الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة وهران للعلوم و التكنولوجيا محمد بوضياف

THÈSE

En vue de l'obtention du

Diplôme de Doctorat en Sciences

Présenté par : MEZZOUDJ ep. BOUMAZZA Fréha

Intitulé

Contribution par des Méthodes Statistiques à l'Amélioration de la Reconnaissance Automatique de la Parole

Faculté ***: Mathématique et Informatique***

Département ***: Informatique***

Spécialité ***:Informatique***

Option ***:Reconnaissance des Formes et Intelligence Artificielle***

Devant le Jury Composé de :

Membres de Jury	*Grade*	*Qualité*	*Domiciliation*
RAHAL Sidi Ahmed Hebri	*Professeur*	*Président*	**USTO-MB**
BENYETTOU Abdelkader	*Professeur*	*Encadrant*	**USTO-MB**
BELKADI Khaled	*MCA*		**USTO-MB**
KOUNINEF Belkacem	*Professeur*		**INTTIC**
MEFTAH Boujellal	*MCA*	*Examinateurs*	**UMSM**
TLEMSANI Redouane	*MCA*		**INTTIC**

Année Universitaire : 2017-2018

الحمد لله وحده والصلاة والسلام على من لا نبي بعده

قال الله تعالى" :قالوا سبحانك لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم"

قال البويطي :"لما أتم الشافعي كتابه ناولنيه فقال:هاك، خذ هذا الكتاب على خطأ كثير فيه"، فقلت: يا أباعبد الله أصلحه لنا، فقال:" كيف؟وقد قال الله تعالى:"و لو كان من عند غير الله لوجدوا فيه إختلافا كثيرا "،أبى الله العصمة إلا لكتابه"

ذكر عن خالد المزني(كاتب الإمام الشافعي)أنه قال" :قرأت كتاب الرسالة على الإمام الشافعي ثمانين مرة، فما من مرة إلا وكان يقف على خطأ، فيقول الشافعي :هيه، أ ب الله أن يكون كتاباً صحيحاً غير كتابه."

Résumé

Beaucoup d'efforts ont été consacrés au développement des systèmes robustes de la reconnaissance automatique de la parole au cours des dernières décennies. Une importance particulière est dédiée à la modélisation du langage qui a conduit à la prédominance des méthodes statistiques. Ceci peut être expliqué par le changement d'objectifs dans la reconnaissance de la parole. Alors que la recherche de la reconnaissance vocale était initialement liée à la reconnaissance des phonèmes et des mots isolés formant un petit vocabulaire fermé, la reconnaissance vocale continue spontanée à grand vocabulaire est un problème plus important. Les modèles basés sur la grammaire sont de bonnes solutions pour des tâches très restreintes mais ils échouent en cas de grandes tâches de vocabulaire. Au lieu des systèmes fondés sur des règles de grammaires, il est recommandé d'utiliser des modèles de langage statistiques complets et robustes pour les tâches à vocabulaire important.

Selon l'état de l'art, l'amélioration de la reconnaissance automatique de la parole peut être réalisée selon l'amélioration au niveau des modèles acoustiques, le vocabulaire et/ou les modèles de langage en appliquant des adaptations ou des sélections de données pertinentes, etc.

C'est dans le cadre de ce dernier contexte que, la présente thèse s'inscrit en exploitant les modèles de langage selon les axes d'investigation suivants :

— Quelle est la faisabilité, la difficulté et l'intérêt d'utiliser les modèles standards de langage à base de n-grammes et à base de réseaux de neurones ?
— Quel est l'avantage d'utiliser la sélection des données textuelles pour la modélisation du langage pour la reconnaissance de la parole ? et comment procéder ?

Il important de noter que malgré la difficulté technique rencontrée lors de la réalisation de cette thèse, nous avons pu contribuer en :

— L'analyse et l'évaluation des modèles de langage standards n-gramme en utilisant différentes méthodes de lissage.
— L'analyse et l'évaluation des modèles neuronaux en utilisant différentes architectures directe simple, profonde, large et récurrente.
— La sélection des données textuelles pour une meilleure modélisation du langage pour la reconnaissance de la parole en adaptant le principe de différence d'entropie croisée.
— La proposition d'un nouveau critère pour la sélection des données textuelles à base du critère de la différence quadratique moyenne des probabilités.

Nos travaux ont montré l'importance des modèles de langage comme étant composantes linguistiques du système de la reconnaissance automatique de la parole et la possibilité de leur amélioration.

Mots clé : reconnaissance automatique de la parole ; modèle de langage ; n-gramme ; méthodes de lissage ; réseaux de neurones ; apprentissage profond ; sélection des données textuelles ; entropie croisée.

Abstract

A lot of effort has been put into developing robust ASR systems during last decades. In language modeling that led to the dominance of statistical methods. This is explained by the change of goals in speech recognition. While originally speech recognition research was bound to recognition of isolated words or strings of isolated commands that form small and closed vocabulary (i.e. recognition of numbers), at present the large vocabulary continuous speech recognition is an issue and words can not be regarded as isolated units. Grammar-based models are good solutions for very restricted tasks (for example, like in simple interactive voice response systems) but they fail in case of large vocabulary tasks. Less time and effort are needed to create a statistical LM ; at the same time the resulting model appears much more comprehensive and robust for large-vocabulary tasks, when compared to any rule-based one.

According to the state of the art, the improvement of automatic speech recognition can be achieved according to the improvement in the acoustic models, the vocabulary and/or the language models by applying adaptations, or Selection of relevant data, etc.

It is in this last approach that the present thesis fits in trying to exploit the language models along the following lines of investigation :

- What is the feasibility, the difficulty and the interest to use the standard models of language based on n-grams and the neural models ?

- What is the advantage of using textual data selection for language modelling for speech recognition ? And how ?

It is interesting to note that despite the technical difficulty, we contributed in this thesis by :

- Analysis and evaluation of n-gram standard language models using different smoothing methods.

- Analysis and evaluation of neural language models using different architectures (feed-forward, simple, large, deep and recurrent neural networks).

- The selection of textual data for improving language modelling dédicated to speech recognition by the principle of cross-entropy.

