont évalués en termes de taux de mots erronés (ou WER pour Word Error Rate).

$$WER = \frac{S + D + I}{N} \times 100 \tag{1.7}$$

où S correspond aux substitutions, D aux suppressions (ou élisons), I aux insertions et N est le nombre de termes dans la référence.

Ce taux est calculé après alignement dynamique de l'hypothèse du décodeur avec une transcription de référence, à l'aide d'un calcul de distance d'édition minimale entre mots. Le résultat sera le nombre minimal d'insertions, de substitutions et d'élisions de mots, pour pouvoir faire correspondre les séquences de mots de l'hypothèse et de la référence. D'après sa définition, le WER peut être supérieur à 100% à cause des insertions.

La boîte à outils SCTK4 (Scoring Toolkit) du National Institute of Standards and Technologies (NIST) fournit le programme *sclite* pour aligner les hypothèses et les références, calculer les WER et faire des analyses fines des erreurs. Cet outil peut fournir des informations très utiles comme les mots les plus substitués, insérés ou élidés ; des taux d'erreurs par locuteur peuvent être également obtenus (si les segments possèdent une étiquette de locuteur).

1.3 Problématique de la thèse

1.3.1 Reconnaissance automatique de la parole pour des langues peu dotées

En laissant de côté les problèmes liés aux critères qui définissent une langue et en particulier la délicate distinction entre langue et dialecte, le nombre de langues dans le monde est en général estimé à 6000 [Amorrortu *et al.*, 2004]. La distribution géographique des langues est très inégale selon les continents. Pour un total estimé à 6000 langues, presque deux tiers proviennent des continents africains et asiatiques (un tiers pour chaque continent), alors que seulement 3% sont des langues européennes. Enfin, les langues des continents américains et de la zone pacifique représentent respectivement 15% et 18% des langues du monde [Grimes, 2000]. Selon [Crystal, 2000], 82% des langues du monde ont moins de

100,000 locuteurs, et 56% moins de 10,000 locuteurs. Un faible nombre de locuteurs n'est pas le facteur unique déterminant le rayonnement d'une langue, néanmoins ces pourcentages montrent qu'une majorité de langues risque de disparaître au profit d'autres langues dominantes [Hagège, 2002]. Pour des institutions comme l'UNESCO, le développement de ressources et d'outils numériques pour de telles langues est une étape nécessaire pour tenter de préserver une diversité linguistique menacée.

Les technologies de reconnaissance automatique de la parole sont réservées, pour l'instant, à un très petit nombre de langues. Il s'agit des langues des pays dits développés, ou de langues qui suscitent un intérêt économique ou politique. Les langues minoritaires ou les langues venant de pays en voie de développement sont moins abordées par la communauté du traitement automatique de la langue naturelle. Mais au cours des dernières années, les langues minoritaires et les langues peu dotées ont attiré une attention croissante dans la communauté du traitement automatique de la langue naturelle. Des projets qui visent à la revitalisation, la standardisation et à la normalisation linguistique ont été lancés pour favoriser l'usage de ces langues et pour contribuer à leur survie. L'augmentation du nombre de pages sur l'Internet en langues minoritaires en est une illustration.

Le développement d'un système de reconnaissance automatique de la parole continue à grand vocabulaire dans une nouvelle langue nécessite de rassembler une grande quantité de corpus de parole, contenant des signaux de parole pour l'apprentissage des modèles acoustiques du système. De tels corpus et systèmes sont désormais disponibles pour la plupart des langues occidentales comme l'anglais, le français, l'espagnol, et pour quelques langues asiatiques comme le chinois, le japonais, le coréen ainsi que pour l'arabe. Pour beaucoup d'autres langues, ces ressources ne sont pas encore disponibles ou elles sont disponibles en quantité très limitée. De nombreux termes plus ou moins équivalents existent dans la littérature pour désigner les langues, qui sont pour certaines, parlées par des millions de personnes, mais qui ne disposent pas d'une activité et de ressources numériques importantes. Une qualification semble être plus utilisée, il s'agit de l'expression *langues peu dotées* en français, et *under-resourced languages* en anglais. L'informatisation des langues peu dotées a été étudiée dans la thèse de V. Berment

intitulée "Méthodes pour informatiser des langues et des groupes de langues peu dotées" [Berment, 2004]. Dans la thèse "Reconnaissance automatique de la parole pour des langues peu dotées" [Le, 2006], les langues bien dotées qui sont les quelques langues qui possèdent des ressources en quantité importante, sont opposées aux langues peu dotées qui disposent de peu de ressources linguistiques servant à élaborer les systèmes de reconnaissance ou de TALN.

Les progrès considérables qui ont été réalisés depuis les années 1990, ont permis l'émergence de recherches et de nombreux projets sur l'adaptation rapide des systèmes à des langues qui ne disposeraient pas, a priori, de quantités de données suffisantes. Les trois types de données nécessaires à l'élaboration d'un système de reconnaissance de la parole actuel sont de grands corpus de textes (typiquement entre quelques dizaines et quelques centaines de millions de mots), un corpus audio de parole transcrite (typiquement entre quelques dizaines et quelques centaines d'heures), ainsi qu'un lexique de mots donnés avec leur prononciation et des variantes éventuelles. Le projet actuel SPICE (Speech Processing : Interactive Creation and Evaluation Toolkit), par exemple, de l'université Carnegie Mellon, s'est intéressé entre autres à l'afrikaans, au bulgare, au vietnamien, à l'hindi, au konkani, au telugu et au turc [Schultz *et al.*, 2007]. Des projets plus anciens visaient à collecter des ressources pour des langues peu dotées, comme par exemple le projet Babel sur cinq langues est-européennes (bulgare, estonien, hongrois, roumain et polonais) [Roach *et al.*, 1996].

Des travaux récents ont principalement cherché à limiter le temps et les moyens nécessaires à la constitution des corpus d'apprentissage audio et textes, et ont mis l'accent sur la modélisation acoustique en étudiant la portabilité rapide des modèles acoustiques d'une langue (ou multilingues) vers une autre. Dans le projet mentionné SPICE [Schultz *et al.*, 2007], des modèles multilingues sont utilisés pour initialiser les modèles acoustiques (technique dite de *bootstrap*, présentée dans la section 1.2.4), et sont entraînés de manière itérative pour devenir dépendants de la langue cible. Dans [Le, 2006], des mesures de proximité entre modèles acoustiques de phones sont proposées pour sélectionner les meilleurs modèles d'initialisation multilingues. Ces travaux ont montré qu'avec un petit corpus audio de parole transcrite (quelques heures de données) collecté pour une nouvelle langue, un

modèle acoustique d'une performance acceptable peut être construit à partir des modèles acoustiques multilingues pré-existants.

En ce qui concerne la création d'un lexique de prononciation, l'approche la plus couramment utilisée lorsque peu de connaissances linguistiques sont accessibles sur la langue étudiée pour générer des prononciations, est d'associer un phone à chaque graphème. Cette approche est appelée modélisation acoustique à base de graphèmes. Elle a le double avantage de permettre la génération d'un lexique très simplement et très rapidement. Cette méthode a été étudiée pour différentes langues peu dotées ou moyennement dotées : arabe [Abdou, 2004], russe [Stücker et Schultz, 2004], vietnamien et khmer [Le, 2006], mais également pour des langues bien dotées : allemand, anglais, espagnol [Killer *et al.*, 2003], allemand, anglais, italien, néerlandais [Kanthak et Ney, 2002].

Des efforts sont également concentrés sur le développement d'outils destinés à collecter des données afin de rendre les langues concernées un peu mieux dotées. Le recueil de signaux de parole est une tâche lourde. Les campagnes d'enregistrement mobilisent d'importantes ressources humaines pour guider ou assister les locuteurs dans leur tâche de diction, pour organiser l'enregistrement, pour préparer les scénarios et les données, etc. Les outils comme EMACOP (Environnement Multimédia pour l'Acquisition et la gestion de Corpus Parole) [Vaufreydaz *et al.*, 1998] dévelopé au laboratoire LIG/GETALP (ex CLIPS) permettent d'organiser la campagne d'enregistrement sur le réseau, en mode client-serveur, de plusieurs locuteurs en même temps. Ce type de logiciel permet d'accélérer le recueil de signaux de parole pour la modélisation acoustique. En effet, le manque de données acoustiques peut être en partie résolu par une méthodologie efficace de collecte de données ou d'adaptation de modèles acoustiques existants (méthodes translingues).

Concernant le recueil de données textuelles en grande quantité, une approche intéressante consiste à « aspirer » un grand nombre de sites Web dans la langue donnée et à filtrer les données récupérées pour les rendre exploitables. Une telle approche a déjà été validée pour une langue bien dotée telle que le français [Vaufreydaz, 2002]. Les problèmes spécifiques pour les langues peu dotées concernent le nombre de sites Web qui peut être peu important, la vitesse de transmission, et la

faible qualité des documents qui nécessite plus d'outils de traitement. Dans [Pelligrini, 2008], les expérimentations montrent que le manque de textes est le facteur le plus limitant lors de l'élaboration d'un système de RAP pour une langue peu dotée dans la mesure où il n'est pas possible de remédier, de quelque façon que ce soit, à l'absence de textes disponibles, due en particulier à un très petit nombre de sites Internet dans la langue étudiée.

Plusieurs travaux ont tenté de pallier les problèmes liés à la modélisation du langage pour les langues peu dotées, à savoir des taux de mots inconnus élevés et des modèles de langage peu fiables à cause du manque de données d'apprentissage. Dans la thèse « Transcription automatique de langues peu dotées » [Pelligrini, 2008], les recherches portent principalement sur la modélisation lexicale, et en particulier sur la sélection des unités lexicales, mots et sous-unités, utilisés par les systèmes de reconnaissance automatique pour les langues comme l'amharique et le turc. Des stratégies similaires ont été utilisées également dans la thèse « Sauvegarde du patrimoine oral africain : conception de système de transcription automatique de langues peu dotées pour l'indexation des archives audio » [Nimaan, 2007] pour modéliser une autre langue africaine : le somali.

Les systèmes de reconnaissance automatique de la parole sont pour la plupart des systèmes à vocabulaire fermé, c'est-à-dire que seuls les mots du lexique de prononciations peuvent être reconnus. Ainsi, le manque de textes fait que les taux de mots hors-vocabulaire peuvent être très élevés, typiquement au dessus de 5%. D'autre part, les modèles de langage sont estimés sur très peu d'occurrences des différents n-grammes, et sont pour cette raison peu fiables (nombreux replis sur des n-grammes d'ordre inférieur). Ce phénomène de taux de mots hors-vocabulaire élevé est encore plus prononcé pour les langues avec un système d'écriture sans séparation explicite entre les mots ou les langues ayant une grande richesse au niveau de la morphologie. Nous allons présenter dans la section suivante ces problèmes, en particulier les problèmes liés aux langues ayant un système d'écriture sans séparation explicite entre les mots.

1.3.2 Langues non segmentées

La description très générique de méthode de modélisation statistique du langage dans la section 1.2.3 pourrait nous amener à conclure que ces techniques de modélisation développées initialement pour les langues comme le français ou l'anglais peuvent être appliquées sans spécialisation à n'importe quelles autres langues. Théoriquement, il est vrai que l'estimation de modèles de langage n-gramme a besoin tout simplement d'une quantité suffisante de corpus de texte de la langue en question pour calculer les probabilités des séquences des mots (3-grammes des mots par exemple). Le *mot* qui est généralement l'unité de base dans la modélisation statistique du langage est naturellement définie comme une séquence de caractères séparée par les espaces pour les langues comme le français ou l'anglais. Mais pour beaucoup d'autres langues comme le chinois ou les langues dans notre contexte d'étude : khmer, vietnamien, laotien et thai , cette définition n'est pas aussi naturelle et adéquate.

Tandis que le mot est généralement l'unité de base dans la modélisation statistique du langage, l'identification de mots dans un texte n'est pas une tâche simple même pour les langues qui séparent les mots par un caractère (un espace en général). Pour les langues dites non segmentées qui possèdent un système d'écriture sans séparation évidente entre les mots, les n-grammes de mots sont estimés à partir de corpus d'apprentissage segmentés en mots. La segmentation automatique n'est pas une tâche triviale et introduit des erreurs à cause des ambigüités de la langue naturelle et la présence de mots inconnus dans le texte à segmenter.

Alors que le manque de données textuelles a un impact sur la performance des modèles de langage, les erreurs introduites par la segmentation automatique peuvent rendre ces données encore moins exploitables. Une alternative possible consiste à calculer les probabilités à partir d'unités sous-lexicales. Parmi les travaux existants qui utilisent des unités sous-lexicales pour la modélisation du langage, nous pouvons citer [Kurimo *et al.*, 2006], [Abdillahi *et al.*, 2006] et [Afify *et al.*, 2006] qui utilisent les morphèmes respectivement pour la modélisation de l'arabe, du finnois, et du somali. Pour une langue non-segmentée comme le japonais, le caractère (idéogramme) est utilisé dans [Denoual et Lepage, 2006].

1.3.3 Sujet de thèse

Ce travail de thèse fait partie du projet de développement des activités de recherche dans le Département Génie Informatique et Communication (GIC) de L'Institut de Technologie du Cambodge (ITC). Plusieurs résultats sont attendus à la fin de cette thèse. Premièrement, du point de vue du développement de l'activité de recherche au GIC, le choix du sujet de thèse dans le domaine du traitement automatique de la parole en langue khmère, langue officielle du pays, est un créneau à saisir pour afficher la spécificité de notre future équipe de recherche et pour contribuer de manière concrète au développement informatique de la langue khmère.

L'aspect opérationnel de ce travail de thèse consiste dans un premier temps à constituer les ressources linguistiques nécessaires : le corpus de parole, le corpus de texte, le dictionnaire de prononciation et à développer les outils de base pour traiter ces données. Ensuite, il consiste à développer un système de reconnaissance automatique de la parole de l'état de l'art pour la langue khmère (broadcast news) à partir de ces ressources. Cette contribution permet de doter la langue khmère de ressources linguistiques numériques qui sont indispensables pour développer des outils de traitement automatique de la langue et pour poursuivre des travaux de recherche plus avancés dans le domaine. Ce travail permet également de revisiter les méthodes et les outils de l'état de l'art proposés pour la collecte rapide de données et le développement rapide d'un système de RAP pour une nouvelle langue peu dotée.

La contribution scientifique de cette thèse concerne la spécificité des langues traitées dans notre contexte d'étude : les langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots qui nécessite la segmentation automatique en mots pour rendre la modélisation de langage n-gramme applicable. Pour tenter de pallier les problèmes, à savoir des taux de mots inconnus élevés et des modèles de langage peu fiables à cause de manque de données textuelles et les erreurs de segmentations, nous avons axé nos recherches principalement sur la modélisation lexicale, et en particulier sur le choix des unités lexicales efficaces, mots et sous-unités, utilisées par les systèmes de reconnaissance. Le problème de la faible quantité de données textuelles implique de réfléchir à des techniques

de modélisation lexicale et sous-lexicale (mots, groupe de caractères, caractères), permettant ainsi de réduire la taille du vocabulaire de l'application, tout en essayant d'exploiter au mieux les données. Nous proposons de traiter ce problème en exploitant plusieurs vues sur les données textuelles dans la modélisation du langage. Nous travaillons à la fois au niveau du modèle de langage en créant des vocabulaires hybrides à partir d'unités lexicales et sous-lexicales, et au niveau du système en combinant des sorties de différents systèmes de RAP pour décoder une meilleure hypothèse. Ces méthodes de modélisation multi-unités sont appliquées et validées dans les systèmes de RAP pour les langues suivantes : le khmer, le vietnamien, le thai et le laotien.

1.4 Conclusion

Dans ce chapitre, nous avons présenté le contexte et la motivation de notre thèse. Nous avons par la suite abordé le principe de la reconnaissance de la parole par l'approche statistique, et décrit les différentes composantes d'un système de RAP standard : modèles acoustiques, modèles de langage, dictionnaire de prononciation, et présenté très brièvement le principe des décodeurs. En fin de chapitre, nous avons présenté la problématique principale du travail de thèse : les problèmes liés au développement d'un système de RAP pour les langues peu dotées et ayant un système d'écriture non segmenté.

Chapitre 2

Reconnaissance automatique de la parole en langue khmère

2.1 Introduction

Le développement d'un système de Reconnaissance Automatique de la Parole continue grand vocabulaire (RAP) pour une langue peu dotée comme le khmer est une tâche qui conduit à trois challenges : (1) le manque de ressources linguistiques sous forme numérique (corpus de texte et de parole), (2) le système d'écriture sans séparation explicite entre les mots, qui nécessite une segmentation automatique pour que la modélisation statistique du langage soit applicable et (3) les caractéristiques acoustiques et phonologiques de la langue qui sont encore assez peu étudiées.

Ce chapitre présente une vue d'ensemble concernant le développement d'un système RAP pour le khmer. Nous décrivons tout d'abord notre méthode de collecte de données linguistiques pour le développement rapide d'un nouveau système de RAP pour une langue peu dotée. Le problème du manque de données textuelles et de la présence d'erreurs lors de la segmentation en mots est abordé. Nous traitons ce problème en exploitant plusieurs vues sur les données textuelles dans la modélisation du langage. Pour la modélisation acoustique, nous présentons et comparons des méthodes de génération automatique de dictionnaires de prononciation

à base de graphèmes et à base de règles de conversion graphème-phonèmes pour le khmer. Enfin, des expérimentations sont menées pour tester et comparer les approches proposées.

2.2 Présentation de la langue khmère

Le khmer est la langue officielle du Cambodge parlée par plus de 16 millions d'habitants. C'est une langue appartenant au groupe des langues môn-khmères de la famille des langues austro-asiatiques. Le khmer est principalement parlé au Cambodge et dans certaines régions de la Thaïlande (les khmers surin) et du Viêt-Nam (delta du Mékong, les khmers Krom).

Le système d'écriture du khmer est alphasyllabique. L'alphabet khmer moderne possède 33 consonnes, 23 voyelles dépendantes et 14 voyelles indépendantes, sans compter les ligatures, les diacritiques et la ponctuation. Chaque consonne appartient à l'une des deux séries. Si une voyelle est associée à la première série (ou 1^{er} registre) elle produit un certain son et si elle est associée à la deuxième série (ou 2^{eme} registre) elle produit un autre son. Ainsi les voyelles ont deux prononciations possibles.

L'écriture du khmer est sans séparation entre les mots. On place un espace entre des groupes de mots pour marquer une pause (équivalent à une virgule ou un point-virgule en français). Ceci pose des problèmes en traitement automatique qui devront être résolus. A titre d'exemple, la figure 2.1 présente une phrase khmère et la segmentation en mots de cette phrase.

Phrase en khmer : ព្រះពុទ្ធជាព្រះបរមគ្រូនៃយើង
Segmentation en mot : ព្រះពុទ្ធ ជា ព្រះបរមគ្រូ នៃ យើង
Traduction : Le bouddha est notre maître suprême

FIGURE 2.1 – *Exemple d'une phrase en khmer.*

Le khmer est une langue atonale, contrairement à ses voisines thaïes, laotiennes ou vietnamiennes. Cependant, le khmer possède comme ses cousines austro-asiatiques

plusieurs registres vocaliques : les voyelles peuvent être allongées (dites voyelles longues), raccourcies (dites voyelles brèves), diphtonguées, reposer sur des consonnes aspirées ou non aspirées, ce qui en modifie complètement le sens.

Cette particularité fait du khmer l'un des plus riches systèmes vocaliques au monde. Au niveau de la phonétique, il y a 29 phonèmes vocaliques : 10 phonèmes longs, 7 phonèmes brefs et 12 diphtongues. Une analyse phonétique détaillée de la langue khmère a été effectuée dans le cadre de notre projet TALK [Seng et Sam, 2005]. La langue khmère se compose de monosyllabes et polysyllabes. Les polysyllabes sont courantes surtout dans leur forme bi- et tri-syllabique. La structure syllabique générale du khmer se retrouve sous la forme suivante :

$$C_1(C_2)(C_3)V(C_4) \tag{2.1}$$

avec C_i consonne et V voyelle [Huffman, 1970]. Nous notons que la consonne initiale et la voyelle du noyau sont obligatoires et les autres consonnes sont facultatives. Cependant, pour les voyelles courtes, la consonne finale est obligatoire. La consonne C_3 est rarement présente dans les mots khmers.

La langue khmère a bénéficié d'un grand nombre d'emprunts au sanskrit, au pali et au français ainsi que, dans les milieux urbanisés, au chinois. La plus grande partie du vocabulaire administratif, militaire et littéraire est empruntée au sanskrit. Avec l'introduction du bouddhisme au début de XV^{me} siècle, le pali devient une source d'emprunts lexicaux très importante. Plus récemment, du fait de l'occupation française en Indochine, certains mots ont été empruntés au français, mais orthographiés en khmer.

2.2.1 Le khmer, une langue peu dotée ?

En utilisant la méthode proposée dans [Berment, 2004], nous pouvons évaluer de manière quantitative le degré d'informatisation d'une langue en utilisant le protocole suivant : pour chaque service ou ressource informatique de la langue en question, un groupe d'utilisateurs représentatifs des locuteurs de la langue attribue un niveau de criticité et une note. La moyenne pondérée des notes reflète

	Services / ressources	Criticité (/10)	Note (/20)	Note pondérée (Criticité x Note)
Traitement du texte				
	Saisie simple	10	16	160
	Visualisation / impression	10	14	140
	Recherche et remplacement	8	12	96
	Sélection du texte	6	12	72
	Tri lexicographique	5	0	0
	Correction orthographique	2	0	0
	Correction grammaticale	0	0	0
	Correction stylistique	0	0	0
Traitement de l'oral				
	Synthèse vocale	5	0	0
	Reconnaissance de la parole	5	0	0
Traduction				
	Traduction automatisée	8	4	32
ROC				
	Reconnaissance optique de caractères	9	0	0
Ressources				
	Dictionnaire bilingue	10	4	40
	Dictionnaire d'usage	10	0	0
Total		88		540
Moyenne				**6,14 / 20**

TABLE 2.1 – *Tableau d'évaluation du niveau d'informatisation pour le khmer*

leur satisfaction globale des outils ou ressources informatiques disponibles pour cette langue. La criticité est une mesure de l'importance relative d'un service pour un groupe d'évaluation donné. Pour une langue, il y a 5 groupes de services ou ressources à évaluer tels que : le traitement de texte, le traitement de l'oral, la traduction automatique, la reconnaissance optique de caractère et les ressources linguistiques. Le tableau 2.1 est le résultat d'évaluation du degré d'informatisation de la langue khmère effectué par V. BERMENT dans sa thèse [Berment, 2004] soutenu en 2004. La moyenne de 6,14/20 montre que le khmer est une langue très peu dotée.