- The proposal of a new measure for the selection of textual data based on mean-square difference of probabilities.

Our work has shown the importance of the linguistic component of the automatic speech recognition system, the language model and the possibility of contributing to its improvement.

Key words : Automatic speech recognition ; language model ; n-grams ; smoothing methods ; neural networks ; deep learning ; textual data selection ; cross-entropy.

Dédicaces

Que ALLAH, mon dieu unique, clément et puissant protège et récompence mon cher mari Samir et mes enfants chéris Hind, Anes, Marwa et Mouaad, qui ont vécu de très prés la réalisation de cette thèse jour après jour, pour leurs aides, leurs encouragements et surtout leur patience envers moi durant la préparation de ce travail.

Vous avez toujours cru en moi, même dans les moments les plus difficiles, vous avez supporté mes humeurs au gré de cette thèse et vous avez dû faire beaucoup de sacrifices et de concessions : *merci beaucoup* et *c'est à vous que je dédie ce travail* !

Aussi, je le dédie à ma chère et gentille mère, qui me comble sans cesse par ses sincères prières.

Je le dédie egalement à l'âme de mon cher père, qui m'a toujours dirigé et encouragé pour me consacrer à la recherche scientifique.

Remerciments

Je suis très reconnaissante au Professeur BENYETTOU Abdelkader pour m'avoir mis sur le chemin de la recherche scientifique en général et m'a guidé particulièrement vers le sentier du domaine de la *reconnaissance automatique de la parole*. Aussi, je tiens à lui exprimer ma profonde gratitude pour ses conseils, son soutien technique et moral et sa patience envers moi pour la réalisation et la finalisation de cette thèse.

Mes remerciements et mes sincères salutations s'adressent aux honorables membres du jury de cette thèse. Je remercie le Professeur RAHAL Sidi Ahmed Hebri de l'université des Sciences et de Technologie d'Oran-Mohamed Boudiaf (USTO-MB), pour avoir accepter de présider le jury. Je remercie également le Professeur KOUNINEF Belkacem de l'Institut National des Télécommunications et des Technologies de l'Information et de la Communication d'Oran (INTTIC), le Docteur BELKADI Khaled de l'USTO-MB, le Docteur MEFTAH Boujellal de l'université Mustapha STAMBOULI de Mascara et le Docteur TLEMSANI Redouane de l'INTTIC d'Oran d'avoir pris le soin d'examiner et d'évaluer ce travail.

Je remercie le Docteur Denis JOUVET, Directeur de recherche du groupe Multispeech à LORIA de Nancy, et le Docteur David LANGLOIS, maître de conférence à l'université de Lorraine, pour leurs aides précieuses pendant mon stage de perfectionnement à INRIA-LORIA Nancy. Durant ce séjour, j'ai réalisé une partie importante de ce travail.

Aussi, je tiens à remercier spécialement le Professeur Kamel SMAILI, le Professeur Jean-paul HATON de l'université de Lorraine Nancy, le Professeur Holger SCHWENK, le Docteur Fethi BOUGHERAS de l'université de le Mans et le Docteur Andreas STOLCKE de *SRI International* pour leurs conseils et leurs disponibilités dans les moments de nécessité.

Je remercie sincèrement les services de la Post-graduation et l'administaration en général de l'Université des Sciences et de la Technologie d'Oran-Mohamed Boudiaf (USTO-MB) pour leur disponibilité. Je remercie mes collègues et amis à l'USTO-MB, particulièrement ceux du laboratoire SIMPA.

Je remercie sincèrement les services de la Post-graduation et l'administaration en général, ainsi que mes amis et mes collègues de travail à l'Université de Hassiba Benbouali de Chlef (UHBC) pour leurs divers aides.

Enfin, je remercie tous ceux qui m'ont aidé pour la réalisation de cette thèse d'une façon ou d'une autre !

Table des matières

Résumé ii
Abstract iii
Dédicaces iv
Remerciments v
Table des matières vi
Table des figures ix
Liste des tableaux x

1 Introduction générale **1**
1 Contexte de la thèse 1
2 Motivation et contribution 3
3 Structure du document 4

2 Linguistique **6**
1 Introduction 6
2 Linguistique 7
2.1 Contexte historique 7
2.2 Domaines de la Linguistique 9
3 Phonologie 10
3.1 Phonème 11
3.2 Allophone 12
4 Phonétique 13
4.1 Phonétique articulatoire 14
4.2 Phonétique acoustique 15
4.3 Phonétique auditive 17
5 Outils d'analyse de la parole 20
6 Conclusion 25

3 Reconnaissance Automatique de la Parole **26**
1 Introduction 26
2 Contexte historique 27
3 Formulation probabiliste d'un système de RAP 28
4 Composantes d'un système de RAP 29
4.1 Analyse acoustique 30
4.2 Modéle acoustique 34
4.3 Modèle de Langage 46
4.4 Algorithme de décodage 48
5 Evaluation d'un système de RAP 51
5.1 Mesures d'erreurs 52
5.2 Intervalle de confiance 53

6 Transcription de la parole . . . 53
7 Compagnes d'évaluation de la RAP . . . 55
8 Plateformes pour les Systèmes de LVCSR . . . 59
9 Systèmes populaires de la RAP . . . 60
9.1 Système de transcription de LIUM . . . 60
9.2 Système de la transcription de LIMSI . . . 61
9.3 Système de la transcription de LORIA . . . 61
10 Conclusion . . . 62

4 Modèles de Langage 64
1 Introduction . . . 64
2 Traitement du Langage Naturel et la Modélisation du Langage . . . 65
3 Ressources pour les Modèles de Langage . . . 66
3.1 Corpus de textes . . . 66
3.2 Vocabulaire . . . 67
4 Évaluation des Modèles de Langage . . . 70
5 Modèles de Langage n-gramme . . . 71
5.1 Estimation des Modèles de Langage n-gramme . . . 73
5.2 Méthodes de Lissage des Modèles de Langage n-gramme . . . 74
6 Modèles de Langage avancés . . . 78
6.1 Modèles de Langage spatiaux discrets . . . 78
6.2 Modèles de Langage neuronaux continus . . . 81
7 Expérimentations et évaluation . . . 89
7.1 Corpora et outils utilisés . . . 90
7.2 Estimation et lissage des ML n-gramme . . . 92
7.3 Modèles de Langage n-gramme . . . 96
7.4 Modèles de Langage neuronaux . . . 98
8 Conclusion . . . 101

5 Sélection des données textuelles pour les modèles de langage 103
1 Introduction . . . 103
2 Mesures de confiance et analyse des erreurs . . . 104
3 Modèles neuronaux profonds de parole . . . 105
4 Sélection des données de parole . . . 108
5 Sélection des données textuelles . . . 109
6 Expérimentations sur la sélection des données textuelles . . . 111
6.1 Données textuelles utilisées . . . 112
6.2 Pré-sélection sur le corpus complet . . . 114
6.3 Stratégie de sélection des données textuelles . . . 116
6.4 Expérimentation sur le corpus complet . . . 117
6.5 Expérimentation de la transcription . . . 122
6.6 Exploration des données textuelles . . . 123
6.7 Nouveaux critères proposés pour la sélection des données textuelles 128
7 Conclusion . . . 135

6 Conclusion générale 137
1 Conclusion . . . 137
2 Perspectives . . . 138

Contribution Personnelle **139**

Bibliographie **140**

Table des figures

2.1 Domaines de la linguistique selon la représentation statique. 10
2.2 Domaines de la linguistique selon la représentation dynamique. 10
2.3 Modèle mécanique de la production de la parole. 14
2.4 Vue de l'appareil phonatoire. 15
2.5 Principe du modèle source –filtre de Fant. 16
2.6 Courbes d'égale sensation sonore. 18
2.7 Anatomie de l'oreille humaine. 19
2.8 Oscillogramme et Spectrogramme d'une onde acoustique. 22
2.9 Spectrogramme à bandes étroite et large 23

3.1 Architecture des composantes d'un système de RAP. 29
3.2 Processus d'extraction des 39 coefficients MFCC 33
3.3 Un modèle acoustique d'un mot. 35
3.4 Un exemple d'HMM gauche-droit. 37
3.5 Une simple gaussienne et un mélange de deux gaussiennes. 39
3.6 Exemple d'un HMM avec des fonctions de densités GMM. 39
3.7 Variantes de prononciation pour la séquence de mots *Premier Ministre*. . . 43
3.8 Le graphe de l'algorithme de *Viterbi*. 49
3.9 Un exemple de graphe de mots. 52

4.1 Exemple de N-gramme de mots et de caractères. 72
4.2 Architectures du MLP et du RNN. 82
4.3 Représentation distribuée de 2-dimensions des mots. 84
4.4 Représentations des caractéristiques espacées. 85
4.5 Architecture directe d'un ML neuronal. 86
4.6 Modèle de Langage à base de réseau de neurones récurent. 88

5.1 Perplexité des MLs générés par trois types de sélection. 115
5.2 Pricipe de la sélection de donnée par le critère $dXent$. 116
5.3 Perplexité du ML-Test appris par les données du Gw sélectionnées. 117
5.4 Principe de la sélection des données par l'approche 1. 118
5.5 Principe de la sélection des données par l'approche 2. 119
5.6 ML-Test appris sur les données sélectionnés du corpus complet. 120
5.7 Principe de la sélection des données par l'approche 3. 121
5.8 ML-Test appris sur les données sélectionnés du Gw + les autres corpora. . 122
5.9 Principe de la reconnaissance du système de transcription -Loria 123
5.10 Principe de la sélection des données par le critère de $MSDP$ 130
5.11 Perplexité du ML-Test appris sur les données sélectionnées du Web. 131
5.12 Perplexité du ML-Test appris sur les données sélectionnées du Np. 132
5.13 Perplexité du ML-Test appris sur les données sélectionnées du Gw. 133

Liste des tableaux

2.1 Les Voyelles et les Consonnes de l'Anglais. 11
2.2 Les Voyelles et les Consonnes du Français. 12
2.3 Tableau de l'alphabet phonétique international. 13

4.1 Taille du corpus ETAPE. 90
4.2 Taille du corpus NAACL. 91
4.3 Perplexité pour les differents ordres n pour le corpus ETAPE. 92
4.4 Perplexité pour les différents ordres n pour le corpus NAACL. 92
4.5 Utilisation du vocabulaire et du *UNK* pour les OOV avec le corpus ETAPE. 93
4.6 Utilisation du Vocabulaire et du *UNK* pour les OOV pour le corpus NAACL. 93
4.7 Optimisation des paramètres de le lissage additif avec les données ETAPE. 94
4.8 Influence du hyper-paramètre du décompte absolu avec les données ETAPE. 95
4.9 Optimisation du hyper-paramètre du lissage additif sur les données NAACL. 95
4.10 Optimisation du hyper-paramètre du décompte absolu sur les données NAACL. 96
4.11 Perplexité des ML avec différents lissages sur les données de ETAPE. . . . 97
4.12 MLs avec différents lissages sur les données NAACL. 97
4.13 Perplexité du ML sur les données ETAPE. 100
4.14 Perplexité du ML sur les données de ETAPE. 100

5.1 Corpus textuel disponible de Mutlispeech (2015). 112
5.2 Statistiques des fichiers de *développement* et de *test*. 112
5.3 Perplexités des ML entrainés avec les differentes sources de données. 113
5.4 ML reférentiel, interpolé à partir des MLs individuels. 114
5.5 Poids et perplexités des MLs par sources de données. 115
5.6 Modèle de Langage obtenu par l'approche 2. 120
5.7 Modèle de Langage obtenu par l'approche 3. 121
5.8 Résultats de la transcription de la parole sur le corpus ETAPE Dev. 124
5.9 Perplexités des ML appris sur les données Tr et Web utilisées. 125
5.10 Taille par mots (sans <s> et </s> des fichiers Web. 125
5.11 Les données Tr, Web et Np pour l'apprentissage des MLs. 126
5.12 Taille par mots (sans <s> et </s> des fichiers Np. 126
5.13 Perplexités de ML appris avec les données de Tr et Np. 127
5.14 Perplexités des ML appris sur les données de Tr et Gw. 127
5.15 Taille par mots sans <s> et </s> des fichiers Gw. 128
5.16 ML(1) appris sur des données sélectionnées par la $dXent$. 133
5.17 ML(2) appris sur des données sélectionnées par la $dXent$. 134
5.18 ML(3) appris sur des données sélectionnées par la $dXent$. 134
5.19 ML(4) appris sur des données sélectionnées par la $dXent$. 134