2.2.2 Traitement automatique de la langue khmère

Le khmer est classé comme une langue peu dotée à cause de la limite des outils informatiques liés à son traitement automatique. Avant l'arrivée de Unicode, le caractère khmer était codé de manière non standardisée, ce qui entraînait des échanges sur Internet difficiles et limités. La multiplicité des encodages non standardisés existant et plus généralement l'absence de norme sont donc une première

difficulté pratique pour le traitement de textes khmers, les textes étant associés à des polices particulières. Aujourd'hui encore, lorsqu'un courrier électronique en khmer est envoyé, l'auteur indique quelle police utiliser et joint parfois la police à son envoi pour s'assurer que le courrier peut être affiché correctement. L'encodage des caractères khmer devient standardisé avec Unicode mais son usage reste encore très limité comme la majorité des systèmes ne supportent pas encore la dernière version de Unicode contenant l'encodage des caractère khmers.

Le système d'exploitation le plus utilisé de type Microsoft Windows n'est pas encore disponible en version khmère. La localisation des logiciels doit d'abord passer par la traduction des mots techniques en informatique vers la langues khmère (problème de terminologie). Le projet KhmerOS [1], mené par Open Institut [2], a contribué à localiser le système d'exploitation Opensource (SUSE Linux version 10) et plusieurs autres applications Opensource : bureautique (Openoffice), navigateur Internet et courrier électronique (Firefox et Thunderbird) en khmer.

L'Internet est la principale source de collecte de textes pour constituer des corpus de taille importante. La présence d'une langue sur Internet est un bon indicateur de son niveau d'informatisation. Le fait qu'une langue soit déclarée langue officielle, ou qu'elle soit parlée par un grand nombre de locuteurs, n'implique pas forcément une présence importante de cette langue sur Internet. Le nombre de pages web dans cette langue et le nombre d'utilisateurs peuvent être utilisé pour mesurer la présence d'une langue sur Internet. Pour estimer le nombre de pages web contenant du texte khmer sur l'Internet, nous utilisons le moteur de recherche google pour recherche plusieurs mots khmers les plus courants de la langue et nous regardons le nombre de résultats donnés par ce moteur. En utilisant cette méthode, le nombre de pages web contenant le texte khmer sur Internet en 2009 est estimé à environ 430,000 pages comparé à 220,000 en 2006 (pages en Unicode seulement). Le dynamisme d'une langue peut être observé par l'évolution de nombre des utilisateurs Internet du pays. Selon le site Internet World Stats [3], le nombre des utilisateur Internet au Cambodge en 2009 est estimé à 74,000 personnes pour une population de 14 millions (0,5%) comparé à seulement

1. www.khmeros.info
2. www.open.org.kh
3. Internet World Stats : www.internetworldstats.com

6,000 personnes en 2000, soit une croissance de 1,133.3%. Cette tendance de forte croissance du nombre d'utilisateurs internet peut être également observée dans beaucoup autres pays d'envoi de développement.

Il existe quelques travaux récents sur les traitements plus avancés de la langue khmère. Un correcteur orthographique du texte khmer est proposé dans [Puthick, 2005]. Une phase de segmentation est nécessaire avant de vérifier l'orthographe de chaque mot dans le texte khmer. La segmentation est faite en utilisant un modèle HMM pour modéliser la frontière entre les mots. A notre connaissance, le premier système de reconnaissance automatique de la parole en langue khmère (parole lue) est développé dans le cadre de la thèse de V-B LE [Le, 2006]. Dans le travail de LE, les méthodes de collection rapide des données sur Internet et la modélisation acoustique à base de graphème ont été utilisées. Une analyse acoustique et phonologique de la langue khmère a été effectuée dans le cadre du projet TALK [Seng et Sam, 2005] à l'Institut de Technologie du Cambodge.

2.3 Recueil de ressources linguistiques

Pour élaborer notre système de reconnaissance automatique de la parole en langue khmère, trois types de données ont été utilisés : un corpus de parole issue d'émissions de radio transcrites manuellement, un corpus de textes issus de sites Web de journaux en ligne et un vocabulaire.

2.3.1 Corpus de parole

Pour obtenir rapidement un corpus de parole khmère, nous avons enregistré des émissions de type bulletin d'information des chaînes de radio locales à Phnom Penh, Cambodge, en coopération avec l'Institut de Technologie du Cambodge (ITC). Une campagne de transcription manuelle des signaux a été organisée à l'ITC. 20 étudiants volontaires motivés à contribuer au développement des ressources pour la langue khmère ont été recrutés et formés. En utilisant le logiciel open source Transcriber [Barras *et al.*, 2001], 6h25mn de signaux ont été transcrits. Ce corpus de parole contient 3200 phrases prononcées par 8 locuteurs (3

femmes). 160 phrases ont été extraites afin de constituer les données de test dans nos expérimentations. Le tableau 2.2 montre la répartition de notre corpus de test et d'apprentissage.

Corpus	Durée du signal	Nombre de pharses (nombre de mots)	Nombre de mots uniques
test	25mn	160 (1951)	736
Apprentissage	6h	3040 (39745)	4253
Total	6h25mn	3200 (41696)	4278

TABLE 2.2 – *Répartition du corpus d'apprentissage et corpus de test*

2.3.2 Vocabulaire

Un vocabulaire peut être défini comme une liste close d'unités lexicales qui peuvent être reconnues par un système de reconnaissance automatique de la parole. La taille du vocabulaire et la sélection des unités lexicales dans le vocabulaire influencent fortement les performances du système de transcription automatique (la perplexité des modèles de langages, l'espace de recherche, le taux de reconnaissance, ...) puisque toutes les unités hors-vocabulaire ne peuvent pas être reconnues par le système de reconnaissance.

Le vocabulaire utilisé dans notre système provient de deux sources. La première source est le dictionnaire Khmer *Chuon Nath* (1966, Institut Bouddhiste, Phnom Penh) est le dictionnaire de référence pour le khmer. La version numérique de ce dictionnaire est éditée par l'Institut Bouddhiste du Cambodge, 16,000 mots sont extraits à partir de ce dictionnaire. Comme il est un peu daté, les mots nouveaux de la langue khmère moderne ne sont pas dans ce dictionnaire. La segmentation manuelle d'une partie du corpus de texte (1000 phrases) permet d'extraire 4,000 mots nouveaux comprenant aussi les noms propres. Le vocabulaire utilisé dans notre travail contient alors 20,000 mots. Ce vocabulaire est utilisé pour la segmentation automatique de notre corpus de texte en mots et pour la génération de dictionnaire de prononciation pour la modélisation acoustique.

2.3.3 Corpus de texte

Une grande quantité de données textuelles est nécessaire pour obtenir une estimation précise des probabilités des n-grammes d'un modèle de langage. Comme décrit dans la section 1.3.1, la collecte de textes à partir du web est devenue une approche standard, car ceci permet d'obtenir gratuitement et rapidement une grande quantité de textes. Un robot explore le web et extrait les textes pour construire un corpus. Ces méthodes s'appliquent bien aux langues comme le français ou l'anglais qui disposent d'une grande couverture sur l'Internet. Cependant, les problèmes pour les langues peu dotées comme le khmer, concernent le nombre limité de sites web, la faible vitesse de transmission et la qualité variable des documents qui nécessitera alors plus d'outils de traitement. On préférera par exemple des sites de nouvelles en khmer, au fort contenu rédactionnel au lieu de parcourir tous les sites web contenant très peu de données exploitables.

Une fois les pages html récupérées, des traitements sont nécessaires afin de construire un corpus de texte :

- extraction du contenu textuel à partir du document html,
- conversion des encodages (de l'encodage non standardisé vers le standard Unicode khmer),
- segmentation du texte en différentes unités lexicales et/ou sous-lexicales pour la modélisation statistique du langage,
- transcription des caractères spéciaux,
- conversion des chiffres en mots,
- et normalisation des orthographes car il existe des mots qui possèdent plusieurs orthographes.

Nous avons adapté la boîte à outils ClipsTextTk [Le *et al.*, 2003], développée pour le français et pour le vietnamien, en y ajoutant les traitements spécifiques à la langue khmère. Nous avons introduit des outils pour la conversion de l'encodage, la transcription des caractères spéciaux et des nombres. La normalisation des orthographes consiste à adopter une seule orthographe cohérente pour le mot qui s'écrit de plusieurs façons. Dans beaucoup de cas, la normalisation est faite à base de règles assez simples bien définies dans la langue. Pour les mots qui ne

peuvent pas être normalisés avec les règles, on parcourt notre dictionnaire pour identifier les différentes formes d'écriture possibles pour ensuite les uniifier dans notre corpus. Pour la segmentation automatique, nous avons développé les outils de segmentation de texte khmer en mots, en syllabes et en groupe de caractères. Les détails sur ces méthodes de segmentation seront données dans la section 2.3.4.

A partir de 3 sites Web de journaux en ligne[4], environ 25000 pages html en langue khmère ont été collectées (soit 500 Mo). Après le traitement, notre corpus de texte est constitué de 0,5 millions de phrases, et 15,5 millions de mots (segmentation automatique).

2.3.4 La segmentation automatique

En ce qui concerne la modélisation statistique du langage, nous trouvons dans la littérature que le mot est depuis longtemps l'unité de traitement de base la plus utilisée. Cependant, des travaux récents [Denoual et Lepage, 2006] et [Kurimo *et al.*, 2006] montrent qu'il existe d'autres unités (le caractère et le morphème) qui pourraient être de bons candidats à utiliser comme unité de base (ou en complément du mot), en particulier pour les langues sans séparateurs ou les langues morphologiquement riches. Pour le khmer, les unités sous-lexicales comme la syllabe, le groupe de caractères (appelé aussi cluster de caractères ou CC) présentent un potentiel pour la modélisation du langage. Dans cette perspective de multi unité sous lexicale pour la modélisation statistique du langage, nous avons besoin d'outils de segmentation qui permettent de segmenter un texte khmer en différentes unités : le mot, la syllabe, le groupe de caractères et le caractère. Le figure 2.2 est un exemple de la segmentation d'une phrase khmère en différentes unités.

Segmentation en mot

La segmentation de textes en mot est l'une des tâches fondamentales dans le traitement automatique des langues naturelles (TALN). Beaucoup d'applications de TALN nécessitent en entrée des textes segmentés en mots avant d'effectuer les

4. www.everyday.com.kh, www.rfa.org/khmer, www.cchrcambodia.org

Phrase	ព្រះពុទ្ធជាព្រះបរមគ្រូនៃយើង									
Mot Segmentation 1	ព្រះពុទ្ធ		ជា	ព្រះ	បរមគ្រូ			នៃ	យើង	
Mot Segmentation 2	ព្រះពុទ្ធ		ជា	ព្រះ	បរម		គ្រូ	នៃ	យើង	
Syllabe	ព្រះ	ពុទ្ធ	ជា	ព្រះ	ប	រម	គ្រូ	នៃ	យើង	
CC	ព្រះ	ពុ	ទ្ធ	ជា	ព្រះ	ប	រ	ម	គ្រូ	នៃ	យើង
Traduction	Le boudha est notre maître suprême									

FIGURE 2.2 – *Exemple de segmentation d'une phrase khmer en différentes unités*

autres traitements car le mot est considéré comme l'unité linguistique et sémantique de base. Pour des langues comme le français et l'anglais, il est assez naturel de définir un mot comme une séquence de caractères séparés par des espaces. Cependant, pour les langues non segmentées, la segmentation en mots n'est pas un problème simple. A cause des ambiguïtés dans la langue naturelle, une séquence de caractères peut être segmentée de plusieurs façons. De plus, il peut exister des désaccords entre différentes personnes sur la segmentation d'une phrase donnée. Ce désaccord existe car il y a souvent différentes conventions de segmentation et la définition du mot dans une langue est souvent ambiguë. Cette ambiguïté ne pose pas vraiment de problème pour l'être humain parce qu'une segmentation incorrecte donne généralement une phrase incompréhensible.

La technique générale de segmentation en mots emploie un algorithme qui recherche dans un dictionnaire les mots correspondant à ceux du texte et qui, en cas d'ambiguïté, sélectionne celui qui optimise un paramètre dépendant de la stratégie choisie. Dans les stratégies les plus courantes, l'optimisation consiste à :

- maximiser la taille des mots, pris un par un de gauche à droite, avec retour arrière en cas d'échec (« plus longue chaîne d'abord » ou « longest matching »),
- minimiser le nombre de mots dans la phrase entière (« plus petit nombre de mots » ou « maximal matching »).

Ces techniques recourent intensivement à des dictionnaires, qu'il faut donc créer. Bien que cela puisse être fait automatiquement par apprentissage à partir

d'un corpus, ces dictionnaires ont souvent été créés manuellement.

Les travaux de recherche sur la segmentation automatique en mots de la langue chinoise et thaïe sont très actifs. Parmi les travaux qui utilisent ces techniques, nous pouvons citer [Zhang *et al.*, 2008a] pour le chinois et [Haruechaiyasak et Kongyoung, 2008] pour le thaï. La performance de ces méthodes est acceptable en général (autour de 95% de mots corrects) mais elle dépend fortement de la taille et de la qualité des dictionnaires utilisés pour la segmentation. La performance diminue en présence de cas d'ambiguïté et de mots inconnus.

Il existe des méthodes plus élaborées qui utilisent des méthodes statistiques et/ou passent par une phase d'apprentissage. Dans [Wu, 2003], pour une phrase chinoise à segmenter, un treillis de tous les mots possibles est construit en fonction d'un vocabulaire. Ensuite, des méthodes statistiques sont appliquées pour décoder le chemin le plus probable sur le treillis. Une méthode statistique et linguistique de segmentation en mots est aussi proposée et implémentée sur la langue thaïe [Meknavin *et al.*, 1997]. Dans cette méthode, le contexte des mots est analysé linguistiquement pour déterminer la segmentation la plus probable.

Les méthodes de l'état de l'art utilisent la combinaison de dictionnaire avec les statistiques pour obtenir un meilleur résultat. Cependant, les méthodes statistiques nécessitent de disposer d'un grand corpus de texte segmenté au préalable manuellement. Les méthodes statistiques et les méthodes d'apprentissage complexes ne sont pas appropriées dans notre contexte des langues peu dotées car les ressources nécessaires pour implémenter ces méthodes n'existent pas. Pour une langue considérée, nous cherchons des méthodes de segmentation performantes, rapides, faciles à implémenter et qui tirent, au mieux, bénéfice des ressources limitées existantes pour la langue.

La deuxième édition d'une évaluation sur la segmentation en mots en chinois, le « Second International Chinese Word Segmentation Bakeoff », a eu lieu en 2005. Dans le compte rendu des résultats [Emerson, 2005], on indique que la gestion des mots OOV est le principal problème malgré les améliorations par rapport aux résultats de la première évaluation en 2003. Néanmoins en reconnaissance de la parole, l'unité de mesure des performances pour le chinois mandarin est le caractère et non le mot.

2.3.5 La segmentation automatique pour le khmer

Segmentation en mots

Dans le but d'implémenter un outil simple de segmentation en mots, utilisable facilement pour plusieurs de langues, nous avons choisi la méthode de segmentation à base d'un vocabulaire de mots. Cette méthode utilise un algorithme de programmation dynamique qui recherche dans un vocabulaire les mots du texte et qui le segmente en utilisant une critère d'optimisation. Trois critères d'optimisation sont implémentés :

- « plus longue chaîne d'abord » (Longest Matching)
- « plus petit nombre de mots » (Maximal Matching).
- la fréquence uni-gramme des mots (modèle de langage uni-gramme)

Dans la dernière stratégie, notre fonction de coût est la probabilité d'apparition de la chaîne de mots que nous venons de segmenter. C'est-à-dire, la segmentation repose sur un modèle statistique de langage uni-gramme. Pour créer un modèle de langage uni-gramme de bonne qualité, nous avons besoin d'un grand corpus de texte segmenté. Puisqu'un tel corpus n'existe pas dans notre travail, nous décidons d'utiliser un petit corpus de texte (500 phrases) segmenté manuellement pour créer un tel modèle de langage uni-gramme.

Pour évaluer la performance de nos outils de segmentation, nous avons expérimenté sur un corpus de test qui contient 1000 phrases. Après la segmentation manuelle, nous obtenons 31042 mots et on peut générer à partir de ce corpus un vocabulaire de 4875 mots. Pour voir l'impact des mots hors-vocabulaire (OOV) sur la performance des méthodes de segmentation automatique à base de dictionnaire, nous avons créé des scénarios de test avec différents taux OOV. A partir de notre vocabulaire de 4875 mots initial qui représente 0% de OOV, nous avons enlevé successivement les mots les moins fréquents pour créer plusieurs vocabulaires avec taux de OOV croissants (de 5% à 50%) par rapport au corpus de test. Les performances de segmentation sont présentées dans le tableau 2.3.

Nous observons que dans le cas d'absence de mots hors vocabulaire, la performance est autour de 93% pour les trois méthodes mais la performance chute

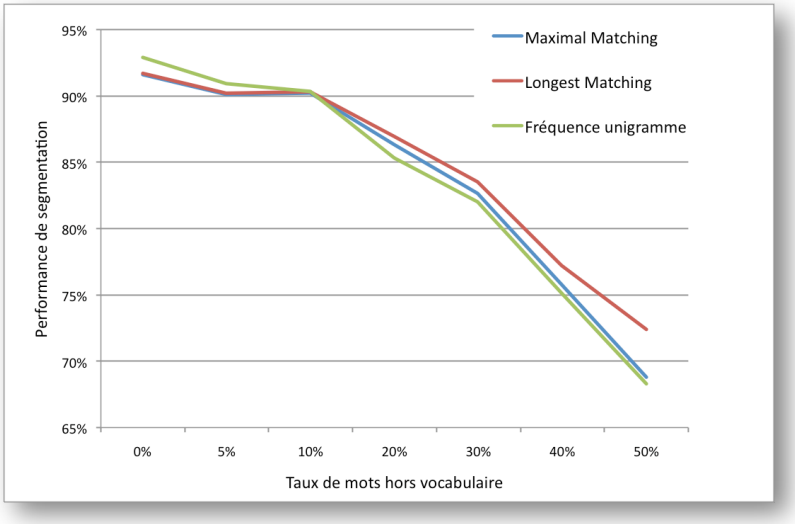


FIGURE 2.3 – *Taux des mots corrects pour les 3 méthodes de segmentation à base de vocabulaire en fonction du taux de mots hors-vocabulaire*

dramatiquement quand le taux des mots hors vocabulaire augmente dans le corpus à segmenter. Pour les langues peu dotées, il est difficile d'obtenir un dictionnaire avec un taux des mots hors-vocabulaire faible. Dans ce cas, on risque donc d'atteindre une mauvaise performance de segmentation automatique sur le corpus d'apprentissage et la performance du modèle de langage appris à partir de ce corpus mal segmenté sera alors médiocre.

Segmentation en syllabes

La syllabe est considérée comme l'unité structurante de la langue et elle est reconnue comme unité fondamentale dans plusieurs applications dans le domaine du traitement automatique de la parole. Généralement, la structure d'une syllabe se décompose souvent en 3 parties : l'attaque (une ou plusieurs consonnes -facultatif), le noyau (une voyelle ou une diphtongue - obligatoire) et la coda (une ou plusieurs consonnes facultatif).

La syllabe khmère respecte cette structure et est généralement sous forme : $C_1(C_2)(C_3)V_1(C_4)$ (les C_i et les V_i étant des consonnes et des voyelles en khmer,

(-) est facultatif) [Huffman, 1970].

L'outil Sylla proposé dans [Berment, 2004] permet de créer le modèle syllabique pour le khmer à base de règle. Ce modèle syllabique a été implémenté dans [Le, 2006] pour segmenter le texte khmer en syllalbe et donne une performance de 81,5%.

A cause de la caractéristique facultative des consonnes sur l'attaque et sur la coda, il y a des ambiguïtés de segmentation d'une phrase en syllabes ce qui diminue la performance. En plus, il y a en langue khmère des mots avec des exceptions, en particulier des mots de racine Pali et Sancrit qui ne respectent pas cette règle. Pour obtenir une meilleure performance, nous réutilisons la méthode à base d'un vocabulaire de syllabe pour segmenter le texte khmer en syllabe. Dans un premier temps, le modèle syllabique est utilisé pour segmenter un corpus de texte en syllabes pour extraire une liste préliminaire de syllabes. Cette liste est ensuite corrigée et enrichie manuellement pour obtenir un vocabulaire de syllabes de base. Ce vocabulaire de syllabes est ensuite utilisé en entrée du programme de segmentation à base de vocabulaire syllabe pour segmenter le corpus de texte en syllabes. Cette méthode donne une performance de 90% de syllabes segmentées correctement.

Segmentation en cluster de caractères

La notion de groupe de caractères (characters cluster) est abordée dans [Theeramunkong *et al.*, 2000] pour la langue thaïe. Dans la langue khmère, un groupe de caractères est une unité non ambiguë et non séparable avec une structure bien définie.

Dans l'encodage Unicode, un groupe de caractère khmer peut être décrit en BNF comme :

$$B(S) * (C)(V)(O) \tag{2.2}$$

où

– B est une consonne ou une voyelle indépendante,

– S est une consonne souscrite ,
– C est un signe spécial utilisé pour changer le son de la consonne khmère,
– V est une voyelle dépendante,
– O des autres signes du khmer (diacritique),
– (-) est facultatif.
– * nombre d'occurrence est égale à 0 ou plus

Phrase	ព្រះពុទ្ធ						
Segmentation en caractères	ព	្រ	ះ	ព	ុ	ទ	្ធ
BNF	B	S	V	B	V	B	S
Cluster de caractères	ព្រះ			ពុ		ទ្ធ	
Traduction	Le boudha						

FIGURE 2.4 – *Exemple de règle de segmentation en cluster de caractères*

Le travail de segmentation de texte en groupe de caractères est assez trivial. Notre programme qui implémente cette règle en utilisant des expressions régulières pour détecter la frontière entre les groupes de caractères khmers donne une performance de segmentation en cluster de caractères de 100%. Le figure 2.4 donne un exemple de segmentation d'un mot khmer en clusters de caractères.

2.4 Modélisation de prononciation

Un dictionnaire de prononciation (ou dictionnaire phonétique) est une ressource essentielle pour les systèmes de synthèse et de reconnaissance automatique de la parole. Au début de ce travail, il n'existait à notre connaissance aucun dictionnaire phonétique sous forme électronique pour le khmer. Alors que les corpus de texte et de parole peuvent être collectés, le dictionnaire de prononciation n'est généralement pas directement disponible. Bien qu'un dictionnaire de prononciation créé manuellement donne une bonne performance, la tâche est très lourde à réaliser et demande des connaissances approfondies sur la langue en question.

Dans le cadre de notre travail, nous avons opté pour les méthodes automatiques pour générer la prononciation des mots khmers en se fondant sur les règles. La première approche est d'utiliser des connaissances linguistiques pour définir les règles de conversion graphème-phonème. Une deuxième approche mieux adaptée aux langues peu dotées consiste à utiliser directement les graphèmes comme les unités de modélisation sans la conversion graphème-phonème.

Dictionnaire phonétique à base de règle de conversion graphème-phonème

Notre tâche consiste à générer automatiquement les prononciations pour les 20,000 mots du vocabulaire khmer décrits précédemment en utilisant les règles de conversion graphème-phonème.

Les mots khmers sont généralement monosyllabiques et bi-syllabliques. Comme mentionné dans la section 2.3.5, la structure syllabique générale du khmer se retrouve sous la forme : $C(C)(C)V(CF)$ où C est une consonne initiale, $C(C)$ est une double consonne, V est une voyelle, et CF est une consonne finale. Le tableau 2.5 décrit la correspondance phonétique entre les consonnes initiales, les doubles consonnes et les voyelles avec les symboles API (Alphabet Phonétique International).

Type of phone		Phone symbols
Initial Consonants	Single *CI*	k kʰ ŋ c cʰ ɲ d tʰ n t tʰ b p pʰ m j r s ʊ h l ʔ
	Consonant Cluster *CC*	85 double consonants cluster possible. Please refer to [13] for a complete list
Vowels *V*	Short	i e ɨ ə a a u o
	Long	iː eː ɛː ɨː əː aː aː uː oː ɔː
	diphthong	iə ei ɨə əi ao uə ou ɔə eə̆ uə̆ oə̆
Final Consonants *CF*		k c t p h n ŋ ɲ m j l ʊ ʔ

FIGURE 2.5 – *Les phonèmes khmers*

A l'aide des règles de prononciation décrites dans le manuscrit pour apprendre à lire le khmer [Huffman, 1970], Nous avons extrait 20 règles pour reconnaître et phonétiser les syllabes khmères. Le tableau 2.6 présente ces 20 règles.

La génération de prononciation d'une syllabe se fait en deux phases :

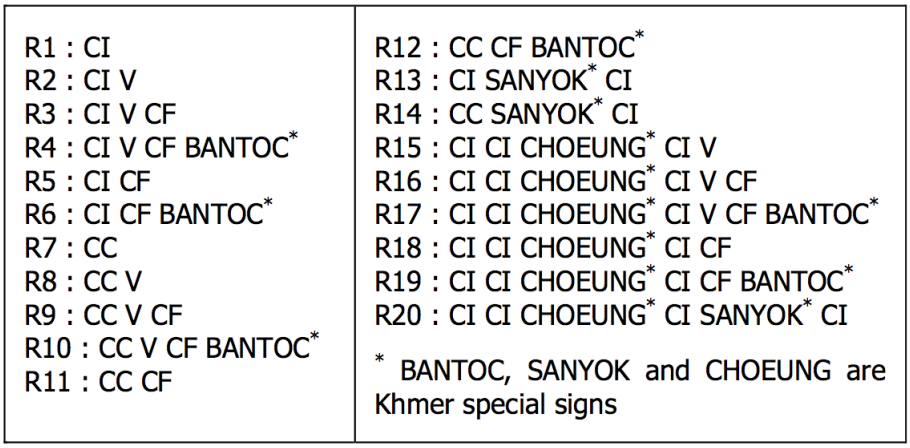


FIGURE 2.6 – *Règles pour les syllabes khmères*

1. La reconnaissance de la structure syllabique de la séquence en entrée
2. La phonétisation qui consiste à faire la conversion graphème-phonème utilisant les règles de prononciation en fonction de la structure syllabique détectée.

Pour générer la prononciations d'un mot monosyllabique S, nous appliquons toutes les règles dans l'ordre de R_1 à R_{20} sur S. Quand la structure syllabique du mot S est reconnue par une règle R_i où $1 \leq i \leq 20$, le programme va générer la séquence de phonèmes en utilisant la règle de conversion graphème-phonème pré-définie par la règle R_i. Un mot polysyllabique $W = S_1S_2...S_n$ est reconnu par une règle formée par la concaténation des n règles de base. Pour phonétiser les mots bi-syllabiques, nous devons générer $20 \times 20 = 400$ règles.

La version actuelle de notre outil ne permet pas de générer les variantes de prononciation. Ces 20 règles de base permettent de phonétiser les mots monosyllabiques et polysyllabique simple, les mots d'origine Pali et Sanskrit et les mots qui contiennent les exceptions ne peuvent être phonétisés par cet outil. Mais pour ces mots qui ont les exceptions dans la prononciation, le dictionnaire Khmer *Chuon Nath* (1966, Institut de Bouddhiste, Phnom Penh) propose la phonétique en khmer en utilisant l'écriture khmer simple qui peut être phonétisée par nos règles. Parmis les 20000 mots de notre vocabulaire, nous arrivons à phonétiser 18500 mots. Il nous reste 1500 mots que nous avons phonétisé manuellement.

Dictionnaire phonétique à base de graphème

Khmer word	Grapheme based pronunciation
ចចក	Ca Ca Ka
ចតុមុខ	Ca Ta U Mo U Kha
ក្រាមដី	Ka COENG Ro OO Mo Da II

FIGURE 2.7 – *Dictionnaire de prononciation en khmer à base de graphèmes*

Cette méthode est fondée sur le graphème et consiste à représenter chaque graphème khmer comme une unité de modélisation. Le processus de génération pour un mot est tout simplement une conversion des caractères Unicode khmers vers leur nom Romain de Unicode. Il y a au total 77 graphèmes dans l'alphabet khmer moderne : 33 consonnes, 16 voyelles dépendantes, 16 voyelles indépendantes et 12 diacritique et signes. Le tableau 2.7 donne un exemple de quelques mots khmers dans leur représentation graphèmique utilisant sur le nom Romain de chaque symbole Unicode khmer.

2.5 Modélisation acoustique

Notre dictionnaire de prononciation à base de graphèmes contient 77 graphèmes utilisés comme unités de modélisation. Dans le cas de la modélisation à base de phonèmes, nous utilisons les phonèmes simples comme unités de base. Un cluster de consonnes est considéré comme une séquence de 2 consonnes simples : /pt/ → /p/ + /t/. Comme les voyelles longues et courtes possèdent les mêmes propriétés acoustiques mais avec des durées de voisement différentes (les voyelles longues sont en général deux fois plus longues que les voyelles courtes), une voyelle longue est représentée comme la concaténation de deux voyelles courtes : /e :/ → /e/ + /e/. De la même manière, les diphtongues sont considérées comme une séquence de voyelles simples. Nous avons finalement 33 phonèmes.

Nous utilisons SphinxTrain pour entraîner les modèles acoustiques (HMMs). Des modèles indépendants du contexte (CI) et dépendants du contexte (CD avec

1000 états) à base de graphèmes et de phonèmes sont construits à partir des corpus de parole décrits en section 2.2. Nous obtenons ainsi 4 modèles acoustiques, à savoir *Graphème_ CI, Graphème_ CD, Phonème_ CI* et *Phonème_ CD*.

2.6 Modélisation du langage

Pour une langue ayant un système d'écriture sans espace entre les mots comme le khmer, on doit utiliser un système imparfait de segmentation automatique en mots, qui introduira des erreurs pendant la segmentation. Une alternative possible consiste à calculer les probabilités à partir d'unités sous-lexicales. Ces dernières permettent une estimation des probabilités plus précise, le vocabulaire étant généralement plus petit et améliore le taux des mots hors vocabulaire. En contrepartie, la couverture des n-grammes est plus réduite.

Comme présenté dans la section 2.3.4, dans le cas du khmer, le texte peut être segmenté en mots, syllabes ou clusters de caractères. A priori, les clusters de caractères semblent être une bonne unité de modélisation, étant donnée que la segmentation est triviale et sans ambiguïté.

Dans notre système khmer, nous essayons de comparer la performance des différentes unités lexicales (mots) et sous-lexicales (syllabes et cluster de caractères) dans la modélisation statistique de langage. Le corpus d'apprentissage est d'abord segmenté en mots en utilisant la méthode à base de dictionnaire (vocabulaire de 20k mots). Un vocabulaire de 8800 syllabes est obtenu en segmentant le vocabulaire de mots. Pour obtenir le vocabulaire de clusters de caractères (CC), nous segmentons notre corpus d'apprentissage en cluster de caractères et tous les clusters de caractères sont extraits pour former le vocabulaire. Un vocabulaire de 3500 clusters de caractères est obtenu. Pour apprendre nos différents modèles de langage à base de différentes unités lexicale et sous-lexicale, nous devons re-segmenter le corpus vers chaque unité. Le tableau 2.3 donne les détails sur ces différents corpus.

Unité	Taille de vocabulaire	Taille de corpus re-segmenté
Mot	20000	15,5M
Syllabe	8800	35,4M
CC	3500	49.5M

TABLE 2.3 – *Taille de vocabulaire et de corpus d'apprentissage resegmenté.*

2.7 Résultats d'expérimentation

Les expérimentations sont menées avec Sphinx3 [13]. La topologie des modèles est un HMM de 3 états avec 16 Gaussiennes par état. Le vecteur de paramètres contient 13 MFCCs, ses premieres et secondes dérivées. En plus du taux d'erreur de mots (WER), nous utilisons systématiquement le taux d'erreur de syllabes (SER) et le taux d'erreur de cluster de caractères (CCER) pour l'évaluation du système de RAP, car la segmentation de mots et syllabes khmers n'est pas triviale et les erreurs de segmentation pourraient empêcher une comparaison correcte entre les systèmes. Les tests sont effectués sur le corpus de test qui contient 160 phrases (environ 25mn de parole).

2.7.1 Modèle acoustique à base de Phonème Vs Graphème

Dans cette expérimentation, nous voulons comparer la performance de nos différents modèles acoustiques : le modèle à base de phonème et celui à base de graphème. La table 2.4 montre les résultats avec différents modèles acoustiques.

Modèle acoustique	WER	SLER	CCER
Graphème_CI	64,9	39,9	33,6
Graphème_CD	47,8	26,9	21,7
Phonème_CI	57,9	38,2	31,9
Phonème_CD	49,6	25,1	19,1

TABLE 2.4 – *Modèle acoustique à base de Phonème Vs Graphème*

Les résultats de la table 2.4 montrent que les modèles dépendants du contexte sont meilleurs que les modèles indépendants du contexte, même si la quantité de données d'apprentissage est faible (moins de 7h). Les performances des modèles

à base de graphèmes et à base de phonèmes sont très comparables, ce qui montre le potentiel de l'approche à base de graphèmes dans le contexte de cette langue peu dotée. La différence observée entre WER et CCER est partiellement due aux erreurs de segmentation dans les hypothèses et les références. Ceci confirme que le CCER semble plus adapté pour les évaluations car il permet des comparaisons plus justes (sans erreurs de segmentation).

2.7.2 Modèles mot/sous-mot

Pour observer le potentiel des différentes unités lexicales et sous-lexicales, trois modèles de langage trigrammes sont appris en utilisant respectivement le mot, la syllabe et le cluster de caractères, comme unité de base de la modélisation.

Modèle de langage	Modèle acoustique	CCER
LMmot	Graphème_CD	21,7
LMsyl	Graphème_CD	25,0
LMcc	Grapheme_CD	34
LMmot	Phonème_CD	19,1
LMsyl	Phonème_CD	26,3

TABLE 2.5 – *Performance des modèles mots/sous-mots*

Les résultats de la table 2.5 montrent que le mot reste la meilleure unité malgré les erreurs de segmentation. Une explication possible est qu'un mot khmer se compose en moyenne de 3,2 syllabes et de 4,3 clusters de caractères. Par conséquent, la couverture du modèle trigramme de syllabes et de clusters de caractères est beaucoup plus réduite que celle du modèle trigramme de mots.

2.8 Conclusion

Ce chapitre présente le développement d'un système de RAP pour le khmer. Une méthode de collecte rapide de données linguistiques a été utilisée pour la construction des corpus. Pour la modélisation acoustique, les résultats montrent que la modélisation à base de graphèmes présente un bon potentiel dans le cas d'une langue peu dotée comme le khmer. Pour traiter le problème du manque

de données et des erreurs de segmentation dans la modélisation du langage, nous avons essayé d'exploiter différentes vues sur les données en utilisant les unités lexicales et sous-lexicales. Les résultats des tests ont montré que le mot reste la meilleure unité de modélisation. Dans le chapitre suivant, nous allons analyser comment ces différentes unités lexicales et sous-lexicales peuvent être exploitées simulatnément dans la reconnaissance automatique de la parole des langues peu dotées et non-segmentées comme le khmer.

Chapitre 3

Utilisation de multiples unités lexicales dans le système de RAP

3.1 Introduction

Dans ce chapitre, nous souhaitons analyser comment les différentes unités lexicales et sous-lexicales peuvent être exploitées dans la reconnaissance automatique de la parole des langues peu dotées et non-segmentées. Nous traitons ce problème en exploitant simultanément plusieurs vues sur les données textuelles dans la modélisation. Nous proposons d'aborder ce problème à plusieurs niveaux. D'une part, au niveau du modèle de langage en créant des modèles à partir de vocabulaires hybrides qui utilisent à la fois des unités lexicales et sous-lexicales. Ensuite, au niveau du système, nous proposons de combiner des sorties de différents systèmes fondés sur ces différentes unités pour décoder une meilleure hypothèse. Nous appliquons enfin ces deux méthodes à la reconnaissance automatique de la parole du vietnamien et du khmer.

3.2 Les unités utilisées dans la modélisation statistique du language

3.2.1 Le mot : unité de base

Le mot est généralement l'unité de référence en reconnaissance automatique de la parole. Si l'on rappelle l'équation de base de la reconnaissance de la parole $M' = argmaxP(M|O) = argmaxP(O|M)P(M)$, développée dans la section 1.2.2, le but est de trouver la meilleure séquence de mots M à partir du signal O. Les systèmes de reconnaissance sont donc en général évalués avec une mesure d'erreur moyenne sur les mots, appelée WER pour *Word Error Rate*. Rappelons la définition donnée à la section 1.2.7 : WER est défini par la somme de trois types d'erreurs qui sont l'insertion I, la substitution S et l'élision D de mots, divisée par le nombre de mots N de la référence : $(I + S + D)/N$. Ce taux est calculé après alignement dynamique de l'hypothèse du décodeur avec une référence.

En linguistique, la notion de mot est souvent s'écrite comme problématique, des difficultés apparaissant lorsque l'on veut délimiter les mots. Toutes les langues ne séparent pas les mots à l'écrit. C'est le cas de nombreuses langues asiatiques comme le chinois, le japonais, le thaï et le vietnamien, le laotien et le khmer. Pour ces langues, des algorithmes de segmentation en mots sont utilisés en prétraitement ou post-traitement. Il s'agit généralement d'algorithmes fondés sur un lexique déjà constitué mais ces techniques automatiques font encore actuellement l'objet de recherches. Comme nous avons montré dans la section 2.3, la qualité du lexique (taux de mots hors-vocabulaire) utilisé pour la segmentation automatique a une influence directe sur la performance de segmentation. Pour les langues dites peu dotées, il est difficile d'obtenir un lexique relativement complet avec un taux de mots hors-vocabulaire faible. On doit utiliser dans ce cas, un système imparfait de segmentation automatique en mots, qui introduira des erreurs dans l'estimation du vocabulaire et du modèle de langage. Cela mène à un taux de mots hors-vocabulaire plus important pour la modélisation statistique du langage. La quantité de données disponible étant limitée, l'estimation des probabilités du modèle de langage n'est pas bonne et cela nuit d'autant plus à la performance du

modèle de langage.

3.2.2 Sous-unités

Une sous-unité désigne une unité lexicale plus petite que le mot. La recherche d'unités lexicales peut conduire à des décompositions de mots en sous-unités qui font sens, il peut s'agir par exemple de morphèmes, de syllabes ou de caractères.

En reconnaissance de la parole, l'utilisation d'unités lexicales plus petites que les mots n'est pas nouvelle mais fait l'objet de recherches actuelles en particulier pour des langues qui forment les mots par composition de morphèmes grammaticaux et/ou lexicaux. Pour les langues comme le finnois, l'estonien, le turc, et l'arabe, en raison de leur morphologie particulièrement riche, les tailles de lexique augmentent très vite avec la taille des textes. De très grands lexiques sont donc nécessaires pour avoir une couverture lexicale correcte. Pour ces langues, l'utilisation d'unités plus petites que le mot parait donc très intéressante, en terme de taux de mot hors-vocabulaire et de problèmes de manque de données textuelles.

Parmi les travaux existants qui utilisent les unités sous-lexicales pour la modélisation du langage, nous pouvons citer [Afify *et al.*, 2006] et [Abdillahi *et al.*, 2006] qui utilisent les morphèmes pour l'arabe et le somali. Pour une langue non-segmentée comme le chinois, le caractère est utilisé dans [Luo *et al.*, 2009]. Dans [Nimaan, 2007], des sous-unités appelées racines sont utilisées pour la reconnaissance automatique et la recherche d'information pour une langue peu dotée, la langue somali parlée notamment à Djibouti. La sélection automatique de sous-unités lexicales est utilisée pour la modélisation de la langue amharique dans [Pelligrini, 2008].

Deux approches sont souvent utilisées pour déterminer les sous-unités à utiliser dans la modélisation :

- approche supervisée, qui fait appel aux règles grammaticales dépendantes de la langue et à des connaissances linguistiques,
- approche non-supervisée, avec peu ou pas de connaissance linguistique et indépendante de la langue.

Approche supervisée

Un analyseur morphologique est nécessaire pour obtenir les morphèmes. Certaines langues bénéficient de cet outil qui donne toutes les analyses morphologiques possibles d'un mot donné en entrée, c'est-à-dire que le mot est décomposé en sous-unités qui sont données avec leurs traits grammaticaux. Un exemple d'analyseur morphologique très utilisé est TreeTagger [Schmid, 1994]. Il a été adapté à plus d'une dizaine de langues, l'allemand, l'anglais, le français, l'italien, le néerlandais, l'espagnol, le portugais, le bulgare, le russe, le grec et le mandarin.

Pour construire un analyseur morphologique, les données nécessaires sont un lexique de mots « racines » ou lemmes, un lexique de morphèmes, les deux lexiques étant donnés avec les informations syntaxiques associées. Des listes de combinaisons possibles et impossibles de morphèmes (propriétés morphotactiques) ainsi que les règles associées (propriétés morphographémiques) sont également nécessaires. Cependant, l'analyseur morphologique souffre aussi de problème de OOV car il utilise un lexique et des règles grammaticales.

Même si le morphème est logiquement la sous-unité lexicale de choix pour la modélisation des langues morphologiquement riches, il a besoin des outils dépendant de la langue comme l'analyseur morphologique. Le manque d'un bon analyseur morphologique peut amener une mauvaise décomposition en morphèmes et la taille limitée du dictionnaire racine peut conduire à une faible couverture. Dans ce cas, les méthodes dites non-supervisées peuvent être utilisées pour pallier ce problème.

Approche non-supervisée

Les approches non supervisées visent à faire intervenir le moins possible de connaissances linguistiques. Elles peuvent éventuellement faire appel à des règles ou à des heuristiques pour initialiser les modèles.

La sous-unité similaire au morphème dite "morphe" peut être obtenue par l'heuristique fondée sur une approche de type Minimum Description Length (MDL), qui apprend un lexique d'unités sous-lexicales de manière non-supervisée à partir

de donnée d'apprentissage en mots.

Morfessor [Creutz et Lagus, 2006] est un algorithme d'apprentissage non-supervisé qui ne nécessite aucune connaissance a priori de la langue étudiée et les morphes qu'il propose pour un lexique donné dépendent uniquement des mots de ce lexique. Morfessor propose deux modes d'utilisation :

- un mode "entraînement" : un modèle de découpage des mots d'un lexique donné (avec éventuellement les comptes d'occurrences des mots) est créé. L'entraînement est du type maximisation a posteriori (MAP) et utilise des propriétés exprimées sous forme de probabilités ou pseudo-probabilités comme par exemple la probabilité des séquences de caractères qui composent les mots du lexique,
- un mode "décodage" : un modèle de décomposition des mots créé au préalable peut être utilisé pour découper un nouveau lexique de mots. Le choix des découpages de mots est réalisé à l'aide d'un algorithme de type Viterbi qui maximise les découpages donnant des unités les plus fréquentes possibles. En effet, souvent plusieurs découpages sont possibles et dans ce cas, l'algorithme choisit en se basant sur la fréquence des morphes comme critère de sélection.

Morfessor n'a besoin d'aucune hypothèse sur le nombre de sous-unités qui composent les mots, pour cette raison cette méthode ressemble aux méthodes de segmentation de textes. D'autre part, son cadre purement probabiliste correspond très bien au domaine de la reconnaissance de la parole et rend l'outil particulièrement simple à adapter. Dans [Pelligrini, 2008], un paradigme statistique fondé sur une méthode d'apprentissage sur corpus, qui tente de trouver des décompositions de mots adaptées à la tâche de reconnaissance, est proposé. Morfessor est adapté pour effectuer cette tâche de sélection automatique de sous-unités lexicales.