Chapitre 1

Introduction générale

1 Contexte de la thèse

Depuis longtemps, la parole est le moyen principal de communication entre les humains. Les recherches en traitement de la parole consistent à reproduire automatiquement les capacités d'un être humain à extraire des informations du flux de la parole produite par un autre être humain. Cette tâche ne peut être effectué par un seul et simple système informatique mais elle est plutôt subdivisée en plusieurs sous-problèmes relatives au type d'information à extraire et à reconnaître.

Selon Jean-Paul Haton et al. [Haton et al., 2006], le terme de *traitement automatique de la parole*, englobe plusieurs thèmes qui peuvent utiliser des techniques similaires et peuvent même être utilisé simultanément dans une même application tel que : la *reconnaissance automatique de la parole*, le traitement et la reconnaissance des dialectes d'une langue, le codage et la compression de la parole, la synthèse de la parole, la reconnaissance et la vérification du locuteur, l'identification de la langue, la détermination de l'état émotionnel d'un locuteur, etc.

La reconnaissance automatique de la parole est particulièrement le processus par lequel un ordinateur transforme un signal acoustique de parole en texte ou en action, ce qui facilitera la communication entre les humains et la machine. Brièvement et selon Bristow (1986), nous pouvant dire que : *Automatic speech recognition is about computers learning how to communicate with humans, rather than vice versa.*

Le domaine de la reconnaissance automatique de la parole se caractérise par un développement avancé d'applications pratiques pluridisciplinaires allant de la dictée vocale à la téléphonie et en passant par plusieures applications de dialogue homme-machine qui touche différents secteurs : militaire, éducatif, sanitaire, industriel, etc.

Beaucoup d'efforts ont été consacrés au développement des systèmes de reconnaissance automatique de la parole robustes au cours des dernières décennies en se focalisant sur la modélisation du langage qui a conduit à la prédominance des méthodes statistiques. Ceci est expliqué par le changement d'objectifs dans la reconnaissance de la parole. La recherche de la reconnaissance vocale était initialement liée à la reconnaissance des phonèmes, des mots isolés ou de chaînes de commandes isolées qui forment un petit vocabulaire fermé. Cependant, la reconnaissance vocale continue à grand vocabulaire est un problème important où les mots ne peuvent être vus comme des unités isolées. Les modèles de langage basés sur la grammaire sont de bonnes solutions pour des tâches très restreintes par exemple, comme dans les systèmes simples de réponse vocale interactive, mais ils échouent en cas tâches à grand vocabulaire. Il faut moins de temps et d'efforts pour créer un modèle de langage statistique. Aussi, ce genre de modèle résultant apparaît

beaucoup plus complet et robuste pour les tâches à vocabulaire important, par rapport à tout système dr reconnaissance fondé sur des règles de grammaires. Les efforts de recherche continus et toujours en expansion permettent à ce domaine de rester toujours intéressant et attirant pour de nouveaux chercheurs.

Nous nous intéressons dans cette thèse à la *reconnaissance automatique de la parole* et à la contribution pour son amélioration, du fait que plusieurs problèmes relatifs à la variabilité de la parole caractérisent ce domaine de *difficile* jusqu'à l'heure actuelle, malgré les niveaux conceptuels et technologiques élevés acquis par les humaines :

- Le signal de la parole présente différents types de variabilité :
  - variabilité intra-locuteur dûe au mode d'élocution (accent régional, physiologie du locuteur, style (préparé ou spontanée), vitesse de la parole (rapide, moyenne, lente), age du locuteur et son état emotionel, etc.).
  - variabilité inter-locuteur dûe aux différences entre locuteurs (du même sexe ou de sexe différent).
  - variabilité dûe au moyen d'acquisition du signal (type de microphone ou téléphone), de la qualité de la transmission du signal (courte ou longue distance) ou aux différences entre environnements acoustiques (présence de bruits, de musique, etc.).
- Chaque son élémentaire ou phonème peut être modifié par son contexte : le phonème qui le précède et celui qui lui succède.
- Il est difficile de déterminer à priori le nombre de mots constituant la phrase et leurs frontières.
- Il est nécessaire de traiter une grande quantité de données phonétiques, pour entraîner un bon système de reconnaissance, ce qui entraîne une grande complexité des calculs même avec l'utilisation d'algorithmes de traitements sélectifs.

Aussi, ces problèmes qui ne sont pas exhaustives font de la reconnaissance automatique de la parole un domaine particulier et étroitement lié à d'autres disciplines : la linguistique, la phonologie, la phonétique, le traitement du langage naturel, le traitement du signal, l'intelligence artificielle, l'apprentissage automatique, la reconnaissance des formes, etc.

Selon Rabiner, la reconnaissance automatique de la parole se développent depuis les années trente et s'étale sur 5 générations [Rabiner and Juang, 2008]. Ce domaine utilise des méthodes et des modèles de plus en plus combinés et parallèles et des corpus de parole de plus en plus complexes et grands.

Historiquement, les premiers systèmes de reconnaissance automatique de la parole permettaient de reconnaître des mots isolés. Une des manière les plus simples pour résoudre ce problème est de procéder par comparaison entre le signal observé et une réalisation de référence de chaque mot et choisir la réalisation la plus proche du signal correspond en identifiant les phonèmes par des formants, des traits distinctifs ou globalement des mots en se basant sur les propriétés spectrales au cours du temps. Par la suite est apparu la reconnaissance automatique de la parole continue où les mots présents dans le message audio ne sont pas séparés pas de pause entre les mots. C'est en fait une manière quasi-naturelle et proche à la façon dont les êtres humains s'expriment. Cette tâche nécessitait l'analyser du message audio dans son ensemble. Les premières études concernant la parole continue traitaient de la parole lue avec des conditions contrôlées telles que : parole lue et stable, enregistrements contrôlés de quelques dizaine de mots prononcés de manière stable par un locuteur sélectionné, grammaire rigoureuse, pas de perturbation du signal dûes par exemple à l'émotion, à des hésitations ou a des changements non-voulus de la position du locuteur par rapport au micro, enregistrement effectué dans un studio avec du matériel d'enregistrement adapté.