L'approche non-supervisée présente plusieurs avantages par rapport aux règles grammaticales (supervisé). Nous n'avons plus besoin de créer des règles, ni besoin de connaissance linguistique pour procéder. Malgré la fait qu'un très bon analyseur morphologique soit disponible pour le finnois, [Kurimo *et al.*, 2006] a montré que la modélisation de la langue finnoise fondés sur les morphes fournits par Morfessor est aussi performante que le modèle fondé sur une analyse morphologique à base des règles.

3.3 Modèle de langage hybride

Un modèle dit hybride est un modèle qui utilise plusieurs unités dans la modélisation. L'idée sous-jaccente à cette notion de modèle de langage hybride est de tirer parti des points forts de chaque unités lexicale et sous-lexicale dans l'estimation du modèle. Par exemple, l'avantage principal des modèles de langage à base de mot est le fait que la couverture en n-gramme est plus longue que lorsqu'on utilise des unités sous-lexicales. Mais le mot souffre du problème de taux de mots hors vocabulaire et de l'explosion de la taille de vocabulaire, en particulier dans le contexte des langues non segmentées ou des langues morphologiquement riches. Cependant, le problème de mots hors-vocabulaire, et la taille de vocabulaire peut être résolu en utilisant une unité plus petite que le mot.

Dans la littérature, il existe des travaux qui visent à utiliser de multiples unités dans la modélisation statistique du langage. Dans [Arisoy *et al.*, 2006], des modèles de langage dits unifiés sont créés en combinant trois types d'unités : mots, morphèmes et racines pour la reconnaissance automatique de la parole en langue turque. Le vocabulaire qui se compose de ces trois unités, réduit de façon significative le taux de mots-hors vocabulaire. Les travaux sur une langue non-segmentée comme le chinoise cherchent également à combiner les unités lexicales et sous-lexicales dans la modélisation statistique de langage. Dans [Chen *et al.*, 2000], la combinaison entre les caractères et les mots chinois dans le modèle du langage a été expérimentée dans un système de transcription automatique de journaux télévisés en mandarin. L'expérimentation a montré que la meilleure performance est obtenue lorsqu'on utilise un vocabulaire créé en ajoutant dans la liste des caractères chinois (7000 caractères), les 20% des mots les plus fréquents du corpus d'apprentissage. Un article plus récent [Luo *et al.*, 2009] par le même groupe d'auteurs, a montré que les mots en chinois donnent une meilleure performance que les caractères dans la modélisation malgré un taux de mots hors-vocabulaire plus important. La combinaison des sorties des systèmes à base de mot et à base de caractère en utilisant la méthode ROVER a été également testé mais sans donner une gain de performance. Il est suggéré que le choix des unités de modélisation et de segmentation reste encore un problème important pour la reconnaissance automatique de la parole en mandarin.

Dans le cadre de cette thèse, des modèles de langage hybrides sont étudiés pour les langues non segmentées (application à la langue khmère et vietnamienne). L'idée est qu'à partir d'un vocabulaire initial d'unités sous-lexicales, nous ajoutons progressivement les n mots les plus fréquents pour former un nouveau vocabulaire. En faisant varier n, nous obtenons plusieurs vocabulaires hybrides. Le corpus d'apprentissage est ensuite segmenté en différentes unités en fonction du vocabulaire hybride utilisé pendant la segmentation. Les modèles du langage trigrammes sont finalement entraînés et utilisés dans le système de reconnaissance automatique. Notre objectif est d'étudier si les modèles hybrides qui exploitent plusieurs unités dans la modélisation donnent une meilleure performance qu'un modèle simple qui utilise une seule unité.

3.4 Combinaison de systèmes

Dans le but d'accroître la robustesse des systèmes, les méthodes qui consistent à combiner plusieurs systèmes de reconnaissance différents afin de profiter de leur éventuelle complémentarité ont été étudiées dans [Lecouteux, 2008], [Barrault, 2008] et [Li, 2005]. La combinaison se fait généralement après le décodage (fusion tardive), on parlera alors de combinaison post-décodage. Ces méthodes s'applique sur le graphe d'hypothèses entier (treillis) ou sur les N-meilleures hypothèses issues de différents systèmes. Le principe est de considérer et comparer les différentes hypothèses générées par plusieurs systèmes de reconnaissance afin de constituer une nouvelle hypothèse. De nombreuses techniques peuvent être employées pour combiner, comme la comparaison des probabilités a posteriori des mots ou l'utilisation de mesures de confiance.

La combinaison de systèmes de reconnaissance de la parole est bénéfique particulièrement lorsque les systèmes sont complémentaires. Dans notre contexte, nous souhaitons exploiter les multi-unités dans la reconnaissance automatique de la parole en essayant de combiner les unités lexicales et sous-lexicales. Rappelons que l'unité lexicale de type "mot" permet une bonne couveture en tri-gramme mais rencontre le problème des mots OOV, tandis que l'unité sous-lexicale de type caractère, syllabe ou morphème donne une plus courte historique de tri-gramme

mais elle n'a pas pas ou très peu de problème de mots hors vocabulaire. La complémentarité de ces différentes unités pourrait être exploitée avec cette approche de combinaison post-décodage de systèmes.

Nous proposons d'utiliser dans le cadre de notre travail deux méthodes de combinaison de systèmes :

- Combinaison par consensus de type ROVER [Fiscus, 1997] sur les hypothèses des différents systèmes,
- Combinaison des treillis des hypothèses transformés ensuite en réseau de confusion (confusion network) pour le décodage.

3.4.1 Combinaison par consensus : ROVER

ROVER (Recognizer Output Voting Error Reduction) développé par NIST, est destiné à réduire le taux d'erreur de mots à partir des transcriptions de plusieurs systèmes. La méthode consiste en un vote sur l'ensemble des hypothèses alignées. ROVER s'appuie sur deux modules : le premier fusionne les transcriptions pour en faire un réseau de confusion et le second effectue un vote sur chaque nœud du réseau :

$$Score(w) = \alpha \frac{C_i(w)}{N} + (1 - \alpha) S_i(w) \tag{3.1}$$

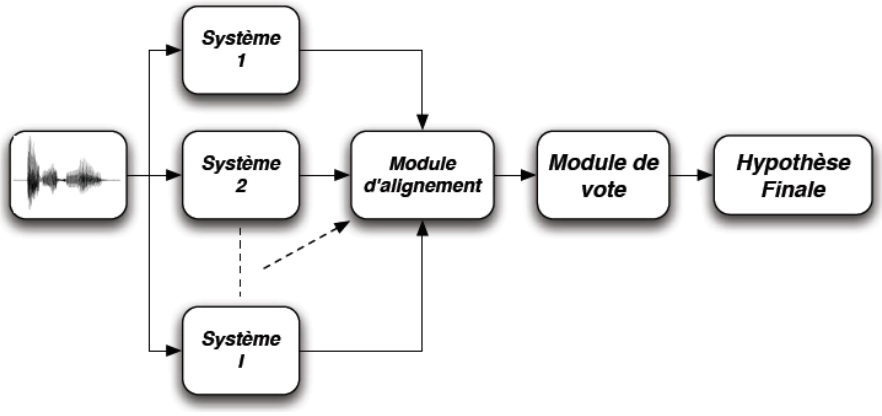


FIGURE 3.1 – *Principe de combinaison via ROVER*

Où N est le nombre de systèmes combinés, $S_i(w)$ est le score de confiance du mot w à la position i de l'hypothèse, et $C_i(w)$ le nombre d'occurrences du mot w à la position i de l'hypothèse, α est calculé empiriquement sur un corpus d'apprentissage.

ROVER (figure 3.1), dans sa première présentation offre trois types de vote selon :

- la fréquence d'apparition : c'est le vote majoritaire pur. Dans ce cas, le paramètre α est égal à 1.0,
- le score de confiance moyen : le nombre d'occurrences et les scores associés à chaque hypothèse de mot par les différents systèmes permettent de sélectionner les meilleures hypothèses. α est estimé sur des données différentes du corpus de test,
- le score de confiance maximum : l'hypothèse ayant le score maximum parmi les hypothèses proposées sera sélectionnée. α est également estimé sur un corpus.

Comme expliqué dans [Hillard *et al.*, 2007], l'alignement est dépendant de l'ordre des permutations effectuées par le système. Le résultat est donc dépendant de l'ordre de combinaison des hypothèses de phrase de chaque système. Il a été montré que les meilleurs résultats sont obtenus lorsque les systèmes sont ordonnés par ordre croissant de taux d'erreur mot. Schwenk et Gauvain dans [Schwenk et Gauvain, 2000] proposent une amélioration de ROVER, constatant que l'information linguistique n'est pas utilisée lors d'une application classique du ROVER. En effet, le vote s'effectue entre n transcriptions, et le mot choisi ne prend pas en considération l'historique des précédents élus. Ceci se vérifie d'autant plus lorsque le choix s'effectue entre plusieurs mots qui ont des scores similaires. Pour résoudre ce problème, ils introduisent des informations linguistiques au sein de l'algorithme de décision. Ainsi, pour chaque vote, l'aspect linguistique est pris en compte. Lorsque ROVER ne peut pas prendre de décision, le modèle linguistique apporte sa contribution, améliorant nettement le résultat de la combinaison.

Dans le cadre de cette thèse, afin d'étudier le potentiel de la combinaison des unités lexicales et sous-lexicales, nous appliquons une méthode qui combine des listes des N-meilleurs hypothèses décodées par différents systèmes et tente

d'en extraire les meilleures hypothèses. D'abord, chaque système décode une liste de N-meilleures hypothèses et toutes les hypothèses sont ramenées à l'unité la plus petite commune. Nous combinons ensuite les sorties de ces systèmes avec l'approche ROVER, nous appliquons l'algorithme de vote majoritaire basé sur le nombre d'occurrences des unités dans la liste des hypothèses pour décoder de nouvelles hypothèses.

3.4.2 Combinaison des treillis

Afin de décoder la meilleure hypothèse pour une signal de parole en entrée, la plupart des systèmes de reconnaissance génèrent d'abord un treillis contenant toutes les hypothèses avec des scores issus des différents modèles. La méthode de combinaison d'hypothèses ROVER présentée précédemment travaille au niveau des hypothèses qui sont généralement obtenues à partir de ce treillis. La combinaison des systèmes directement au niveau de treillis pourrait être plus avantageuse car un treillis contient plus d'informations et sa présentation est compacte. Dans [Mangu *et al.*, 2000], on propose un algorithme de minimisation du taux d'erreur appliqué sur les treillis de mots et montre que l'utilisation de treillis de mot permet un meilleur gain de performance que la minimisation du taux d'erreur appliqué sur la liste des N-meilleures hypothèses.

Rappelons que notre objectif est d'exploiter les multiples unités dans le système de reconnaissance automatique de la parole. Nous souhaitons combiner les treillis des systèmes à base de différentes unités lexicales et sous-lexicales pour ensuite décoder de meilleures hypothèses.

Plusieurs travaux sur la combinaison de systèmes ont utilisé les treillis et proposé des méthodes de combinaison. Dans [Meng *et al.*, 2008], la performance globale du système de reconnaissance de mots clés en langue chinoise (*spoken term detection system*) est améliorée en utilisant la méthode de combinaison des treillis d'hypothèses issus de plusieurs systèmes basés sur différentes unités de modélisation : mot, syllabe sans ton et syllabe avec ton. Les treillis de différentes unités sont d'abord convertis vers une sous-unité commune avant d'être fusionnés pour construire un grand treillis. L'exploitation du treillis fusionné donne une meilleure

performance que l'utilisation de chaque treillis séparé.

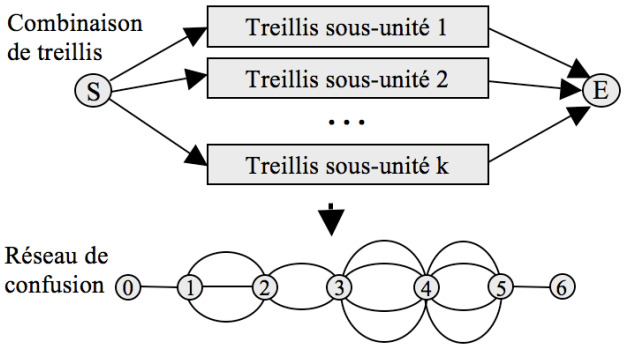


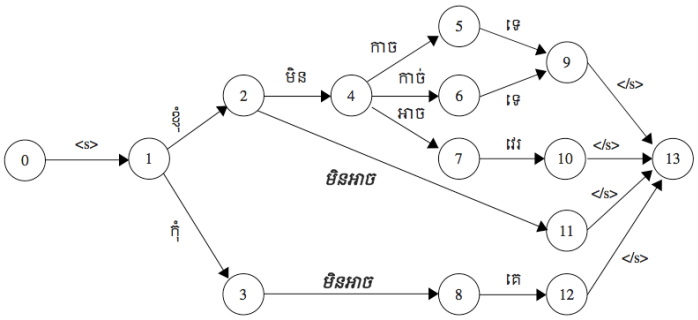
FIGURE 3.2 – *Combinaison des treillis de sous-unités et le réseau de confusion obtenu*

Dans cette thèse, nous utilisons une méthode similaire pour combiner les treillis à base des différentes unités lexicales et sous-lexicales. Cette méthode a été présenté dans [Le *et al.*, 2008] en collaboration avec V-B. Le, ancien Post-doc au LIG, actuellement chercheur au LIMSI.

La combinaison des treillis, montrée dans la figure 3.2, s'effectue en plusieurs étapes :

- **Décomposition** : les treillis basés sur les différentes unités doivent être convertis en une sous-unité unique et commune avant de pouvoir être fusionnés.
- **Combinaison** : les treillis décomposés sont ensuite combinés pour former un grand treillis. Cette opération peut être considérée comme "l'union" des treillis. Nous créons un nouveau treillis ayant comme nœud initial S et nœud final E. Nous relions ensuite S avec tous les nœuds initiaux de tous les treillis à combiner et les nœud finaux de tous les treillis à combiner avec E.
- **Re-scoring** : le treillis combiné peut être éventuellement re-scoré avec un modèle du langage d'ordre plus grand pour optimiser les scores du modèle du langage.
- **Décodage** : le grand treillis est converti en réseau de confusion (confusion network) avant de décoder les hypothèses finales.

La figure 3.3 est un exemple de décomposition d'un treillis généré par le système de reconnaissance automatique de la langue khmère.



(a) treillis contenant des unités lexicales

(b) treillis après la décomposition en unités sous-lexicales

(c) Réseau de confusion correspondant

FIGURE 3.3 – *Exemple de notre décomposition en treillis*

Nous avons dans le treillis original (a), des nœuds qui contiennent les mots composés qui doit être décomposé si nous voulons obtenir un réseau de confusion en terme d'unités sous-lexicales. La décomposition d'un nœud consiste à le fractionner et à ajouter un ou plusieurs nœuds avec les sous-unités et les arcs pour relier ces nouveaux nœuds. Le nombre de nouveau nœuds à ajouter dépend de nombre de sous-unités dans le mot à décomposer. Il faut également redistribuer les scores et les informations temporelles de l'unité originale vers ces sous-unités venant d'être ajoutées.

Plus précisément, la décomposition d'un treillis peut être décrite de la façon suivante :

- à partir d'une liste de mots à décomposer et de leur décomposition, tous les mots décomposables dans le treillis sont identifiés,
- chaque mot identifié est ensuite décomposé en une séquence de sous-unités. Les nouveaux nœuds et arcs sont ajoutés dans le treillis en fonction du nombre de sous-unités contenues dans le mot,
- les scores acoustiques et les informations temporelles (durée, étiquette temporelle) du mot initial sont redistribués proportionnellement aux nouveaux nœuds venant s'ajouter dans le treillis,
- Une approximation est faite pour le score du modèle de langage. Le score du modèle du langage du mot original est transféré vers la première sous-unité. Les autre sous-unités qui suivent reçoivent un score égal à 0. En faisant cette approximation, on suppose qu'il existe un seul chemin partant de la première sous-unité vers la dernière sous-unité du mot.

Puisque les treillis sont générés à partir de différents systèmes avec différentes unités lexicales et sous-lexicales, une étape de normalisation est nécessaire. La probabilité à posteriori des phrases peut être normalisée par la somme des probabilité à posteriori de toutes les phrases dans le treillis. Dans [Mangu *et al.*, 2000], l'équation 3.2 de normalisation est donnée par :

$$P(W^k|A) \approx \frac{P(W^k)P(A|W^k)}{\sum_{k=1} P(W^k)P(A|W^k)} \tag{3.2}$$

où k s'étend sur l'ensemble des hypothèses générés par le décodeur.

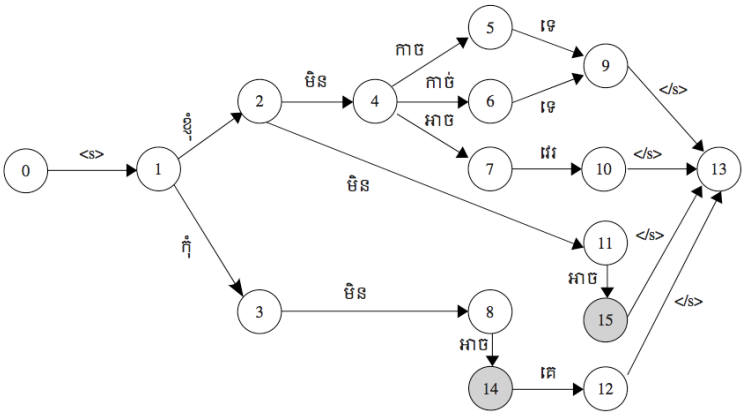
Cette étape de normalisation peut être utilisée dans la méthode de combinaison. Avant de combiner, tous les treillis sont décomposés et normalisés.

La boîte à outils SRILM toolkit (v 1.5.2) fournit un outils *lattice-tool* permettant de faire la décomposition de treillis. Mais avec cet outil, les scores et les informations temporelles (durée, étiquette temporelle) du mot original ne sont attribués qu'à la première sous-unité du mot après sa décomposition. Les autres sous-unités qui suivent, reçoivent un score et une durée égale à 0. Puisque nous passons par le réseau de confusion pour le décodage des hypothèse finales et que l'algorithme qui fait la conversion d'un treillis vers le réseau de confusion prend en compte la durée et l'étiquette temporelle de chaque nœud, la façon d'attribuer des scores de *lattice-tool* peut causer de faux alignements dans le processus de conversion. Figure 3.4 donne un exemple sur la décomposition du treillis (a) de la figure 3.3 par *lattice-tool* et montre le réseau de confusion obtenu à partir de cette décomposition. Deux nouveau nœuds (14 et 15) sont ajoutés dans le treillis et les mêmes étiquettes temporelles que les nœuds numéro 8 et 13 sont respectivement attribuées aux nœudx 14 et 15. Cette façon de décomposer cause un mauvaise alignement au moment de la conversion en réseau de confusion. La méthode décrite précédemment est donc plus adaptée.

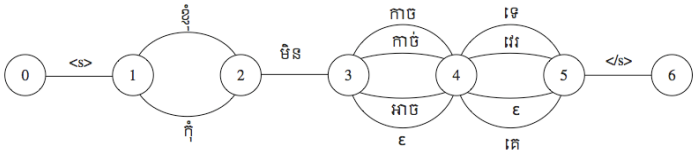
3.5 Expérimentations

L'objectif principal de nos expérimentations est d'exploiter les unités lexicales et sous-lexicales pour la reconnaissance automatique de la parole en utilisant les méthodes de combinaison décrites dans la section précédente. Pour tester et comparer ces méthodes de combinaison, nous mettons en place les tests sur deux langues non segmentées : le khmer et le vietnamien. En khmer et en vietnamien, la segmentation automatique d'un texte en mots n'est pas triviale. Par contre le texte en vietnamien est naturellement segmenté en syllabes. Pour le khmer, la segmentation d'un texte en une sous-unité appelée "clusters de caractères" (CC) est triviale. Nous souhaitons tester cette approche de multiples unités pour la reconnaissance automatique de la parole de ces deux langues.

Pour entraîner les modèles de langage hybrides, nous créons d'abord des vo-



(a) treillis avec la décomposition par *lattice-tool*



(b) réseau de confusion correspondant

FIGURE 3.4 – Exemple de décomposition en treillis de *lattice-tool*

cabulaires hybrides en ajoutant progressivement les n mots les plus fréquents du corpus de texte (segmenté automatiquement en mots) dans le vocabulaire $V0$ de sous-unités lexicales (syllabe vietnamien ou groupe de caractères khmer). En faisant varier n de 0 à T (T étant la taille du vocabulaire de mots), différents vocabulaires hybrides sont créés : $V0, V1k, V5k, V10k, V15k, V20k$ ...VTk. Le corpus d'apprentissage est ensuite automatiquement re-segmenté à l'aide de chaque vocabulaire hybride pour entraîner les modèles de langage hybrides. La performance de ces modèles est évaluée dans le système de reconnaissance automatique de la parole.

Au niveau des sorties de systèmes, la méthode ROVER et la combinaison de treillis sont utilisées pour combiner la sortie du système à base de mot $Vmot$ et celle du système à base de sous-mot $V0$. Dans l'approche ROVER, chaque système décode d'abord une liste des N-meilleurs hypothèses ($N = 20$). Les hypothèses en mot sont automatiquement décomposées en sous-mots pour obtenir une unité commune. Nous combinons ensuite toutes les hypothèses (40 hypothèses) en utilisant l'algorithme ROVER pour obtenir de nouvelles hypothèses.

En ce qui concerne la combinaison des treillis, chaque système doit décoder les hypothèses sous forme de treillis. Les treillis de mot sont décomposés en treillis de sous-mots en utilisant le mécanisme de décomposition de treillis décrit dans la section 3.4.2. La combinaison des sorties du système $V0$ et $Vmot$ donne un grand treillis qui est ensuite converti en réseau de confusion pour décoder les meilleures hypothèses.

Dans les sections suivantes, nous présentons nos expérimentations sur la langue khmère et la langue vietnamienne. Nous décrivons d'abord la configuration de nos systèmes de reconnaissance. Les résultats obtenus avec les différentes méthodes et les analyses sur ces résultats sont également présentés.

3.5.1 Application à la langue khmère

Système de RAP khmer

Le système de reconnaissance pour le khmer est développé avec Sphinx3. La topologie des modèles est un HMM de 3 états avec 16 Gaussiennes par état. Le vecteur de paramètres contient 13 MFCCs, ses premieres et secondes dérivées. Le corpus de texte est collecté à partir des sites des journaux en ligne et contient environ 15 millions de mots. Le vocabulaire de mots contient 20k mots. La segmentation de ce dictionnaire en cluster de caractères donne un vocabulaire de 3500 clusters de caractères. Plus de détails sur la configuration du système khmer ainsi que la description des ressources utilisées sont données dans le chapitre 2.