Les chercheurs du domaine sont passés à l'exploitation de milliers d'heures de parole spontanée et/ou conversationnelle de locuteurs connus ou inconnus dans des conditions variées, disponibles sur les chaînes de radios et de télévisions ou lors des réunions. La parole spontanée est la parole que l'on rencontre quotidiennement. Il s'agit de messages non préparés, énoncés par les locuteurs de manière vraiement naturelle. Tout d'abord la parole est souvent bruitée, le matériel d'acquisition peut être différent d'un locuteur à l'autre, il arrive qu'il y ait un fond musical à la parole, les mêmes locuteurs peuvent être enregistrés dans différentes conditions, il peut y avoir plusieurs locuteurs qui parlent en même temps, ils peuvent hésiter ou bredouiller, etc. De plus, différents styles de parole peuvent se côtoyer : de la parole préparée, presque lue, à la parole très spontanée. Le style est également très différent du style employé dans des documents écrits. Toutes ces contraintes rendent la transcription de la parole spontanée difficile. Les données préférables à utiliser pour cette tâche sont les enregistrements de conversations, d'interveiws et même de réunions.

Par la suite, il était nécessaire que les systèmes et les algorithmes qui ont été développés pour la parole lue commencaient à être adaptés à la parole spontanée. Les ordinateurs en réseau de plus en plus performants avec des capacités de stockage toujours croissantes, l'utilisation des méthodes informatiques spécifiques et adaptées d'un formalisme mathématique puissant ont contribué efficacement à pousser la recherche dans le domaine de la reconnaissance automatique de la parole. Cependant, ces efforts et ces outils ne sont certainement pas les seules explications de ces progrès. Aussi, l'évaluation systématique des moteurs de reconnaissance pour différentes tâches relatives à la parole dans le cadre de grandes campagnes internationales et européennes (DARPA/NIST aux Etat-unis, ESTER et ETAPE en France, etc.) a soutenue efficacement les progrès grâce à la production de corpus communs partagés par la communauté scientifique. Sachant que le traitement et la reconnaissance automatiques de la parole, et en particulier la transcription automatique, sont gourmands en corpus oraux, mais également en ressources textuelles écrites et en transcriptions de l'oral.

2 Motivation et contribution

Un système de reconnaissance de la parole et précisément *à grand vocabulaire* est constitué grossièrement de plusieurs modules : des modèles acoustiques, des modèles de langage ou dits linguistiques, un vocabulaire et un module de recherche spécifique. Ce système transcrit le signal acoustique et génère en sortie la suite de mots la plus probable connaissant les modèles acoustiques et linguistiques. Cette suite de mots est issue du parcours d'un graphe de mots généré par un algorithme de recherche qui permet de trouver les N-meilleures séquences de mots en fonction de leurs scores acoustique et linguistique.

De nombreux travaux ont été effectués et différentes méthodes d'élagage, d'optimisation et d'adaptation sont proposées pour améliorer les différents modules des systèmes de reconnaissance de la parole à grand vocabulaire, visant à améliorer leurs qualités. Malgré tout ces efforts, les performances même si elles sont améliorées, restent encore insuffisantes et difficile à atteindre lorsque la parole à reconnaitre est spontanée, conversationnelle, bruitée et/ou enregistrée dans des conditions naturelles, réelles et/ou différentes au données de l'apprentissage.

L'amélioration de la reconnaissance automatique de la parole continue à grand vocabulaire (*Large Vocabulary Continuous Speech Recognition* - LVCSR) peut être réalisée selon les approches statistiques suivantes :

— Amélioration au niveau des modèles acoustiques, en appliquant des adaptations sur les caractéristiques audio ou sur les modèles acoustiques eux même, ou en sélectionnant les données audio les plus pertinents pour la reconnaissance, ou encore utiliser pour l'apprentissage de ces modèles au lieu des modèles de Markov (Gaussien Markov Model - GMM) d'autres outils tels que : les réseaux de neurones, etc.
— Enrichir le vocabulaire en intégrant le plus de mots possibles pour la transcription et/ou prétraiter les noms propres et autres pour minimiser les mots autres que le vocabulaire considéré (*Out Of Vocabulary* - OOV).
— Amélioration au niveau des modèles de language, en appliquant des adaptations, ou des sélections de données textuelles pertinentes, ou en utilisant pour l'apprentissage au lieu des modèles standard n-gramme d'autres modèles pour la modèlisation du langage tels que : les réseaux de neurones (*Deep learning*), etc.

Dans le cadre de cette thèse, c'est sur cette dernière approche que nous concentrons nos efforts en essayant d'explorer à fond les modèles de langage dédiés pour la reconnaissance de la parole. Plus précisement, afin de contribuer à l'objectif d'améliorer le système de reconnaissance de la parole dédié à la transcription des émissions radiophoniques et télévisées utilisées durant la campagne d'évaluation ETAPE par le Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA), nous nous consacrerons sur la modélisation statistique du langage.

Les axes d'investigation suivants nous intéressent le plus :

— Quelle est la faisabilité, la difficulté et l'intérêt d'utiliser des modèles de langage n-gramme avec differentes méthodes de lissage et les modèles de langage neuronaux ?
— Quelle est l'avantage d'utiliser la sélection des données textuelles pour la modèlisation du langage dédiée à la reconnaissance de la parole ? et comment procéder ?

Il est intéressant de noter qu'une difficulté technique majeure réside dans la nécessité d'utiliser du matériel informatique puissant et parallèle avec des corpus de données (audio/textuels) trés volumineux pour une réelle évaluation de la reconnaissance de la parole. Malgrés qu'une telle situation n'était pas toujours présente durant l'élaboration de notre travail, nous avons réalisé durant cette thèse les contributions scientifiques suivantes :

— L'analyse et l'évaluation des modèles de langage standards à base de n-gramme en utilisant différentes méthodes de lissage.
— L'analyse et l'évaluation des modèles de langage neuronaux en utilisant différentes architectures (directe, simple, large, profonde et récurrente).
— La sélection des données textuelles pour une meilleure modèlisation du langage pour le système de reconnaissance de la parole de LORIA - Nancy par le critère de l'entropie croisée de moore [Moore and Lewis, 2010].
— La proposition d'une nouvelle mesure pour la sélection des données textuelles.

Nos travaux ont montré l'importance des *modèles de langage* qui sont les composantes linguistiques d'un système de la reconnaissance automatique de la parole et les différentes possibilités de leurs améliorations.

3 Structure du document

Dans le deuxième chapitre, une présentation générale de la linguistique, la phonologie, la phonétique et leurs différents domaines est introduite. Dans le troisième chapitre, l'architecture des systèmes de la reconnaissance automatique de la parole continue à grand vocabulaire est présentée. Nous avons fait le point sur un état de l'art des particularités

et des méthodes d'adaptation utilisées pour la reconnaissance de la parole spontanée et conversationnelle à grand vocabulaire. Nous nous sommes intéressés également au système de transcription du LORIA, le Laboratoire Lorrain de Recherche en Informatique et ses Applications, sur lequel s'appuie des expériences importantes menées durant ce travail de thèse. Le quatrième chapitre est dédié aux modèles statistiques de langage ou dits modèles linguistiques, ce module du système de reconnaissance nous intéresse particulièrement le plus dans notre travail. Un état de l'art sur les différentes variantes de ces modèles de langage ainsi que les techniques relatives, proposées dans la littérature est présenté : les modèles statistiques n-gramme, leurs méthodes de lissage, les modèles statistiques neuronaux directs à base de perceptron multi-couches et à base d'architecture récurrente, etc. A ce niveau, nous présentons une comparaison des différents modèles de langage exploités et entraînés en utilisant différentes méthodes de lissage et différente techniques de modèlisation sur deux corpus textuels. Dans le cinquième chapitre, nous introduisons les différentes méthodes statistiques utilisés pour l'amélioration des systèmes actuels de la reconnaissance automatique de la parole. Aussi, nous exposons les travaux spécifiques que nous avons menés afin d'améliorer les performances du système de reconnaissance. La méthodologie utilisée et les résultats obtenus concernant la sélection des données textuelles pour la modélisation du langage à base de n-gramme dédié à la transcription de la parole sont présentés. Enfin, une conclusion générale est faite pour récaputuler les importants concepts traités dans cette thèse et les perspectives proposés.

Chapitre 2

Linguistique

1 Introduction

Toutes les sociétés humaines pratiquent au moins une langue. On en dénombre actuellement plus de 6800 langues parlées [1] dans le monde mais environ 350 seulement écrites, dont beaucoup sont en voie de disparition faute de locuteurs. Chez les humains, l'acquisition du vocabulaire se poursuit tout au long de la vie. Un être humain normalement constitué et inséré depuis sa naissance dans un groupe social est capable vers l'âge de cinq ans [2] de tenir et de comprendre une conversation courante dans sa langue maternelle ; même bien avant qu'il ne maîtrise le raisonnement.

Le rêve de dialoguer naturellement avec des machines comme avec ses semblables n'a pas cessé de séduire les humains. En 1950, Turing [3], le père fondateur de l'informatique, prédit que dans 50 ans, les ordinateurs auront acquis cette capacité. L'échéance est passée sans que la prédiction ne se réalise complètement. Pourtant, dans le domaine de la technologie numérique, beaucoup de rêves apparemment plus difficiles à réaliser ont été largement dépassés. Aujourd'hui, les ordinateurs battent les grands maîtres d'échecs facilement mais ils n'ont toujours pas les compétences langagières d'un enfant de 5 ans. Comment cela se fait-il ? En quoi la capacité de bien parler et bien reconnaître un langage est-elle si difficile à simuler ? Malgré que l'informatique a contribué largement avec des travaux de recherche sur ce sujet et ceci depuis ses tout débuts.

Les langues naturelles sont avant tout orales, beaucoup n'ont d'ailleurs pas de transcription écrite. Il est donc naturel que de nombreuses propriétés de ces langues découlent de considérations acoustiques ou dites phonologiques. La linguistique qui est une science ancienne étudie les langages naturels : la phonétique étudie la prononciation réelle alors que la phonologie étudie le système qui sous-tend cette prononciation.

Immatérielle et invisible, le son de la parole (ou l'onde sonore) possède la vertu d'être partout. Elle emplit l'espace, contourne les obstacles et traverse les parois. Elle est omniprésente [4] et peut revêtir dans un environnement donné une infinité de formes diverses. L'oreille qui possède de remarquables capacités d'adaptation à son environnement a pour habitude de sélectionner une partie des informations – dans le temps comme dans l'espace – pour les amplifier, les moduler ou les classer afin de mieux les intégrer dans le contexte

1. https ://www.survivalinternational.fr/actu/3087, consulté en dec. 2017
2. http ://correspo.ccdmd.qc.ca/Corr5-2/Allo.html#a2, consulté en dec. 2017
3. Turing (1950) dit : *"Nevertheless I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted"*
4. présente partout et tout le temps

suggéré par le message sonore initial ou imaginé par l'auditeur. Cette plasticité auditive permet au sujet d'interpréter le message et de l'hiérarchiser ainsi que les informations qu'il contient. Dès que le cerveau a reconnu leur signification cognitive, culturelle ou affective, il leur attribue des valeurs relatives qui deviennent indépendantes de la nature du message et de la voie sensorielle empruntée [Mercier, 2010].

Ce chapitre est destiné à faire un tour d'horizon sur ces concepts. Nous commençons par dresser un état de l'art de la linguistique qui est la science mère des langages écrits et parlés, puis introduire ses différentes branches : la phonologie, les phonétiques articulatoire, acoustique et auditive, etc. ce qui nous facilitera par la suite un passage aisé au domaine de la *Reconnaissance Automatique de la Parole*.