Les tests sont effectués sur deux jeux de test qui ont différents taux de mots hors vocabulaire. Le jeu de test *evalKh1* contient 162 phrases et a un taux de mots hors vocabulaire de 3%. Le deuxième jeu de test *evalKh2* qui est un sous-ensemble de *evalKh1* contient 50 phrases et a un taux de mots hors vocabulaire de 8%.

Nous utilisons le taux d'erreur de cluster de caractères (CCER) pour l'évaluation du système de RAP, car la segmentation de mots et syllabes khmers n'est pas triviale et les erreurs de segmentation pourraient empêcher une comparaison correcte entre les systèmes.

Modèles hybrides

Dans cette expérimentation, des modèles hybrides pour le khmer sont créés en combinant les mots et les clusters de caractères (CC). En augmentant n de 0 à 20k (la taille du vocabulaire mot du khmer), 5 différents vocabulaires hybrides sont créés : $V1k, V5k, V10k, V15k, V20k$. Nous avons comme référence deux modèles simples : $V0$ fondé sur les cluster de caractères et $Vmot$ fondé sur les mots seuls.

La figure 3.5 donne les résultats de ces modèles hybrides pour la reconnaissance automatique du khmer. Nous pouvons constater que la performance s'améliore au fur et à mesure que les mots sont introduits dans le vocabulaire. Dans notre cas, les modèles hybrides sont légèrement meilleurs que le modèle simple à base de mot ou de cluster de caractères seuls.

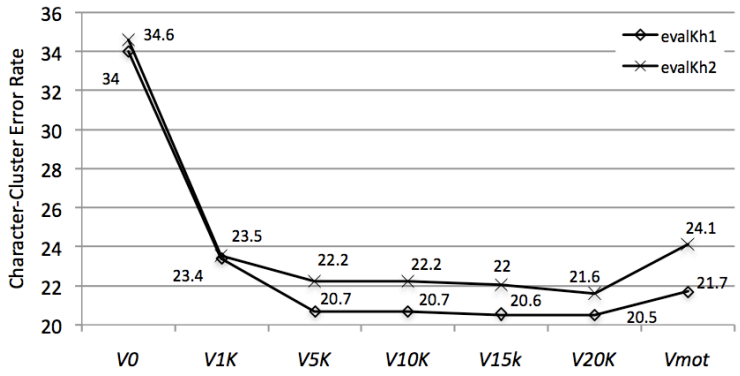


FIGURE 3.5 – *Performance de modèles hybrides CC + n Mots les plus fréquents pour la RAP khmer*

Combinaison par ROVER

Le tableau 3.1 présente les résultats de la combinaison par la méthode ROVER sur les N-meilleurs hypothèses du système khmer à base de mot et celui à base de cluster de caractères.

Bien que le CCER Oracle montre le potentiel de cette approche de combinaison, la méthode de vote simple ne permet pas d'améliorer la performance. Une raison de cet échec est également dûe aux performances trop faibles du modèle

Système	CCER	CCER Oracle
Mot ($Vmot$)	21.7%	-
CC ($V0$)	34%	-
Mot 20 meilleurs	22.4%	15.6%
CC 20 meilleurs	35.8%	27.7%
Mot + CC 40 meilleur	23.1%	11.8%

TABLE 3.1 – *Combinaison des N-meilleurs hypothèse par ROVER sur evalKh1 du khmer.*

$V0$ à base de cluster de caractères par rapport au modèle $Vmot$, ce qui rend la fusion inefficace.

Combinaison des treillis

Dans cette expérimentation, le treillis décodé par le système à base de mots est combiné avec celui généré par le système à base des clusters de caractères. Pour voir la performance du mécanisme de décomposition de treillis et le décodage par le réseaux de confusion (CN pour confusion network), nous essayons de décoder les treillis de mot en les décomposant puis en les convertissant en réseaux de confusion. Nous testons également l'approche de re-scoring du treillis. Dans notre cas, nous utilisons un modèle 8-grammes de cluster de caractères (approximativement équivalent à un modèle bi-gramme de mots en khmer) pour rescorer le treillis avant de décoder par le réseau de confusion.

Le tableau 3.2 donne les résultats de la décomposition du treillis de mots et le décodage par le réseau de confusion, la combinaison des treillis, ainsi que le re-scoring des treillis par un modèle du langage d'ordre plus élevé. Nous observons que la décomposition de treillis et le décodage améliorent légèrement le taux d'erreur CCER mais de façon non significative. La combinaison de treillis mot avec le treillis cluster de caractères ne donne pas véritablement une meilleure performance dans le cas de notre système khmer.

La figure 3.6 présente une vue d'ensemble sur les différents résultats : les modèles simples à base de mots $Vmot$ ou des cluster de caractères $V0$, la combinaison par ROVER, la combinaison des treillis avec ou sans la normalisation de score et le rescoring de treillis avec un modèle du langage de sous-unité (CC) d'ordre plus

Treillis	Modèle de langage utilisé pour re-scorer	CCER sur evalKh1 (3% OOV)	CCER on evalKh2 (8% OOV)
Mot (V_{mot}),	-	21,7%	24,1%
Mot (V_{mot}) décomposé en CC	CC 8-grammes	28,8%	30,2%
CC(V_0)	-	34%	34,6%
CC(V_0)	CC 8-grammes	23,7%	25,1%
$V_{mot} + V_0$	-	25,8%	27,4%
$V_{mot} + V_0$	CC 8-grammes	**20,8%**	**22,3%**

TABLE 3.2 – *Combinaison des treillis.*

élevé. Les résultats montrent que le modèle mot reste très performant pour la modélisation. La meilleure performance est cependant obtenue à partir du rescoring de treillis combinée avec un modèle de langage de 8-gramme de cluster de caractères.

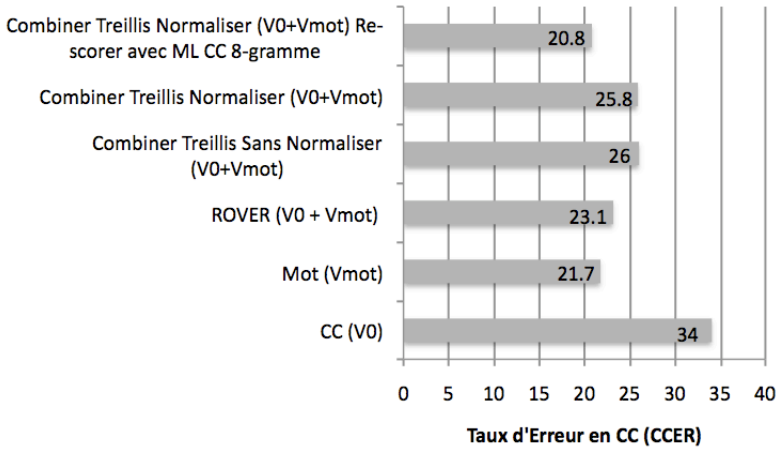


FIGURE 3.6 – *Comparaison des performances des méthodes de combinaison sur evalKh1*

3.5.2 Application à la langue vietnamienne

Système de RAP vietnamien

Les expérimentations sur la langue vietnamienne ont été principalement menées avec V-B. Le, ancien Post-doc au LIG, actuellement chercheur au LIMSI. Le système de reconnaissance automatique de la parole en vietnamien a été développé dans le cadre de sa thèse de doctorat au LIG [Le, 2006]. Ma contribution sur cette expérimentation concerne les modèles hybrides et la combinaison par ROVER

tandis que le système de RAP vietnamien de référence et les expérimentations de combinaison de treillis sont la contribution de V-B Le.

Le système vietnamien est basé sur le décodeur IBIS de Janus [Rogina et Waibel, 1995] dévelopé au laboratoire ISL. La topologie du modèle est un HMM de 3 états avec 32 Gaussiennes par état. Le vecteur de paramètres contient 13 MFCCs, ses premieres et secondes dérivées, l'énergie et le taux de zero-crossing. Les détails du système vietnamien sont données dans [Le *et al.*, 2008].

Le test est effectué sur un corpus qui contient 277 phrases dans le domaine des news. La performance du système est évaluée en terme de taux d'erreur de syllabe (ou SLER pour Syllable Error Rate) puisque la segmentation automatique en mots du texte vietnamien n'est pas un problème trivial et les erreurs de segmentation peuvent empecher une bonne comparaison entre les sorties des différents systèmes.

Modèles hybrides

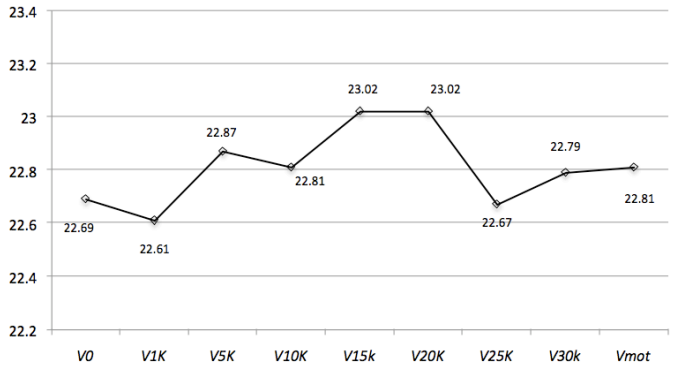


FIGURE 3.7 – *Performance de modèles hybrides : Syllabes + N Mots les plus fréquents en Vietnamien*

Des modèles hybrides pour le vietnamien sont créés en combinant les mots et syllabes. En augmentant n de 0 à 35k (la taille de vocabulaire mot du vietnamien), 8 différents vocabulaires hybrides sont créés : $V1k, V5k, V10k, V15k$, $V20k, V25k, V30K$ et $V35k$. Le vocabulaire $V35k$ correspond en fait au vocabulaire mot. Nous avons deux modèles simples de référence : $V0$ fondé sur les syllabes et $Vmot$ noté aussi $V35k$.

La figure 3.7 donne les résultats de test qui utilisent ces modèles hybrides pour la reconnaissance automatique du vietnamien. Les résultats montrent que le modèle à base de syllabe donne la meilleure performance en gardant une petite taille de vocabulaire. Il est à noter que, en vietnamien, un mot se compose en moyenne de 1,6 syllabes [Le, 2006].

Combinaison par ROVER

Le talbeau 3.3 donne les résultats de la combinaison par la méthode ROVER sur les N-meilleures hypothèse du système vietnamien à base de mots et de celui à base de syllabes comparé aux systèmes de référence $V0$ et $Vmot$. Les résultats montrent que la méthode de combinaison par ROVER améliore la performance dans le cas de la langue vietnamienne. L'amélioration du taux d'oracle peut être aussi observée.

Système	CCER	CCER Oracle
Mot ($Vmot$)	22.69%	-
CC ($V0$)	22.81%	-
Mot 20 meilleurs	22.80%	14.0%
CC 20 meilleurs	22.90%	14.8%
Mot + CC 40 meilleur	21.40%	11.1%

TABLE 3.3 – *Combinaison des N-meilleurs hypothèse par ROVER (vietnamien).*

Combinaison des treillis

Le figure 3.8 montre les résultats de combinaison des treillis de mots avec les treillis de syllabes comparés avec les autres méthodes appliquées sur le vietnamien. Nous pouvons constater que la méthode de combinaison de treillis permet d'améliorer la performance du système. Le gain de performance est obtenu même sans appliquer la normalisation de treillis.

De manière globale, nous pouvons conclure que les deux méthodes de combinaison de systèmes : ROVER et la combinaison de treillis améliorent de façon significative la performance par rapport aux modèles simples. Cela montre le potentiel de l'utilisation des multiples-unités dans la modélisation de la langue

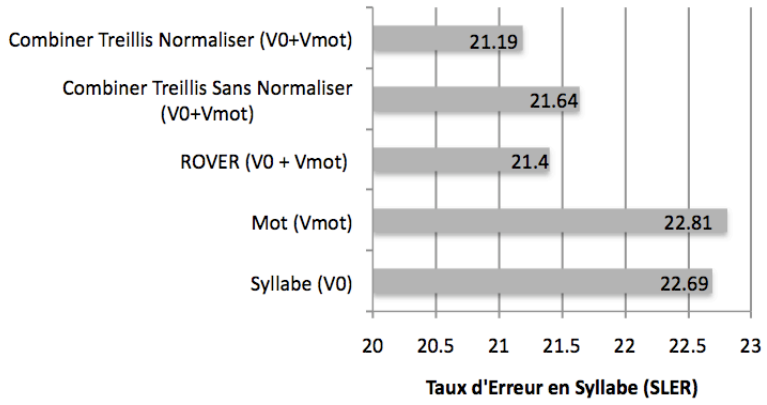


FIGURE 3.8 – *Comparaison de performance de méthodes de combinaison*

vietnamienne.

3.6 Conclusion

Dans ce chapitre, nous avons présenté trois méthodes pour exploiter les multiples unités dans la reconnaissance automatique de la parole pour les langues peu dotées et non segmentées. Pour essayer d'exploiter plusieurs vues sur les données textuelles au niveau de la modélisation du langage, nous proposons des modèles hybrides qui utilisent à la fois des unités lexicales et sous-lexicales dans un vocabulaire hybride. Au niveau du système, nous proposons deux méthodes pour combiner des sorties de systèmes fondés sur les différentes unités. Nous appliquons ces méthodes à la reconnaissance automatique de la parole de deux langues peu dotées et non segmentées : le khmer et le vietnamien. Les résultats d'expérimentations sur le khmer montrent le potentiel du modèle du langage hybride qui réagit mieux au taux de mots hors vocabulaire et permet d'avoir une vocabulaire de reconnaissance de taille relativement petite comparée à une unité comme le mot seul. Les méthodes de combinaison au niveau du système, le ROVER et la combinaison de treillis améliorent la performance dans le cadre des expérimentations sur la langue vietnamienne. Ces améliorations montrent les potentiels de l'utilisation des unités multiples (lexicales et sous-lexicales) dans la reconnaissance automatique de la parole dans le contexte des langues peu dotées.

Chapitre 4

Segmentation multiple pour la modélisation statistique du langage

4.1 Introduction

Tandis que le manque de données textuelles a un impact sur la performance des modèles de langage, pour les langues sans segmentation entre les mots les erreurs introduites par la segmentation automatique peuvent rendre ces données encore moins exploitables. Lors de l'apprentissage d'un modèle de langage, les n-grammes de mots non trouvés dans le corpus d'apprentissage peuvent l'être à cause d'erreurs de segmentation mais aussi parce qu'une séquence de caractères peut avoir plusieurs segmentations possibles mais une seule segmentation a été considérée dans le corpus d'apprentissage. Dans un objectif consistant à mieux exploiter une faible quantité de données textuelles en utilisant différentes vues sur données, nous proposons, dans ce chapitre, une méthode qui effectue des segmentations multiples sur le corpus d'apprentissage au lieu d'une segmentation unique classique. Cette méthode de segmentation multiple basée sur des automates d'état finis permet de générer toutes les segmentations possibles à partir d'une séquence de caractères et nous pouvons ensuite en extraire les n-grammes pour apprendre le modèle de langage. Elle permet de trouver des n-grammes non obtenus par la segmentation unique et d'ajouter de nouveaux n-grammes dans le modèle de

langage. Ceci peut être vu comme une sorte de sur-génégation des n-grammes à partir d'un corpus de texte. Cette approche par segmentation multiple est comparée avec la méthode classique de segmentation unique dans l'apprentissage des modèles de langage pour les systèmes de reconnaissance automatique de la parole en langue khmère, laotienne, thaïe et vietnamien.

4.2 Segmentation multiple

4.2.1 Motivations

Pour apprendre un modèle de langage trigramme, il faut d'abord compter le nombre de trigrammes, de bigrammes et d'unigrammes à partir du corpus d'apprentissage avant de pouvoir estimer leurs probabilités. Le comptage d'un trigramme consiste en sa recherche dans la totalité du corpus de texte.

Si nous voulons chercher une séquence de mots $W = \{w_i w_j w_k\}$ dans une phrase non segmentée en mot, nous effectuons généralement d'abord la segmentation en mots de la phrase. Si la séquence de mot recherchée est effectivement dans la phrase mais que nous ne pouvons pas la retrouver, cela peut provenir de différentes raisons. Il peut y avoir des erreurs de segmentation, ou bien il peut exister plusieurs façons possibles pour segmenter la phrase en question mais un seul choix a été considéré par la méthode de segmentation employée. A cause de la définition ambiguë du "mot" et l'existence de plusieurs conventions de segmentions dans une langue, il est assez souvent possible de segmenter une phrase de plusieurs façons. A titre d'exemple, dans la tâche de segmentation de la campagne SIGHAN (SIGHAN Chinese Language Processing Bakeoff), il existe au moins quatre spécifications de segmentation de texte de chinois mandarin proposées par différentes organisations, à savoir l'Université de Hong Kong, Microsoft Research (Beijing) l'Université de Beijing et Academia Sinica.

Une autre façon de retrouver une séquence de mots dans une phrase non segmentée est de ne pas segmenter la phrase mais nous pouvons plutôt essayer de retrouver la séquence de caractères qui forme W directement dans le flux de caractères de la phrase. L'avantage de cette méthode est que nous n'avons plus besoin

de la segmentation. Cependant, chercher la séquence de caractères qui forme W dans une phrase non segmentée présente le risque de trouver cette séquence dans un autre contexte. Par exemple, si $W = \{w_i, w_j, w_k\}$ forme une séquence de caractères sans espace $C = c_i c_j c_k | c_l c_m c_n c_p | c_q c_r$ (| indique la frontière entre les mots à titre indicatif), la recherche de la séquence C dans une phrase non segmentée $S = ...c_{i-3}c_{i-2}c_{i-1}\mathbf{c_i c_j c_k}|\mathbf{c_l c_m c_n c_p}|\mathbf{c_q c_r}c_{r+1}c_{r+2}...$ permet de trouver évidemment cette séquence C. Mais il se peut que dans la réalité, il n'est pas correct de segmenter la phrase S en $\{...c_{i-3}c_{i-2}c_{i-1}|\mathbf{c_i c_j c_k}|\mathbf{c_l c_m c_n c_p}|\mathbf{c_q c_r}|\ c_{r+1}c_{r+2}...\}$ mais plutôt en $\{...|c_{i-3}c_{i-2}c_{i-1}\mathbf{c_i c_j c_k}|\mathbf{c_l c_m c_n c_p}|\mathbf{c_q c_r}|c_{r+1}c_{r+2}...\}$ qui donne $\{w_{j-1}, w_j, w_k\}$ au lieu de $W = \{w_i w_j w_k\}$.

Pour illustrer la différence entre les deux approches de recherche de séquence de mots dans le texte non segmenté, nous effectuons une expérimentation qui consiste à compter les séquences de trigrammes dans notre corpus de texte khmer avec ces deux méthodes. Les séquence de trigrammes de mots sont générées à partir d'un corpus de développement (150 phrases) segmenté manuellement. Notre corpus d'apprentissage de texte khmer segmenté automatiquement en mots (à base un dictionnaire de 20k mots) contient environ 15 millions de mots. Dans un premier temps, nous comptons le nombre d'occurrences de nos trigrammes dans le corpus segmenté en mot. Dans un second temps, le comptage est effectué au niveau du flux de caractères dans le corpus non segmenté.

Nous vérifions aussi les résultats du comptage pour les n-grammes d'ordre inférieur. Si un trigrammes $w_i w_j w_k$ n'est pas trouvé dans le corpus, nous effectuons le repli vers un bigramme et un unigramme, c'est-à-dire que si $w_i w_j w_k$ n'existe pas, nous essayons de compter $w_j w_k$ puis w_k. Avec le comptage au niveau de flux de caractères, on notera qu'il existe une option plus intéressante pour faire le repli au cas où un trigramme entier n'est pas trouvé dans le corpus. Au lieu de faire un repli entièrement vers le bigramme (un mot est directement enlevé), nous pouvons essayer d'abord de faire un repli de n caractères plus petits qu'un mot (la moitié du mot par exemple). Dans le cadre de notre expérimentation, nous effectuons un repli de 4 caractères maximum sur le trigramme et nous considèrons que la totalité de ce trigramme est trouvé si nous trouvons le trigramme avec moins 4 caractères (voir dernier colonne de table 4.1).

Ordre ngramme	Comptage Mot	Comptage Caractère	Comptage Caractère - 4
3	1248 (38.14%)	1507 (46.06%)	2175 (66.47%)
2	1437 (43.92%)	1320 (40.34%)	919 (28.09%)
1	563 (17.21%)	425 (12.99%)	176 (5.38%)
OOV	24 (0.73%)	20 (0.61%)	2 (0.06%)
Totale	3272	3272	3272

TABLE 4.1 – *Comparaison de différentes technique de comptage des trigrammes.*

Le tableau 4.1 présente les résultats de cette expérimentation. Nous pouvons observer que le comptage au niveau des caractères sans faire la segmentation permet de retrouver plus de trigrammes que le comptage au niveau des mots qui nécessite la segmentation de corpus. Comparée à la technique de comptage au niveau mot classique, ces deux dernières méthodes de comptage sont particulièrement intéressantes pour la modèlisation statistique du langage, car plus de trigrammes implique moins de repli sur les bigrammes dans l'estimation de la probabilité d'une séquence de mots. Ces méthodes sont d'abord intéressantes pour les langues non segmentée en mot car elles apportent une solution afin de palier au problème de la segmentation. Ensuite, le comptage des séquences de caractères est particulièrement utile pour les langues peu dotées car cela peut être considéré comme l'exploitation de plusieurs vues sur les mêmes données.

Dans la pratique, pour calculer un modèle de trigramme de mots, il faut dispenser d'un vocabulaire de mots et d'un corpus segmenté en mot pour faire le comptage des trigrammes. Pour appliquer notre technique de comptage au niveau des caractères, qui suppose de connaître a priori la chaîne (et donc le tri-gramme) recherchée, il faut que nous ayons la liste de tous les trigrammes à rechercher dans le corpus non segmenté. Or, sur un vocabulaire de taille N, nous aurons N^3 trigrammes possibles à compter. Il est difficile voire impossible d'envisager la recherche de N^3 trigrammes si N est de l'ordre de 20k, la taille normale d'un vocabulaire utilisé dans la reconnaissance automatique de la parole grand vocabulaire.