2 Linguistique

La linguistique comme définie par le dictionnaire de la linguistique de Georges Mounin, édition, PUF, 2004 : est la science du langage, c'est-à-dire étude objective, descriptive et explicative de la structure du fonctionnement (linguistique synchronique) et de l'évolution dans le temps (linguistique diachronique) des langues naturelles humaines. Elle s'oppose ainsi à la grammaire (descriptive et normative) et la philosophie du langage (hypothèses métaphysique, biologiques, psychologiques, esthétiques sur l'origine, le fonctionnement, la signification anthropologiques possibles du langage)".

2.1 Contexte historique

Vu l'importance de la langue dans les sociétés des humains, l'étude de ses concepts a commencé tôt. Les dates que nous citons ne sont que des points de repère marquants cette discipline [Alghamdi, 2000], [Tellier, 2008], [Barkat, 2000].

On trouve des témoignages de réflexions sur le langage dès l'Antiquité. Au IVe s. av. J.-C, à l'Inde, Panini (560-480 a.v. J.-C.) a réalisé une première analyse de sa langue maternelle ; le sanskrit. Il a formulé dans son ouvrage intitulé *Ashtadhyayi* les règles de morphologie, de syntaxe et de sémantique de cette langue. Il a discuté des sorties de sons et l'influence des uns sur les autres.

Aussi, des tentatives de trouver la relation entre les noms et leurs significations ont été menés par des philosophes grecs. Au IVe s. av. J.-C., Aristote a introduit des réflexions sur la langue dans son ouvrage : *De l'interprétation in Organon*. Vers Ve-IVe s. av. J.-C., Platon a résumé ses idées dans *Lettre VII in oeuvres complètes*. Au Ier s. av. J.-C., Dionysius Thrax a réalisé un ouvrage complet sur les règles de la langue grec qui est resté une référence incontournable pendant presque mille ans. A leur tour, les savants romains ont contribué à fonder les règles de leur langue : à l'IIe-Ier s. av. J.-C., Varron a introduit ses réflexions sur la langue dans son ouvrage : *De la langue latine*.

Vers la fin du XVIIIe siècle - début du XIXe, la linguistique s'est développée pour devenir une science à part entière. La linguistique historique et comparative a apparu [Fiala, 2004], son objectif est d'établir une parenté génétique entre les familles de langues qui lui sont attribuées quelques nominations bien plus tard [5].

5. La famille des langues *Chamito-Sémitiques* (dites aussi afro-asiatiques) est un groupe de langues parlées dès l'Antiquité au Moyen-Orient, au Proche-Orient et en Afrique du nord. Leurs nominations est d'après les noms *cham* et *Sem*, deux fils du prophète Noé. Les langues les plus répandues de cette famille sont : l'arabe, l'amharique, le tigrigna, l'hébreu, le maltais, etc. Aussi, la famille des langues *Indo-Européennes* qui regroupe la plupart des langues parlées aujourd'hui en Europe, mais aussi l'hindi, le persan et le sanskrit, ou encore des langues mortes comme le latin ou le hittite.

Avec l'avènement de l'Islam au 7ème siècle et pour une grande part à cause de la doctrine concernant le miracle de l'inimitabilité linguistique du *Quran* ; qui est la parole divine relève à l'origine de l'oralité, les savants musulmans arabes et non-arabes se sont mis à étudier la langue arabe profondément et complètement.

La première tentative d'écrire une grammaire arabe a commencé avec les travaux du grammairien *Abu Al-Aswad Al-Duali* (603–688) qui a rajouté des points sur ou sous les mêmes anciennes lettres, pour passer de 18 lettres d'alphabet à 28.

A son tour, le philologue et le poète *Khalil Ibn Ahmed Al Farhidi*(718-791) qui a introduit le système diacritique de l'arabe a publié le premier dictionnaire arabe *Kitab al-Ayn* (Le livre source), où il a cherché à élucider l'origine des mots arabes. Les mots ne sont pas rangés par ordre alphabétique, mais selon la phonétique, par rapport à la localisation de la prononciation du son le plus profond dans la gorge (ayn) au plus labial (mim).

Le grammairien *Abu Bichr Amr ibn Uthman ibn Qanbar Al-Bichr* (760-796) connu sous le nom de *Sibawayh* a rédigé le livre *Al-Kitâb*, où il a introduit une description des phonèmes arabes et des seize lieux d'articulation relatives. Cette étude été basée sur trois sources considérées comme fidèles à la norme : le *Quran*, la poésie antéislamique et le parler des bédouins originaires de la région de l'auteur (*Al-Basra* au sud de l'Irak).

Aussi, le théologien *Abu Bakr Ibn Mugahid* (857 - 922), *Ibn Jinni*(932-1002), *Ismaïl ibn Hammad al-Jauhari* (-1002), *Ibn Fâris* (1004-), *Abu Muhammad al-Qasim ibn Ali al-Hariri* (1054-1122) et *Ibn Abul-Fadl Jamal ad-Din Muhammad Ibn Manzur* (1233- 1311) ont contribué largement à la linguistique [Catineau, 1960], [Bohas, 2014] et la phonétique arabe.

Le souci majeur de ces travaux était de démontrer les mécanismes de l'articulation des phonèmes arabes en donnant pour chacun d'entre eux la description articulatoire la plus exacte pour assurer une bonne prononciation lors de la lecture coranique. Ainsi, les premières règles de la langue arabe, sa syntaxe, sa morphologie et sa phonologie furent composées doucement et sûrement grâce à ses savants et bien d'autres.

En Europe, les années 1644- 1660 ont été marqué par la publication de l'ouvrage : *Grammaire générale et raisonnée* connue sous le titre de *Grammaire de Port-Royal*, d'Arnaud et Lancelot. Son ambition était de décrire les règles du langage en termes de principes rationnels universels.