D'un autre point de vue, cette technique peut être assimilée au comptage de trigrammes en segmentant le corpus pour obtenir toutes les segmentations possibles. Pour essayer d'appliquer cette technique dans l'estimation de trigrammes, nous proposons une méthode de segmentation multiple qui permet d'approcher le comptage au niveau des caractères.

4.2.2 Estimer les trigrammes avec la segmentation multiple

Contrairement à la méthode de segmentation classique qui recherche dans une séquence de caractères la meilleure segmentation selon un critère d'optimisation, notre approche par segmentation multiple cherche à générer toutes les séquences de mots valides à partir d'une séquence de caractères. C'est à partir de toutes ces séquences de mots que des n-grammes seront comptés pour l'apprentissage du modèle de langage.

Phrase	ព្រះពុទ្ធជាព្រះបរមគ្រូនៃយើង								3-grams Count
Segmentation 1	ព្រះពុទ្ធ w_1		ជា w_2	ព្រះ w_3	បរមគ្រូ w_4		នៃ w_5	យើង w_6	$w_1\ w_2\ w_3$ $w_2\ w_3\ w_4$ $w_3\ w_4\ w_5$ $w_4\ w_5\ w_6$
Segmentation 2	ព្រះពុទ្ធ w_1		ជា w_2	ព្រះ w_3	បរម w_7	គ្រូ w_8	នៃ w_5	យើង w_6	$w_2\ w_3\ w_7$ $w_3\ w_7\ w_8$ $w_7\ w_8\ w_5$
Segmentation 3	ព្រះ w_3	ពុទ្ធ w_9	ជា w_2	ព្រះ w_3	បរម w_7	គ្រូ w_8	នៃ w_5	យើង w_6	$w_3\ w_9\ w_2$ $w_9\ w_2\ w_3$
Traduction	Le bouddha est notre maître suprême								

FIGURE 4.1 – *Exemple de segmentation multiple sur une phrase en khmer.*

La figure 4.1 montre un exemple de la segmentation multiple d'une phrase en khmer. Nous montrons trois segmentations possibles d'une séquence de caractères en khmer. La segmentation 1 correspond à la segmentation unique de type « plus longue chaîne d'abord ». Dans le cas de segmentation unique (segmentation 1), nous obtenons 4 tri-grammes. Si nous appliquons la segmentation multiple, nous aurons au total 9 tri-grammes. 5 nouveaux tri-grammes sont obtenus à partir des deux autres segmentations possible (segmentation 2 et 3). Il est à noter que nous ne comptons qu'une seule fois un n-gramme, même s'il se présente plusieurs fois dans les différentes segmentations.

Par rapport à la segmentation unique, la segmentation multiple permet d'obtenir plus des n-grammes. Nous pouvons diviser ces nouveaux n-grammes en trois différentes catégories :

1. des n-grammes de mots qui sont effectivement pertinents pour le corpus d'apprentissage d'origine, non segmenté, mais à cause d'erreurs introduites par la segmentation unique, ils ne sont pas retrouvés lors d'une segmentation unique.

2. des n-grammes de mots qui sont effectivement pertinents pour le corpus d'apprentissage d'origine, non segmenté, mais comme une séquence de caractères peut avoir plusieurs segmentations correctes et qu'un seul choix est considéré lors de la segmentation unique, ils ne sont pas alors retrouvés lors d'une segmentation unique.

3. des n-grammes de mots qui ne sont pas pertinents pour le corpus d'apprentissage. Dans ce cas, la segmentation multiple génère ces n-grammes parce qu'il est possible de segmenter entièrement une phrase en une séquence de mots valide (même si les segmentations sont incorrectes) mais aussi parce que notre méthode de segmentation multiple permet également de générer localement les séquences de mots dans une phrase en marquant les parties restantes qui ne correspondent pas aux mots valide comme « mot inconnu *unk*».

Les n-grammes de catégorie 1 et 2 sont des n-grammes potentiellement utiles pour la modélisation du langage car il s'agit de séquences de mots valides de la langue et ils sont effectivement présents dans le corpus d'apprentissage. Les n-grammes de catégorie 3 peuvent perturber la modélisation.

Notre objectif est de générer les n-grammes qui sont potentiellement utiles pour la modélisation en limitant la génération des n-grammes non pertinents. Nous développons un outil de segmentation multiple qui permet de générer les n-*meilleures* segmentations à partir d'une séquence de caractères donnée en entrée. Nous allons décrire dans la section suivante comment la segmentation multiple est implémentée.

4.2.3 Génération des segmentations multiples par automates d'état fini

Dans cette section, nous présentons notre algorithme basé sur les automates d'état fini pour générer les segmentations multiples à partir d'une séquence de caractères. Il existe des travaux qui utilisent les automates d'état fini pour faire la segmentation unique. Nous pouvons citer [Lee *et al.*, 2003] et [Zitouni, 2006] qui utilise cette technique pour segmenter un texte arabe.

Cadre mathématique

Supposons que nous avons en entrée une chaîne de caractère $S = c_1c_2...c_Q$ de longueur Q et une séquence de mots $m_k = \{w_1^k w_2^k ... w_{L_k}^k\}$ de longueur L_k où m_k est l'une des segmentations possibles M de la chaîne S ($m_k \in M$). Parmi toutes les segmentations possibles M de la chaîne S, la meilleure segmentation $\hat{m}$ est celle qui a la plus grande probabilité :

$$P(\hat{m}) = \max_k P(m_k) \tag{4.1}$$

La probabilité $P(m_k)$ peut être estimée en utilisant un modèle du langage n-gramme :

$$P(m_k) \simeq \prod_{i=1}^{L_k} P(w_i^k | w_{i-1}^k, w_{i-2}^k, ...w_{i-N+1}^k) \tag{4.2}$$

Parmi toutes les segmentations possibles de S, les meilleures segmentations sont l'ensemble des segmentations $\hat{M} = \{\hat{m}_1, \hat{m}_2, ...\hat{m}_{nbest}\}$ où $\hat{m}_i$ est la $i^{ème}$ meilleure segmentation selon la fonction $P(\hat{m}_i)$.

Implémentation

Générer toutes les segmentations possibles à partir de la chaîne S nécessite beaucoup de calcul et nous ne pouvons générer explicitement toutes les segmentations de S pour pouvoir en extraire les les meilleures segmentations $\hat{M}$. Nous allons le faire implicitement en utilisant les automates d'états finis.

L'algorithme décrit précédemment peut être décomposé en plusieurs étapes simples en utilisant les automates d'états finis. La chaîne de caractères en entrée est représentée sous forme d'un accepteur I, chaque arc de I représente un caractère. Etant donnée une liste finie des mots (un vocabulaire), nous pouvons construire un transducteur de mots T qui, en se composant avec I génère un treillis de mots représentant toutes les segmentations possibles en mots de la chaîne en entrée. Comme pour tous les systèmes basés sur un vocabulaire, nous devons traiter les mots hors vocabulaire. Nous utilisons un modèle des mots inconnus sous forme d'un automate qui analyse n'importe quelle séquence de caractères et génère un symbole unique *unk* pour représenter les mots inconnus. Cet automate est obtenu par une opération de fermeture de *Kleene* (parfois appelée étoile de *Kleene*) sur l'ensemble des caractères. Le transducteur de mots T peut être alors décrit comme $T = (\texttt{WD} \cup \texttt{UNK})^*$ où `WD` est un automate qui représente le vocabulaire et `UNK` représente le modèle des mots inconnus. Ici, $\cup$ and $*$ sont respectivement l'opération de l'union et la fermeture de *Kleene*.

Un modèle du langage L est utilisé pour attribuer les scores au treillis de toutes les segmentations possibles obtenu par la composition du transducteur T avec la chaîne I. Un modèle de langage peut être représenté sous forme d'un automate d'état fini où un état est un contexte observé du modèle. Le nombre d'états est limité par V^{N-1}, où V est la taille du vocabulaire et N l'ordre n-gramme du modèle de langage. Un arc sortant d'un état S_i représente un symbole c dans le contexte de S_i et le poids est la probabilité $P(c|S_i)$.

La décomposition de la chaîne de caractères I avec le transducteur de mots T donne un transducteur qui représente toutes les segmentations en mots possible de I. Ce transducteur est ensuite composé avec le modèle de langage L pour obtenir un transducteur scoré. Le chemin ayant le meilleur score est la meilleure segmentation $\hat{m}$ comme définie dans l'équation (4.1). La meilleure segmentation peut être décrite formellement comme :

$$\hat{m} = f(I \circ T \circ L) \tag{4.3}$$

où $\circ$ est l'opération de composition et f est la fonction qui décode le meilleur chemin.

Les *n-meilleures* segmentations $\hat{M}$ sont obtenues par le décodage du treillis final pour sortir les *n-meilleurs* chemins. Ces *n-meilleures* segmentations seront utilisées dans la comptage des n-gramme pour la modélisation du langage comme expliqué dans le figure 4.1.

Dans notre cas, il est important de noter qu'un simple modèle de langage unigramme est utilisé. Ce modèle est estimé à partir d'un petit corpus de texte segmenté automatiquement en mots en utilisant la méthode à base de dictionnaire. Nous utilisons un modèle unigramme parce qu'un modèle d'ordre plus élévé entraîné à partir d'un corpus segmenté automatiquement pourrait influencer le score dans le treillis et les *n-meilleures* segmentations obtenues seraient trop proches de la segmentation donnée par la méthode de segmentation unique utilisée pour apprendre le ML.

L'implémentation de cet algorithme de segmentation multiple à base des automates d'état fini est effectuée en utilisant la boîte à outils AT&T FSM (Finite-State Machine Library) [Mohri *et al.*, 1998]. Le modèle de langage L est créé à l'aide de la librairie GRM (Grammar Library). L'annexe 3 donne les détails sur l'implémentation de notre algorithme de segmentation multiple avec la boîte à outils AT&T FSM.

4.2.4 Les travaux liés

Dans [Zhang *et al.*, 2008b], le problème de segmentation est étudié pour un système de traduction automatique chinois-anglais. Comme il existe plusieurs méthodes et conventions de segmentation en chinois, 4 méthodes de segmentation automatique issues de différentes conventions de segmentation ont été expérimentées dans le système de traduction. Les résultats ont montré la corrélation entre la performance de segmentation et la performance du système de traduction. L'amélioration significative de la performance est obtenue en faisant une interpolation linéaire des modèles de traduction appris sur des données segmentées avec différentes conventions de segmentation.

Un principe similaire est utilisé dans [Do *et al.*, 2009] pour la traduction auto-

matique vietnamien-français. Comme en vietnamien il est possible de segmenter le texte en mots ou en syllabes, l'utilisation de deux tables de traduction dans le système de traduction statistique a été proposée vietnamien-français. La table de traduction à base de mot et celle à base de syllabe sont exploitées simultanément par le système et l'utilisation de multiples tables donne une performance plus élevé que pour chaque table considérée séparément.

4.3 Expérimentations

Pour comparer la performance de segmentation multiple et la segmentation unique à base de dictionnaire dans la modélisation statistique du langage, nous apprenons des modèles de langage trigrammes à partir des corpus d'apprentissage segmentés en mots en utilisant ces deux approches de segmentation.

Premièrement, un modèle de langage appelé lm_{unique} est entraîné avec le corpus de texte segmenté avec la méthode de segmentation classique à base de dictionnaire. Ensuite, nous apprenons les autres modèles à partir du corpus segmenté avec la segmentation multiple en choisissant comme nombre de *n-meilleures* segmentations pour chaque phrase de corpus entre 2, 5, 10, 50 et 100. Après l'apprentissage, nous obtenons plusieurs modèles de langage : lm_2, lm_5, lm_{10}, lm_{50} et lm_{100}. Il est à noter que la segmentation multiple utilise le même dictionnaire que la segmentation unique. Dans ce cas, seuls le compte des bigrammes et des trigrammes évolue, tandis que le nombre d'unigrammes n'est pas modifié entre l'approche classique et l'approche à segmentations multiples.

La performance de chaque modèle issu de ces deux approches de segmentation est évaluée avec plusieurs critères. Nous comparons chaque modèle en terme de nombre de tri-grammes, de couverture en trigrammes (*trigram hits*) sur le corpus de développement segmenté manuellement, de perplexité et de performance pour un vrai système de reconnaissance automatique de la parole. Nos expérimentations sont effectuées sur un groupe de langues en Asie du sud-est : la langue khmère, laotienne, thaïe et vietnamienne. Les caractéristiques communes de ces langues

sont qu'elles sont peu dotées et possèdent un système d'écriture sans séparation explicite entre les mots.

4.3.1 Application à la langue khmère

Notre corpus d'apprentissage de la langue khmère contient 0,5 millions de phrases de type news. Après la segmentation unique automatique à base d'un dictionnaire de 20k mots avec le critère d'optimisation " plus longue chaîne d'abord", nous obtenons un corpus de 15 millions de mots. Cinq autres corpus sont obtenus en effectuant les segmentations multiples avec le nombre de *n-meilleures* segmentations qui varie de 2 à 100. Les modèles de langage sont ensuite appris à partir de ces corpus en utilisant ce même dictionnaire de 20k mots.

Un corpus de développement (dev) de 370 phrases est utilisé pour évaluer la couverture en trigramme et la perplexité des modèles de langage du khmer. Ce corpus est segmenté automatiquement en mots et on peut en extraire 10975 trigrammes. Ces trigrammes sont utilisés pour évaluer le taux de couverture en trigrammes de nos modèles de langage. Les résultats d'expérimentation sont présentés dans le tableau 4.2. Nous présentons dans ce tableau le nombre de trigrammes dans les modèles de langage, la couverture en trigrammes de ces modèles, la perplexité et la performance du système de reconnaissance automatique de la parole en langue khmère (sur un corpus de test constitué de 160 phrases de type news et dont les transcriptions sont différentes de l'ensemble de dev) qui utilise ces modèles lors du décodage. Les détails sur le système de reconnaissance automatique en langue khmère (décodeur, modèle acoustique) sont donnés dans le chapitre 2.

Critère d'évaluation	Modèles issus des différentes segmentations					
	lm_{unique}	lm_2	lm_5	lm_{10}	lm_{50}	lm_{100}
Nombre de trigrammes dans le modèle (millions)	5.67	7.34	8.95	10.17	12.52	13.31
Trigramme hits sur Dev	31%	34.1%	34.6%	35.2%	36.6%	37%
Perplexité	394.9	322.5	348.8	361.8	373.9	374.7
CCER	22	21.7	20.8	**20.5**	20.6	20.7

TABLE 4.2 – *Résultats de la segmentation multiple sur la langue khmère.*

Pour voir le potentiel de l'approche de segmentation multiple dans la modélisation statistique du langage pour des langues peu dotées, nous effectuons aussi les expérimentations sur des corpus de taille plus réduite. La figure 4.3 montre

l'influence du nombre de segmentations sur la performance du système avec les différentes tailles de corpus. La discussion sur ces résultats est donnée dans la section 4.3.5.

LM	CCER Corpus 100%	CCER Corpus -50%	CCER Corpus -75%
lm_{unique}	22	22.8	23.9
lm_2	21.7	22.8	23.6
lm_5	20.8	22.6	23.5
lm_{10}	**20.5**	22.6	23.4
lm_{50}	20.6	22.4	23.5
lm_{100}	20.7	**22**	23.2
lm_{500}	20.9	22.6	**22.9**
lm_{1000}	21	22.8	23

TABLE 4.3 – *Impact du nombre de segmentations sur les différentes tailles de corpus en khmer.*

4.3.2 Application à la langue vietnamienne

Le système vietnamien utilisé dans cet expérimentation est développé dans le cadre de la thèse de V-B Le [Le, 2006]. Le corpus d'apprentissage du vietnamien contient 3 millions de phrases soit plus de 56 millions de syllabes. Après la segmentation unique automatique à base d'un dictionnaire de 30k mots avec le critère d'optimisation « longest matching », nous obtenons un corpus de 46 millions de mots. Les segmentations multiples sont également effectuées avec les nombres de Nseg variant de 2 à 100. Les modèles de langage trigrammes sont ensuite appris à partir de ces corpus en utilisant un dictionnaire de 30k mots (cf expérimentation sur le khmer).

Un corpus de développement (dev) de 100 phrases est utilisé pour évaluer la couverture en trigramme et la perplexité des modèles de langage. Ce corpus est segmenté automatiquement pour obtenir 44k mots. 33310 trigrammes sont générés à partir de ce corpus de développement. Les performances de reconnaissance de la parole sont aussi estimées sur un corpus de test de 400 phrases de type news et dont les transcriptions sont différentes de l'ensemble de dev). Les détails sur le système de reconnaissance automatique en langue vietnamienne sont donnés dans [Le *et al.*, 2008]. Les résultats des expérimentations sur le vietnamien sont dans le tableau 4.4.

Critère d'évaluation	Modèles issus des différentes segmentations					
	lm_{unique}	lm_2	lm_5	lm_{10}	lm_{50}	lm_{100}
Nombre de trigrammes dans le modèle (millions)	20.32	24.06	28.92	32.82	34.2	34.9
Trigramme hits sur Dev	47.7%	48.6%	49.2%	49.4%	49.7%	49.7%
Perplexité	118.9	118.1	125.9	129	133.4	134.8
CCER	27.6	**26.2**	27	26.5	26.7	26.9

TABLE 4.4 – *Résultats de la segmentation multiple sur la langue vietnamienne.*

La figure 4.5 montre l'influence du nombre de segmentations sur la performance du système avec les différentes tailles de corpus. La discussion sur ces résultats est donnée dans la section 4.3.5.

LM	SLER Corpus 100%	SLER Corpus -50%	SLER Corpus -80%	SLER Corpus -90%
lm_{unique}	27.6	28.9	33.7	35.2
lm_2	**26.2**	**27.9**	**32.4**	**34.3**
lm_5	27	28.2	33.6	34.5
lm_{10}	26.5	28.6	33.6	34.7
lm_{20}	26.7	28.6	33.7	34.6
lm_{50}	26.7	29.1	33.8	35.1
lm_{100}	26.9	29.1	34.3	34.9

TABLE 4.5 – *Impact du nombre de segmentations sur les différentes tailles de corpus en vietnamien.*

4.3.3 Application à la langue laotienne

Le laotien est la langue officielle du Laos, parlée par environ 5 millions d'habitants au Laos et en Thaïlande. C'est une langue tonale et d'une manière prédominante monosyllabique. La plupart des mots polysyllabiques dans le vocabulaire ont été empruntés, principalement au khmer, au thaï, au Pali ou au sanskrit. L'alphabet laotien est basé sur l'alphabet siamois ancien, comme pour l'alphabet thaï. Il est composé de 33 consonnes et de 28 voyelles, s'écrit de gauche à droite. Certaines voyelles sont disposées au-dessus ou au-dessous de la ligne des consonnes ; il n'y a ni capitales ni ponctuation spécifique. Comme son voisin khmer, le laotien est une langue non segmentée en mots. Selon [Berment, 2004], le laotien est une langue peu dotée.

Un nouveau système de reconnaissance automatique de la parole en langue laotienne est développé dans le cadre de notre expérimentation. Pour développer ce système, nous appliquons la même approche utilisée pour le développement du

système khmer décrite dans le chapitre 2.

En ce qui concerne la collecte des données textuelles, nous récupérons les textes à partir de 5 sites Internet qui proposent des contenus de type journaux en ligne : Radio France International [1], Radio Chine International [2], Radio Free Asia [3], le journal local Pasaxon [4] et Vientianemai [5]. Pendant 6 mois (janvier-juin 2009), nous obtenons 12k articles sous forme des documents html. Après le traitement et la segmentation automatique en mots, nous obtenons un corpus de texte de 60k phrases qui contient 2.8 millions de mots.

Pour construire notre corpus de parole laotienne, nous avons contacté l'édition laotienne de RFI pour demander les archives des émissions radio en laotien. La transcription manuelle des signaux a été organisée dans le cadre du stage d'un étudiant laotien au Laboratoire Mica, Hanoi, Vietnam. Environ 5h30mn de signaux ont été transcrits à ce jour. Ce corpus de parole contient 4000 phrases prononcées par 13 locuteurs. Après la segmentation manuelle de la transcription de ce corpus, nous obtenons un texte de 60k mots (2200 mots unique). Une grande partie de ce corpus est utilisé comme corpus de test (1h de signal, 800 phrases, 4 locuteurs).

L'approche de la modélisation acoustique à base de graphème est utilisé pour le système laotien. Chaque unité de modélisation est un caractère laotien. Le dictionnaire de prononciation à base de graphèmes est généré directement d'un vocabulaire de 27k mots. Une grande partie de ce vocabulaire (26k mots) est récupérée à partir du site Internet du projet SEAlang [6] qui donne l'accès à des ressources textuelles. La segmentation manuelle de la transcription de notre corpus de signal laotien donne une autre partie de vocabulaire, principalement les nouveaux mots de l'actualité et les noms propres.

Notre système laotien est basé sur le décodeur de Janus Ibis [Rogina et Waibel, 1995]. Un modèle acoustique laotien dépendant du contexte à base de graphèmes

1. Edition laotienne de RFI http ://www.rfi.fr/actulo/pages/001/accueil.asp
2. Edition laotienne de RCI : http ://laos.cri.cn
3. Edition laotienne de RFA : http ://www.rfa.org/lao/. RFA émet en 9 langues asiatiques.
4. http ://www.pasaxon.org.la
5. http ://www.vientianemai.net
6. Southeast Asian Languages, un projet financé par Center for Research in Computational Linguistics. Les ressources disponible en langues thaie, burmane, laotienne, khmère et vietnamienne.

est entraîné à partir des signaux d'apprentissage. Un modèle de langage trigramme est appris sur le corpus de texte segmenté automatiquement en mots avec la méthode de segmentation à base de dictionnaire "plus longue chaîne d'abord". Notre système laotien de base donne un taux d'erreur de mots de 43.7% sur le corpus de test. Nous utilisons dans le cadre des expérimentations sur la langue laotienne le métrique taux d'erreur de mots (WER) au lieu de taux d'erreur de groupe de caractères (CCER) pour mesurer la performance du système car nous ne possédons pas au moment du développement de ce système laotien, l'outil de segmentation d'un texte laotien en groupe de caractères.

Nous appliquons la méthode de segmentation multiple sur le corpus de texte laotien pour entraîner les différents modèles du langage. La table 4.6 donne les résultats de la segmentation multiple sur la langue laotienne. La discussion sur ces résultats est donnée dans la section 4.3.5.

Critère d'évaluation	Modèles issus des différentes segmentations				
	lm_{unique}	lm_2	lm_5	lm_{10}	lm_{50}
Nombre de trigrammes dans le modèle (millions)	0.38	0.42	0.44	0.48	0.50
Trigramme hits sur Dev	30.8%	35.2%	36.8%	37.2%	37.8%
Perplexité	183.7	157.3	161.1	165.7	168.2
WER	43.7%	42.9%	**42.4%**	42.7%	42.9%

TABLE 4.6 – *Résultats d'expérimentations sur la langue laotienne.*

4.3.4 Application à la langue thaïe

Les expérimentations sur le système thaï ont été effectuées pendant mon stage de recherche de deux mois au laboratoire Interactive Systems (interAct), de l'Université de Karlsruhe, Allemagne. Ces travaux ont été réalisés en collaboration avec Sebastian Stüker, chercheur au laboratoire Interactive Systems.

Le système thaï initial est développé au laboratoire Interactive Systems [Charoenpornsawat *et al.*, 2006]. Ce système est basé sur Janus Ibis. Le modèle acoustique à base de phonèmes est entraîné sur le corpus GlobalPhone [Suebvisai *et al.*, 2005] qui contient 20h de signal. Le corpus de texte est du domaine de journaux et contient 3,3 millions de mots (segmentation à base de dictionnaire "plus long chaîne d'abord"). Un vocabulaire de reconnaissance de 7400 mots est utilisé pour entraîner le modèle de langage qui donne une perplexité de 140 et 0% de mots hors

vocabulaire par rapport au corpus de test. Ce système a une taux d'erreur de mots de 8.2% sur un corpus de test de 640 phrases extraites de corpus GlobalPhone.

Dans le cadre de notre expérimentation, un corpus de développement de 600 phrases qui contient 8800 mots est utilisé pour évaluer la couverture en trigrammes et les perplexités des modèles du langage créés dans nos travaux. Les résultats des tests sur le thai sont rapportés dans le tableau 4.7. Nous utilisons dans le cadre de cet expérimentation le métrique taux d'erreur de mots (WER) au lieu de taux d'erreur de groupe de caractères (CCER) pour mesurer la performance du système thaï car nous ne possédons pas l'outil de segmentation d'un texte thai en groupe de caractères.

Critère d'évaluation	Modèles issus des différentes segmentations				
	lm_{unique}	lm_2	lm_5	lm_{10}	lm_{50}
Nombre de trigrammes dans le modèle (millions)	0.34	0.39	0.43	0.46	0.48
Trigramme hits sur Dev	33.9%	38.2%	38.7%	39%	39.1%
Perplexité	110.7	100.4	101.1	101.7	102.2
WER	8.2%	**7.5%**	8.4%	8.6%	8.7%

TABLE 4.7 – *Résultats d'expérimentations sur la langue thaïe.*

4.3.5 Discussion

Les résultats des expérimentations sur ces trois langues montrent que la segmentation multiple permet de générer des nouveaux trigrammes quand le nombre de *n-meilleures* segmentations est augmenté. Nous pouvons observer que l'augmentation du nombre de tri-grammes dans le modèle de langage améliore généralement la couverture en trigramme sur le corpus de développement. Cette amélioration montre que les nouveaux trigrammes générés par cette approche sont potentiellement pertinents pour la modélisation deulangage. Les gains de performances sont également observées dans le système de reconnaissance khmer, laotien et thaï.

L'expérimentation en langue khmère sur les différentes tailles de corpus montre que quand nous diminuons la taille des corpus, l'augmentation du nombre de *n-meilleures* segmentations aide à améliorer la performance du système. Ceci montre le potentiel de l'approche pour la modélisation statistique des langues peu dotées pour les quelles peu de données textuelles sont disponibles.

Dans le cas de la langue khmère, la meilleure performance du système de reconnaissance est obtenue avec le modèle de langage lm_{10} et la performance diminue si nous continuons à augmenter le nombre de *n-meilleures* segmentations. Ce phénomène peut être expliqué par le fait qu'à partir d'un certain point, quand nous continuons à augmenter le nombre de *n-meilleures* segmentations, ces segmentations ne génèrent plus des nouveaux trigrammes qui sont pertinents pour la langue. Au contraire, ces trigrammes perturbent le calcul des probabilités dans le modèle du langage. En regardant le taux de couverture en trigramme dans nos expérimentations, nous pouvons constater que la couverture en trigramme commence à être saturée à partir d'un certain nombre de *n-meilleures* segmentations même si nous continuons à ajouter les nouveaux trigrammes dans le modèle du langage. Cela suggère que les trigrammes générés par la segmentation multiple doivent être filtrés pour éliminer les mauvaises trigrammes avant de les utiliser dans la modélisation du langage. Notre approche qui est fondée sur un modèle de langage unigramme pendant la segmentation multiple pour trier globalement les segmentations ne semble pas suffisante pour filtrer ces mauvaises trigrammes.

La segmentation multiple vise à générer un maximum de segmentations possibles à partir d'un corpus pour obtenir le plus de trigrammes. Cependant, il faut un mécanisme pour filtrer ces trigrammes en ne gardant que ceux qui sont pertinents pour la langue. Plusieurs pistes peuvent être explorées pour essayer d'améliorer la segmentation multiple. Une première approche simple peut être basée sur les fréquences de tous les trigrammes générés. Nous pourrons définir un seuil qui décide si un trigramme est pertinent se fondant sur sa fréquence relative. Dans la littérature, le mécanisme d'élagage (*pruning*) est généralement utilisé pour réduire la taille et la complexité du modèle de langage. Ces mécanismes sont basés sur les critères comme la probabilité, le rang du mot et l'entropie.

4.4 Conclusion

Dans ce chapitre, nous proposons une méthode pour effectuer la segmentation multiple sur le corpus d'apprentissage pour estimer des modèles de langage *n*-gramme dans le contexte des langues peu dotées et non segmentées en mot. La

méthode de segmentation multiple basée sur des automates d'état finis permet de générer plusieurs segmentations à partir de corpus de texte pour en extraire les n-grammes. A travers des expérimentations sur la langue khmère, laotienne et thaïe, nous avons montré que notre approche permet de générer plus de trigrammes que la segmentation unique classique et ces trigrammes sont utiles et pertinents pour la modélisation du langage. Des expérimentations sur la langue khmère, laotienne et thaïe montrent l'amélioration en terme de la couverture en trigramme et de la performance des systèmes de reconnaissance. Une future amélioration de la segmentation multiple consiste à filtrer les trigrammes générés en ne gardant que ceux qui sont pertinents pour la modélisation.

Conclusion

Ce travail de thèse porte sur la reconnaissance automatique de la parole des langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots. Les langues traitées dans notre contexte d'étude sont un groupe de langues peu dotées de l'Asie du sud-est : le khmer, le laotien, le thaï et le vietnamien. Le problème de la faible quantité de données textuelles et les erreurs introduites par la segmentation automatique impliquent de réfléchir à des techniques de modélisation lexicale et sous-lexicale permettant ainsi de réduire la taille du vocabulaire de l'application, tout en essayant d'exploiter au mieux les données. Nous avons proposé de traiter ce problème en exploitant plusieurs vues sur les données textuelles dans la modélisation du langage. Nos recherches sont axées principalement sur la modélisation du langage, et en particulier sur l'utilisation de multiples unités dans le système de reconnaissance automatique de la parole.

La première partie de cette thèse est consacrée au développement d'un système de reconnaissance automatique de la parole pour la langue khmère qui a servi comme système de référence dans nos expérimentations. Nous avons d'abord collecté les ressources linguistiques nécessaires : le corpus de parole, le corpus de texte, le dictionnaire de prononciation et développé les outils de base pour traiter ces données. Une boîte à outils a été développée pour la création de corpus de texte khmer à partir de documents html collectés sur le web. Cette boîte à outils contient les outils pour normaliser l'orthographe du texte khmer et pour faire la segmentation en différentes unités lexicales (mot, syllabe, cluster de caractères). Ensuite, nous avons développé un système de reconnaissance automatique de la

parole de l'état de l'art pour la langue khmère (broadcast news) à partir de ces ressources. Deux approches ont été testées dans la modélisation acoustique pour le khmer. Nous avons appris un modèle acoustique à base de phonèmes et un modèle à base de graphèmes. Le dictionnaire de prononciation à base de phonèmes est généré à l'aide de notre outil de conversion graphème-phonème à base de règles, tandis que le dictionnaire de prononciation à base de graphèmes est obtenu directement en utilisant chaque graphème khmer comme unité de modélisation. Pour la modélisation du langage, nous avons expérimenté plusieurs unités lexicales et sous-lexicales. En plus de l'unité lexicale classique, le "mot", nous avons utilisé deux unités sous-lexicales pour modéliser le khmer, la syllabe et le cluster de caractères. Mais les résultats de test de notre système khmer ont montré que le mot reste l'unité la plus performante malgré les erreurs de segmentation en mots. L'avantage de l'unité plus petite comme le cluster de caractères réside dans le fait que la segmentation est triviale, il n'y a pas de problème de mots hors vocabulaire et la taille de vocabulaire est plus petite par rapport à une unité comme le mot. Ce qui réduit significativement la complexité du système. Les résultats ont aussi montré le potentiel de la modélisation acoustique à base de graphème pour une langue peu dotée comme le khmer quand un dictionnaire de prononciation à base de phonèmes n'est pas disponible.

Avec les mêmes méthodes et outils utilisés pour développer le système khmer, nous avons développé un système de reconnaissance automatique de la parole (broadcast news) pour une autre langue peu dotée, le laotien, langue officielle du Laos, voisin du Cambodge. A notre connaissance, notre système est le premier travail sur la reconnaissance automatique en langue laotienne. Les travaux pour améliorer notre système laotien sont en cours, notamment en exploitant les ressources disponibles en langue thaïe, car il existe de nombreuses similarités entre ces deux langues.

Dans le but de mieux exploiter les données textuelles, nous avons proposé d'exploiter plusieurs unités lexicales et sous-lexicales pour la reconnaissance automatique de la parole des langues peu dotées et non-segmentées. Nous argumentons que l'utilisation de multiples unités au lieu d'une seule unité permet d'exploiter plusieurs vues sur la même donnée et peut permettre de compenser les erreurs

introduites par la segmentation automatique.

Au niveau de la modélisation du langage, nous avons proposé des modèles créés avec des vocabulaires hybrides qui combinent à la fois l'unité lexicale et sous-lexicale et à partir de corpus segmentés en fonction de ces vocabulaires hybrides. L'application de modèles hybrides sur la langue khmère montre que les modèles hybrides créés en combinant les mots et les clusters de caractères donnent une meilleures performance que le modèle qui est fondé sur une seule unité "mot" ou "cluster de caractères". Cependant, les expérimentations sur la langue vietnamienne qui combinent le mot et la syllabe pour créer les modèles hybrides n'ont pas montré la même tendance. Le syllabe reste l'unité la plus performance pour notre système vietnamien, probablement du fait qu'il y a majoritairement des mots monosyllabiques et bisyllabiques en vietnamien et le texte est naturellement segmenté en syllabe.

Au niveau de la fusion de systèmes, nous avons proposé de combiner des sorties de systèmes fondés sur différentes unités dans le but de décoder une meilleure hypothèse. Deux méthodes de combinaisons de systèmes ont été expérimentées. La première méthode de combinaison de type ROVER a été testée pour fusionner la liste des N-meilleures hypothèses de chaque système. L'application de ROVER sur le vietnamien pour combiner les sorties de systèmes fondés sur le mot et la syllabe donne une gain significatif de performance par rapport au systèmes de référence à base de mot ou de syllabe. Dans le cas du khmer, cette méthode n'améliore pas la performance du système quand nous combinons le système à base de mot avec celui à base de cluster de caractères. Une raison de cet échec est probablement dûe aux performances trop faibles du système à base de cluster de caractères par rapport au système à base de mot, ce qui rend la fusion inefficace. Une second méthode qui travaille au niveau des treillis au lieu de la liste des N-meilleures hypothèses a été également employée. Notre méthode de combinaison de treillis propose une façon de décomposer les noeuds de treillis en prenant en compte correctement les informations temporelles et les scores. Cette méthode donne un gain de performance dans les expérimentations sur le système vietnamien. Dans le cas de la langue khmère, l'amélioration de performance est obtenue en effectuant une étape supplémentaire qui consiste à rescorer les treillis combinés avec un

modèle de langage d'ordre plus élevé.

Dans la même optique consistant à mieux exploiter les données textuelles en utilisant multiples vues sur les mêmes données textuelles, nous avons proposé une méthode qui effectue des segmentations multiples sur le corpus d'apprentissage au lieu d'une segmentation unique classique. Cette méthode de segmentation multiple basée sur des automates d'état finis permet de générer les n-meilleures segmentations à partir d'une séquence de caractères et nous pouvons ensuite en extraire les n-grammes pour apprendre le modèle de langage. Elle permet de retrouver les n-grammes non trouvés par la segmentation unique à cause des erreurs de segmentation et de générer de nouveaux n-grammes pour la modélisation statistique de langage. Nous avons appliqué cette approche dans l'apprentissage des modèles de langage pour les systèmes de reconnaissance automatique de la parole en langue khmère, laotienne, thaïe et vietnamienne. Les résultats des expérimentations ont montré que des nouveaux trigrammes ont été générés quand le nombre de *n-meilleures* segmentations a été augmenté. Nous avons pu également observer que l'augmentation du nombre de tri-grammes dans le modèle de langage améliore généralement la couverture en trigramme sur le corpus de développement. Cette amélioration montre que les nouveaux trigrammes générés par cette approche sont potentiellement pertinents pour la modélisation du langage. Les gains de performances sont également observées dans le système de reconnaissance khmer, laotien, thaï et vietnamien. Les expérimentations sur des corpus de taille réduite dans le cas du khmer et du vietnamien ont montré le potentiel de l'approche de segmentation multipe pour la modélisation statistique des langues peu dotées. Notre méthode d'extraction des trigrammes à partir des segmentions multiples est perfectible car elle génère des trigrammes qui perturbent l'estimation de modèle de langage.

Dans la continuité de notre travail, plusieurs travaux sont envisagés. Premièrement, dans le but de mieux doter les deux langues principalement développés dans notre contexte d'études à savoir le khmer et le laotien, la collecte de plus de données est prévue. Une campagne de collecte de données sera effectuée dans le cadre du projet PI [7] via le partenaire du projet, le centre de MICA à Hanoi en collabo-

7. Le projet PI (ANR BLANC 2009-2012) concerne le traitement automatique du langage parlé (notamment la reconnaissance automatique de la parole) pour les langues peu dotées. Les

ration avec l'Institut de Technologie du Cambodge. A terme, notre objectif est de collecter au moins 20h de signal de parole finement annotée pour chaque langue. Des données textuelles seront également collectées de façon continue sur Internet pour augmenter la taille de notre corpus de texte. Ces données seront utilisées pour améliorer la robustesse de nos systèmes et pour servir dans d'autres travaux de recherche sur le khmer et le laotien. Deuxièmement, nous envisageons d'améliorer notre approche par la segmentation multiple qui a montré un potentiel pour les langues peu dotées et non segmentées. Les résultats de nos expérimentations suggèrent que les trigrammes générés par la segmentation multiple doivent être filtrés pour éliminer les mauvais trigrammes avant de les utiliser dans la modélisation du langage. Notre approche qui est fondée sur un modèle de langage unigramme pendant la segmentation multiple pour trier globalement les segmentations ne semble pas suffisante pour filtrer ces mauvais trigrammes. Il faudra travailler aussi au niveau de trigramme en utilisant quelque règles linguistiques simple pour éliminer les séquences impossibles. Pour généraliser l'approche de segmentation multiple, Il est pertinent de valider de cette méthode sur les autres langues peu dotées ayant des caractéristiques différentes de groupe de langues dans notre études, par exemple sur les langues morphologiquement riches qui nécessitent la décomposition en morphèmes au lieu de la segmentation en mots comme les langues non segmentées traitées dans nos études. Il serait également intéressant d'appliquer cette approche sur un autre domaine qui possède une composante linguistique comme la traduction automatique. D'un point de vu technique, comme les systèmes de reconnaissance développés dans le cadre de cette thèse s'appuient essentiellement sur des outils libres, et qu'un certain nombre d'outils de traitement de base ont été développés, il serait nécessaire de rassembler et de bien documenter ces outils pratiques afin que d'autres développeurs puissent les réutiliser et ainsi d'accélérer le développement d'un système de reconnaissance automatique pour une nouvelle langue.

partenaires sont le LIG, le LIA, et le centre international MICA (Hanoï, Vietnam).

Annexe 1 : Implémentation de la segmentation multiple par automates d'états finis

Dans cette annexe, nous décrivons l'implémentation de notre algorithme de segmentation multiple décrit dans la section 4.2.3. Cet algorithme peut être décomposé en plusieurs étapes simples en utilisant les automates d'états finis. L'implémentation de cet algorithme de segmentation multiple à base des automates d'états finis est effectuée en utilisant la boîte à outils AT&T FSM (Finite-State Machine Library) et la librairie GRM (Grammar Library) [Mohri *et al.*, 1998].

Algorithmique

A partir d'une chaîne de caractère en entrée I et d'un dictionnaire, nous souhaitons générer les n-meilleures segmentations de I. Les n-meilleures segmentations sont évaluées selon un modèle de langage L appris sur un corpus segmenté en mot par les outils de segmentation automatique à base de dictionnaire.

Pour ce faire, nous devons tout d'avoir créer un transducteur de mots T à partir du dictionnaire. La décomposition de la chaîne de caractères I avec le transducteur de mots T donne un transducteur qui représente toutes les segmentations en mots possible de I. Ce transducteur est ensuite composé avec le modèle de langage L

pour obtenir un treillis scoré. Les *n-meilleures* segmentations $\hat{m}$ sont obtenues par le décodage du treillis final pour sortir les *n-meilleurs* chemins comme décrit dans l'équation 4.4.

$$\hat{m} = nbestpath(I \circ T \circ L) \tag{4.4}$$

où $\circ$ est l'opération de composition et *nbestpath* est la fonction qui décode les meilleurs chemins.

Les données en entrée

Pour construire notre outil de segmentation multiple, nous avons besoins d'abord d'un dictionnaire de mots. Ensuite pour entraîner le modèle de langage utiliser pour générer les n-meilleures segmentations, nous avons besoin un corpus de texte segmenté en mots. A partir de ces deux éléments nous générons une liste des symboles (les caractères) pour les automates. Plusieurs caractères spéciaux doivent être ajoutés dans la liste des symboles : *space* représente un espace blanc, *epsilon* est un symbole réservé pour la composition des automates, et *start* et *end* sont deux symboles utilisés pour construire le modèle de langage. Pour apprendre le modèle de langage, le corpus d'apprentissage doit être sous un format spécial. Chaque phrase doit commencer par *start* et finir par *end*. Les espaces entre les mots doivent être représentés explicitement par *space*.

Transducteur de mots

L'objectif est de construire un transducteur qui peut convertir une chaîne en entrée en séquences des mots dans le dictionnaire en essayant toutes les possibilités de segmentations possibles. Ce transducteur est construit à partir d'un dictionnaire. Pour une segmentation qui n'utilise que les mots dans le dictionnaire, le score est minimale. Pour celle qui a besoin d'utiliser le modèle de mot inconnu, un score élevé est attribué comme pénalité.

Pour développer ce transducteur, nous utilisons le script perl dans le listing 4.1. Puis nous utilisons les outils de FMS pour créer un transducteur par l'opération de détermination.

```
$ ./makedictransducer.pl  vocabulaire.txt   \
| fsmcompile -t -i symbols | fsmdeterminize >  dictionary.fsm
```

Listing 4.1 – Script pour préparer le donnée de transducteur

```
#!/usr/bin/perl
#maketransducer.pl
#
my $st = 1;
while (<>) {
chop ;
/ \^ (\S+) \s +(\d+) $ / ;
$word = $1;
$id = $2 ;
# skipwords epsilon, begin , end , unknown , space
if ($id < 5 ) {
next ;
}
print ''0 $st epsilon 0 \ n'' ;
for  each $ch (split // , $word) {
print $st++, '' '' , $st , '' '' , $ch , '' 0 \n '' ;
}
print $st++, '' '', $st , ''epsilon $id \n'' ;
print $st++, ''\n''
}
```

Comme pour tous les systèmes basés sur un vocabulaire, nous devons traiter les mots hors vocabulaire. Nous utilisons un modèle des mots inconnus sous forme

d'un automate qui analyse n'importe quelle séquence de caractères et génère un symbole unique *inconnu* pour représenter les mots inconnus. Quand nous annotons une séquence de caractères comme *inconnu*, nous voulons pénaliser celle ci avec un coût (50.0).

```
$ cat  unknowntag
0  1  0  unknown  50.0
1
$ fsmcompile -t -o segments unknowntag > unknowntag.fsm
```

unknowntag.fsm est utilisé pour changer le score des mots inconnus. Pour construire un modèle pour les mots inconnus, nous avons besoin d'un automate qui accepte un nombre illimité de caractères. Une opération de fermeture sur l'ensemble des caractères (symboles) donne un transducteur qui accepte un ou plusieurs caractères. La concaténation avec le modèle *unknowntag.fsm* donne un modèle qui peut sortir un mot inconnu *inconnu*.

```
$ grmcount -n1 -s1 -f2 trainingcorpus.far  \
|  grmmake > unigram.fsm

$ fsmprint unigram.fsm | wc -l
64
$ fsmprint unigram.fsm | tail -61 |  \
perl -ne 'END { print "1"; }
if (/(0\s+0\s+)(\d+)(.*)/) {
print "0\t1\t$2\t0$3\n"; }'  \
|  fsmcompile -t > unigramfilt.fsm

$ fsmclosure -p unigramfilt.fsm  \
| fsmconcat - unknowntag.fsm  \
| fsmrmepsilon > unknownseg.fsm
```

Finalement, une opération d'union entre le dictionnaire et le modèle de mot inconnu suivi par une opération de fermeture de *Kleene* "+" permet d'obtenir un modèle qui est capable de produire toutes les segmentations possibles à partir d'une séquence de caractères donnée en entrée.

```
$ fsmunion unknownseg.fsm dictionary.fsm \
| fsmclosure -p >  wordtransducer.fsm
```

Modèle de langage

Pour générer les n-meilleures segmentations, nous avons besoin d'un modèle de langage pour combiner avec le transducteur de mot pour favoriser les segmentations qui sont correctes.

Un modèle de langage peut être appris sur les données apprentissage avec l'outil GRM. Il est à noter que dans notre cas, un modèle de langage unigramme est utilisé (voir l'explication dans la section 4.2.3).

```
$ cat train.txt \
| farcompilestrings -i segments -u unknown > trainsegments.far
$ grmcount -n1 -s1 -f2 trainsegments.far \
| grmmake | grmshrink > seglm.fsm
```

Génération de n-meilleures segmentations

La chaîne de caractères en entrée est d'abord convertie en un accepteur. Les n-meilleures segmentations sont finalement obtenues en décomposant l'accepteur avec le transducteur de mot et le modèle de langage. Nous pouvons voir dans les résultats de segmentations les meilleures segmentations qui ont les scores égales à 0 et les segmentations qui doit utiliser les mots inconnus *inconnu* on les scores plus élevés.

```
$echo "ព្រះ ពុ ទ្ធ ជា ព្រះ ប រ មេ គ្រូ នៃ យើ ង" | farcompilestrings -i symbols  > input.fsm

$ fsmcompose input.fsm segmenter.fsm seglm.fsm  > lattice.fsm \

$ cat  lattice.fsm | farprintstrings -u -c -n 15 -o segments

ព្រះពុទ្ធ ជា ព្រះបរមេ គ្រូ នៃ យើង      0
ព្រះ ពុទ្ធ ជា ព្រះបរមេគ្រូ នៃ យើង      0
ព្រះពុទ្ធ ជា ព្រះ បរមេគ្រូ នៃ យើង      0
ព្រះ ពុទ្ធ ជា ព្រះ បរមេគ្រូ នៃ យើង      0
ព្រះ ពុទ្ធ ជា ព្រះបរមេ គ្រូ នៃ យើង      0
ព្រះ ពុទ្ធ ជា ព្រះ បរមេ គ្រូ នៃ យើង      0
ព្រះពុទ្ធ ជា ព្រះបរមេគ្រូ នៃ យើង      0
ព្រះពុទ្ធ ជា ព្រះ បរមេ គ្រូ នៃ យើង      0
ព្រះពុទ្ធ ជា ព្រះ បរមេគ្រូ នៃ យើ inconnu       53.5149307
ព្រះពុទ្ធ ជា ព្រះ បរមេ គ្រូ នៃ យើ inconnu       53.5149307
ព្រះ ពុទ្ធ ជា ព្រះ បរមេគ្រូ នៃ យើ inconnu       53.5149307
ព្រះ ពុទ្ធ ជា ព្រះបរមេគ្រូ នៃ យើ inconnu       53.5149307
ព្រះ ពុទ្ធ ជា ព្រះបរមេ គ្រូ នៃ យើ inconnu       53.5149307
ព្រះពុទ្ធ ជា ព្រះ inconnu រមេ គ្រូ នៃ យើង      54.0766296
ព្រះ ពុទ្ធ ជា ព្រះ inconnu រមេ គ្រូ នៃ យើង      54.0766296
ព្រះ ពុទ្ធ ជា ព្រះ បរ inconnu គ្រូ នៃ យើង      54.1804237
```

Documentation sur la boîte à outils AT&T FSM

La boîte à outils AT&T FSM fournit les outils pour construire, combiner, minimiser et parcourir les automates d'états finis. Plus de détails sur FSM se trouve sur www.research.att.com/∼fsmtools/fsm. Un documentation en ligne des outils de FSM est disponible sur : www.research.att.com/∼fsmtools/fsm/tech.html. La librairie pour la définition de modèles de langues et de grammaires d'AT&T GRM est sur www.research.att.com/∼fsmtools/grm.

Annexe 2 : Liste des publications personnelles

Conférences Internationales

[Seng et al., 2008a] Seng, S., Sam, S., Besacier, L., Bigi, B., et Castelli, E. (2008a). *First broadcast news transcription system for khmer language*. In The 6th edition of the Language Resources et Evaluation Conference (LREC 2008), page 4p, Marrakech (Morocco).

[Le et al., 2008a] Le, V., Seng, S., Besacier, L., et Bigi, B. (2008). *Word/sub-word lattices decomposition and combination for speech recognition*. In ICASSP, pages 4321–4324, Las Vegas, NV, USA.

[Seng et al., 2008d] Seng, S., Sam, S., Le, V.-B., Bigi, B., et Besacier, L. (2008d). *Which units for acoustic and language modeling for khmer automatic speech recognition*. In Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU 2008), Hanoi University of Technology, Hanoi (Vietnam).

[Le et al., 2008b] Le, V.-B., Besacier, L., Seng, S., Bigi, B., et Do, T. N. D. (2008). *Recent advances in automatic speech recognition for vietnamese*. In Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU 2008), Hanoi University of Technology, Hanoi (Vietnam).

[Seng et al., 2009a] Seng, S., Besacier, L., Bigi, B., et Castelli, E. (2009a). *Multiple text segmentation for statistical language modeling*. In Interspeech,

Brighton.

Conférences francophones

[Seng et al., 2008c] Seng, S., Sam, S., Le, V.-B., Bigi, B., et Besacier, L. (2008c). *Reconnaissance automa- tique de la parole en langue khmère : quelles unités pour la modélisation du langage et la modélisation acoustique ?* In JEP/TALN 2008, 27èmes Journées d'Etudes sur la parole, AFCP-ATALA , LIA, Université d'Avignon, Avignon.

[Seng et al., 2009b] Seng, S., Bigi, B., Besacier, L., et Castelli, E. (2009b). *Segmentation multiple d'un flux de données texteulles pour la modélisation statistique du langage* In TALN 2009, Senlis, 24-26juin 2009. TALN2009.

Bibliographie

[Abdillahi *et al.*, 2006] Abdillahi, N., Nocera, P. et Bonastre, J.-F. (2006). Automatic transcription of somali language. *In Interspeech*, Pittsburgh PA, USA.

[Abdou, 2004] Abdou, S. (2004). The 2004 bbn levantine arabic and mandarin cts transcription systems. *In DARPA RT-04 Workshop*, New York.

[Afify *et al.*, 2006] Afify, M., Sarikaya, R., Kuo, J., Besacier, L. et Gao, Y. (2006). On the use of morphological analysis for dialectal arabic speech recognition. *In 9th International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP)*, Pittsburgh, Pennsylvania (US).

[Amorrortu *et al.*, 2004] Amorrortu, E., Barrena, A., Idiazabal, I., Izagirre, E., Ortega, P. et Uranga, B. (2004). World languages review synthesis. *In Unesco Etxea*.

[Arisoy *et al.*, 2006] Arisoy, E., Dutağaci, H. et Arslan, L. M. (2006). A unified language model for large vocabulary continuous speech recognition of turkish. *Signal Process.*, 86(10):2844–2862.

[Barras *et al.*, 2001] Barras, C., Geoffrois, E., Wu, Z. et Liberman, M. (2001). Transcriber : Development and use of a tool for assisting speech corpora production. *Speech Commun.*, 33(1-2):5–22.

[Barrault, 2008] Barrault, L. (2008). *Diagnostic pour la combinaison de systèmes de reconnaissance automatique de la parole*. Thèse de doctorat, l'Université d'Avignon et des Pays de Vaucluse, France.

[Baum *et al.*, 1970] Baum, L. E., Petrie, T., Soules, G. et Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *In Ann. Math. Statist.*, volume 41, pages 164–171.

[Berment, 2004] Berment, V. (2004). *Méthodes pour informatiser les langues et les groupes de langues peu dotées*. Thèse de doctorat, UJF.

[Billa et al, 2002] Billa, J. et al (2002). Audio indexing of arabic broadcast news. *In Proceedings of the IEEE International Conference on Acoustique, Speech and Signal Processing*, pages 5–8, Orlando, FL.

[Bisani et Ney, 2003] Bisani, M. et Ney, H. (2003). Multigram-based grapheme-to-phoneme conversion for lvcsr. *In Proceedings of the EUROSPEECH.*, pages 933–936, Geneva, Switzerland.

[Charoenpornsawat *et al.*, 2006] Charoenpornsawat, P., Hewavitharana, S. et Schultz, T. (2006). Thai grapheme-based speech recognition. *In NAACL '06 : Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume : Short Papers on XX*, pages 17–20, Morristown, NJ, USA. Association for Computational Linguistics.

[Chen *et al.*, 2000] Chen, L., Lamel, L., Adda, G. et luc Gauvain, J. (2000). Broadcast news transcription in mandarin. *In Proc. ICSLP'2000*.

[Creutz et Lagus, 2006] Creutz, M. et Lagus, K. (2006). Morfessor in the morpho challenge. *In Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*.

[Crystal, 2000] Crystal, D. (2000). Language death. *In Cambridge University Press*.

[Denoual et Lepage, 2006] Denoual, E. et Lepage, Y. (2006). The character as an appropriate unit of processing for non-segmenting languages. *In Proceedings of the Annual Meeting of the Association for Natural Language Processing*, volume 12, pages 731–734, Japan.

[Do *et al.*, 2009] Do, T. N. D., Le, V.-B., Bigi, B., Besacier, L. et Castelli, E. (2009). Mining a comparable text corpus for a vietnamese-french statistical machine translation system. *In Fourth Workshop on Statistical Machine Translation, March 2009*. EACL 2009.

[Emerson, 2005] Emerson, T. (2005). The second international chinese word segmentation bakeoff. *In 4th SIGHAN Workshop on Chinese Language Processing*, Jeju.

[Fiscus, 1997] Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates : Recognizer output voting error reduction (rover). *In Proc. ASRU*.

[Grimes, 2000] Grimes, B. (1996-2000). Ethnologue : Languages of the world. *In Summer Institute of Linguistics*, Dallas.

[Hagège, 2002] Hagège, C. (2002). *Halte à la mort des langues*. Odile Jacob.

[Haruechaiyasak et Kongyoung, 2008] Haruechaiyasak, C. et Kongyoung, S. (2008). A comparative study on thai word segmentation approaches. *In Proceedings of ECTI-CON*.

[Hermansky et Cox, 1991] Hermansky, H. et Cox, J. (1991). Perceptual linear predictive (plp) analysis resynthesis technique. *In* IEEE, éditeur : *ASSP Workshop on Applications of Signal Processing to Audio and Acoustics Final Program and Paper Summaries*, pages 037–038.

[Hillard *et al.*, 2007] Hillard, D., Hoffmeister, B., Ostendorf, M., Schlüter, R. et Ney, H. (2007). irover : Improving system combination with classification. *In HLTNAACL*, pages 65–68.

[Huffman, 1970] Huffman, F. (1970). *Cambodian System of Writing and Beginning Reader*. Yale University Press.

[Jelinek, 1970] Jelinek, F. (1970). Continous speech recognition by statistical methods. *In* FDF, éditeur : *IEEE*, volume 64 :4, pages 532–556.

[JL. Gauvain et Adda, 2002] JL. Gauvain, L. L. et Adda, G. (2002). The limsi broadcast news transcription system. *In Speech Communication,*, volume 37(1-2), pages 89–108.

[Kanthak et Ney, 2002] Kanthak, S. et Ney, H. (2002). Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. *In Proceedings of ICASSP*, volume 1, pages 845–848, Orlando.

[Killer *et al.*, 2003] Killer, M., Stücker, S. et Schultz, T. (2003). Grapheme based speech recognition. *In Proceedings of Eurospeech*, Genf.

[Kneser et Ney, 1995] Kneser, R. et Ney, H. (1995). Improved backing-off for m-gram language modeling. *In ICASSP*, volume 1, pages 81 – 184. ICASSP.

[Kumar *et al.*, 2007] Kumar, A., Rajput, N., Chakraborty, D., Agarwal, S. K. et Nanavati, A. A. (2007). Wwtw : the world wide telecom web. *In NSDR '07 : Proceedings of the 2007 workshop on Networked systems for developing regions*, pages 1–6, New York, NY, USA. ACM.

[Kurimo *et al.*, 2006] Kurimo, M., Creutz, M., Varjokallio, M., Arisoy, E. et Saraclar, M. (2006). Unsupervised segmentation of words into morphemes – morpho challenge 2005 : Application to automatic speech recognition. *In Proceedings of ICSLP*, Pittsburg.

[Le *et al.*, 2008] Le, V., Seng, S., Besacier, L. et Bigi, B. (2008). Word/sub-word lattices decomposition and combination for speech recognition. *In ICASSP*, pages 4321–4324, Las Vegas, NV, USA.

[Le, 2006] Le, V.-B. (2006). *Reconnaissance Automatique de la parole des langues peu dotées*. Thèse de doctorat, UJF.

[Le *et al.*, 2003] Le, V.-B., Bigi, B., Besacier, L. et Castelli, E. (2003). Using the web for fast language model construction in minority languages. *In 8th European Conference on Speech Communication and Technology (Eurospeech'03)*, pages 3117–3120,, Geneva, Switzerland.

[Lecouteux, 2008] Lecouteux, B. (2008). *Reconnaissance automatique de la parole guidée par des transcriptions a priori*. Thèse de doctorat, Université d'Avignon et des Pays de Vaucluse, France.

[Lee *et al.*, 2003] Lee, Y.-S., Papineni, K., Roukos, S., Emam, O. et Hassan, H. (2003). Language model based arabic word segmentation. *In ACL '03 : Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 399–406, Morristown, NJ, USA. Association for Computational Linguistics.

[Li, 2005] Li, X. (2005). *Combination and Generation of Parallel Feature Streams for Improved Speech Recognition*. Thèse de doctorat, Carnegie Mellon University, Pittsburgh, PA.

[Luo *et al.*, 2009] Luo, J., Lamel, L. et Gauvain, J.-L. (2009). Modeling characters versus words for mandarin speech recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 0:4325–4328.

[Mangu *et al.*, 2000] Mangu, L., Brill, E. et Stolcke, A. (2000). Finding consensus in speech recognition : word error minization and other application of confusion network. *In CSL*, volume 14, pages 373–400.

[Markel et JR., 1976] Markel, J. D. et JR., A. H. G. (1976). Linear prediction of speech. *In ommunication and Cybernetics*, Berlin Heidelberg New York. Springer-Verlag.

[Meknavin *et al.*, 1997] Meknavin, S., Charoenpornsawat, P. et Kijsirikul, B. (1997). Feature-based thai word segmentation. *In NLPRS'97*, Phuket, Thailand.

[Meng *et al.*, 2008] Meng, S., Yu, P., Liu, J. et Seide, F. (2008). Fusing multiple systems into a compact lattice index for chinese spoken term detection. *In ICASSP*, pages 4345–4348, Las Vegas, NV,.

[Mohri *et al.*, 1998] Mohri, M., Pereira, O. et Riley, M. (1998). A rational design for a weighted finite-state transducer library. *In Lecture Notes in Computer Science*, pages 144–158. Springer.

[Nimaan, 2007] Nimaan, A. (2007). *Sauvegarde du patrimoine oral africain : conception de syst'eme de transcription automatique de langues peu dotées pour l'indexation des archives audio.* Thèse de doctorat, Université d'Avignon et des pays du Vaucluse, France.

[Pelligrini, 2008] Pelligrini, T. (2008). *Transcription automatique de langues peu dotées.* Thèse de doctorat, Université Paris-Sud, Paris, France.

[Puthick, 2005] Puthick, H. (2005). Development of a khmer spell checker based on a hidden markov model. Master degree report, The Department of Computer Science, Australian National University.

[R. Iyer et Meteer., 1997] R. Iyer, M. O. et Meteer., M. (1997). Analyzing and predicting language model improvements. *In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding.*

[Roach *et al.*, 1996] Roach, P., Arnfield, S., Barry, W., Baltova, J., Boldea, M., Marasek, Marchal, A., Meister, E. et Vicsi., K. (1996). Babel :

An eastern european multi-language database. *In Proceedings of ICSLP*, volume 3, pages 1892–1893, Philadelphia.

[Rogina et Waibel, 1995] ROGINA, I. et WAIBEL, A. (1995). The janus speech recognizer. *In In ARPA SLT Workshop*, pages 166–169. Morgan Kaufmann.

[S.C. Martin et Ney, 1997] S.C. MARTIN, J. L. et NEY, H. (1997). Adaptive topic dependent language modelling using wordbased varigrams. *In Proceedings of Eurospeech*, volume 3, pages 1447–1450, Rhodes.

[Schmid, 1994] SCHMID, H. (1994). Probabilistic part-of-speech tagging using decision trees. *In Conerence on New Methods in Language Processing*, Manchester.

[Schultz, 2002] SCHULTZ, T. (Septembre 2002). Globalphone : A multilingual speech and text database developed at karlsruhe university. *In ICSLP'02*, Denver, CO, USA.

[Schultz *et al.*, 2007] SCHULTZ, T., BLACK, A., BADASKAR, S., HORNYAK, M. et KOMINEK., J. (2007). Spice : Web-based tools for rapid language adaptation in speech processing systems. *In Proceedings of Interspeech*, pages 2125–2128, Antwerp.

[Schultz et Waibel, 2001] SCHULTZ, T. et WAIBEL, A. (2001). Language-independent and languageadaptive acoustic modeling for speech recognition. *In Speech Communication*, volume 35, pages 31–51.

[Schwenk, 2007] SCHWENK, H. (2007). Continuous space language models. *Comput. Speech Lang.*, 21(3):492–518.

[Schwenk et Gauvain, 2000] SCHWENK, H. et GAUVAIN, J.-L. (2000). Combining multiple speech recognizers using voting and language model information. *In International Conference on Spoken Language Processing, Interspeech*, volume 2,, pages 915–918, Beijing, China.

[Schwenk et Gauvain, 2002] SCHWENK, H. et GAUVAIN, J.-L. (2002). Connectionist language modeling for large vocabulary continuous speech recognition. *In Acoustics, Speech, and Signal Processing.*

[Seng et Sam, 2005] SENG, S. et SAM, S. (2005). 2eme rapport du projet talk : Traitement automatique de la langue khmère. Rapport technique, Institut de Technologie du Cambodge.

[Stolcke, 2002] Stolcke, A. (2002). Srilm - an extensible language modeling toolkit. *In ICSLP*, pages 901–904, Denver, CO.

[Stücker et Schultz, 2004] Stücker, S. et Schultz, T. (2004). A grapheme based speech recognition system for russian. *In Proceedings of SPECOM*, St. Petersburgh.

[Suebvisai *et al.*, 2005] Suebvisai, S., Charoenpornsawat, P., Black, A., Woszczyna, M. et Schultz, T. (2005). Thai automatic speech recognition. *In in Proc. ICASSP*, pages 857–860.

[Theeramunkong *et al.*, 2000] Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T. et Chinnan, W. (2000). Character cluster based thai information retrieval. *In IRAL '00 : Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pages 75–80, New York, NY, USA. ACM.

[Vaufreydaz, 2002] Vaufreydaz, D. (2002). *Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue*. Thèse de doctorat, Université J. Fourier - Grenoble I,, France.

[Vaufreydaz *et al.*, 1998] Vaufreydaz, D., Akbar, M., Caelen, J. et Serignat, J.-F. (1998). Emacop environnement multimédia pour l'acquisition et la gestion de corpus parole. *In JEP'98*, pages 175– 178, Martigny, Switzerland.

[Wu, 2003] Wu, A. (2003). Chinese word segmentation in msr-nlp. *In Proceedings of the second SIGHAN workshop on Chinese language processing*, pages 172–175, Morristown, NJ, USA. Association for Computational Linguistics.

[Zhang *et al.*, 2008a] Zhang, R., Yasuda, K. et Sumita, E. (2008a). Chinese word segmentation and statistical machine translation. *ACM Trans. Speech Lang. Process.*, 5(2):1–19.

[Zhang *et al.*, 2008b] Zhang, R., Yasuda, K. et Sumita, E. (2008b). Improved statistical machine translation by multiple chinese word segmentation. *In StatMT '08 : Proceedings of the Third Workshop on Statistical Machine Translation*, pages 216–223, Morristown, NJ, USA. Association for Computational Linguistics.

[Zitouni, 2006] Zitouni, I. (2006). *Finite state based Arabic word segmentation*. CSLI Publications.

Résumé : Ce travail de thèse porte sur la reconnaissance automatique de la parole des langues peu dotées et ayant un système d'écriture sans séparation explicite entre les mots. La spécificité des langues traitées dans notre contexte d'étude nécessite la segmentation automatique en mots pour rendre la modélisation du langage n-gramme applicable. Alors que le manque de données textuelles a un impact sur la performance des modèles de langage, les erreurs introduites par la segmentation automatique peuvent rendre ces données encore moins exploitables. Pour tenter de pallier les problèmes, nos recherches sont axées principalement sur la modélisation du langage, et en particulier sur le choix des unités lexicales et sous-lexicales, utilisées par les systèmes de reconnaissance. Nous expérimentons l'utilisation des multiples unités au niveau des modèles du langage et au niveau des sorties de systèmes de reconnaissance. Nous validons ces approches de modélisation à base des multiples unités sur les systèmes de reconnaissance pour un groupe de langues peu dotées : le khmer, le vietnamien, le thaï et le laotien.

Mots-clés : reconnaissance automatique de la parole, langue peu dotée, modélisation statistique multi-niveau du langage.

Abstract : This PhD thesis focuses on the problems encountered when developing automatic speech recognition for under-resourced languages with a writing system without explicit separation between words. The specificity of the languages covered in our work requires automatic segmentation of text corpus into words in order to make the n-gram language modeling applicable. While the lack of text data has an impact on the performance of language model, the errors introduced by automatic segmentation can make these data even less usable. To deal with these problems, our research focuses primarily on language modeling, and in particular the choice of lexical and sub-lexical units, used by the recognition systems. We investigate the use of multiple units in speech recognition system. We validate these modeling approaches based on multiple units in recognition systems for a group of languages : Khmer, Vietnamese, Thai and Laotian.

Keywords : Automatic speech recognition, under-resourced language, multi-level statistical language modeling.