Cependant il a fallu attendre le XXe siècle pour voir se dégager une approche scientifique et une conception structurale du langage. Précisément en 1916, deux étudiants du linguiste suisse Ferdinand de Saussure (1857-1913) ont publié ses notes de cours sous le titre de *Cours de linguistique générale* [De Saussure, 1989] qui est devenu un classique dans ce domaine. Depuis cet événement, Saussure est considéré comme le père de la linguistique moderne :

- Il caractérise le langage comme la construction sociale d'un système de signes. Un signe est l'association arbitraire entre un signifiant (défini comme l'image acoustique d'un mot) et un signifié (un concept, la représentation mentale d'une chose). Les signes font sens par les rapports qu'ils entretiennent les uns avec les autres dans le système des signes.
- Il considère que le langage en tant que faculté générale de s'exprimer au moyen de signes se distingue de la parole qui serait plutôt l'utilisation concrète de signes linguistiques particuliers.

Pendant les années 1930-1940, le *cercle de Prague*[6] prolonge les analyses de Saussure et

6. www.universalis.fr/encyclopedie/cercle-de-prague/, consulté en 14 Janvier 2017.

contribue à une linguistique structurale. Ses membres les plus connus Roman Jakobson (1896-1982) et Nicolas Troubetzkoy (1890-1938) ont contribué à l'invention de la phonologie qui se base sur l'étude des sons élémentaires : les phonèmes qui jouent le rôle d'unités distinctives dans une langue donnée.

En 1974, le linguiste français André Martinet (1908-1999) a publié son ouvrage *Éléments de linguistique générale*. Quelques années au paravent, il a été connu pour avoir caractérisé les langues naturelles par la propriété de la double articulation. Toutes les langues humaines sont doublement articulées parce qu'elles combinent des éléments (discrets) à deux niveaux différents [Martinet, 1960] :

— la *première articulation* syntaxique est celle qui permet la combinaison d'unités douées chacune d'une forme vocale et d'un sens qu'on appelle des *monèmes*. Ce terme n'est plus vraiment employé, on utilise plutôt celui de *morphème* ou *mot*.

— la *deuxième articulation* décrit comment chaque *monème* est lui-même décomposable en une succession d'unités phonétiques élémentaires dépourvues de sens qu'on appelle les *phonèmes*.

Parmi les écoles apparu par la suite, celle de Noam Chomsky (1928) linguiste américain d'une productivité exceptionnelle et professeur de linguistique au MIT (*Massachusetts Institute of Technology*) depuis 1961. A travers les différentes théories qu'il a contribué à élaborer, il a toujours favorisé la syntaxe sur tous les autres niveaux d'analyse du langage. Il a toujours pensé que les hommes disposent à la naissance d'un organe du langage de nature mentale. Dans cette perspective, apprendre une langue particulière revient à instancier une grammaire universelle innée.

Les travaux de Chomsky ont marqué l'histoire de la syntaxe des 50 dernières années :

— En 1957 : publication de *Syntactic Structures* [Chomsky, 1969] qui est considéeé comme ouvrage fondateur dans le domaine. Dans les années 60, l'approche se raffine dans la théorie des *Generative Grammar* qui devient ensuite dans les années 70 la théorie standard de la syntaxe.

— Dans les années 80 : l'approche *Principle and Parameters* précise la nature de la grammaire universelle innée postulée par Chomsky. Ainsi, apprendre une langue particulière revient à acquérir son vocabulaire spécifique et à identifier la valeur des paramètres qu'elle instancie. Cette théorie sera connu ensuite sous les termes : *Gouvernment and Binding*.

— Depuis 1995, Chomsky développe un programme minimaliste qui est une reformulation de ses théories précédentes mais orientée vers un principe d'économie.

2.2 Domaines de la Linguistique

La linguistique moderne regroupe un certain nombre d'écoles qui ont toutes en commun le fait d'avoir le *langage* comme objet d'étude mais qui n'abordent pas forcément les problèmes du même point de vue.

Dans un langage, les *morphèmes* constituent la première articulation : ce sont les plus petits éléments significatifs ayant une forme et un sens (voir figure 2.1). Les *phonèmes* constituent la deuxième articulation : ce sont les éléments non significatifs ayant une forme mais aucun sens. Donc, d'après le principe de Martinet [7], on peut classer les domaines de la linguistique selon la dynamique de ses constituants. Ce principe est illustré dans la figure 2.2 :

La phonétique et la phonologie sont deux sciences distinctes [8] mais qui partagent un

7. http ://www.cairn.info/revue-la-linguistique-2001-1-page-5.htm, consulté en 15-1-2017.
8. http ://www.linguistes.com/phonetique/phon.html, consulté en 15-01-2017.

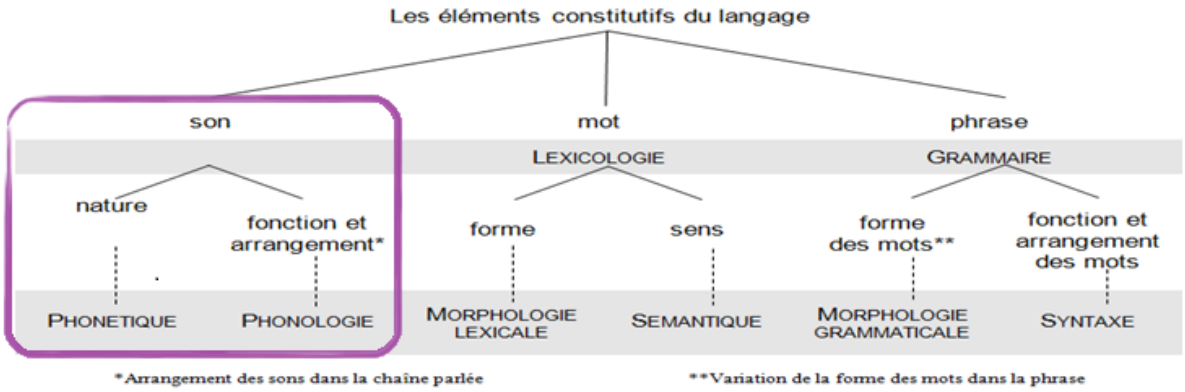

FIGURE 2.1 – Domaines de la linguistique selon la représentation statique des éléments constitutifs du langage [Martinet and Colin, 1967].

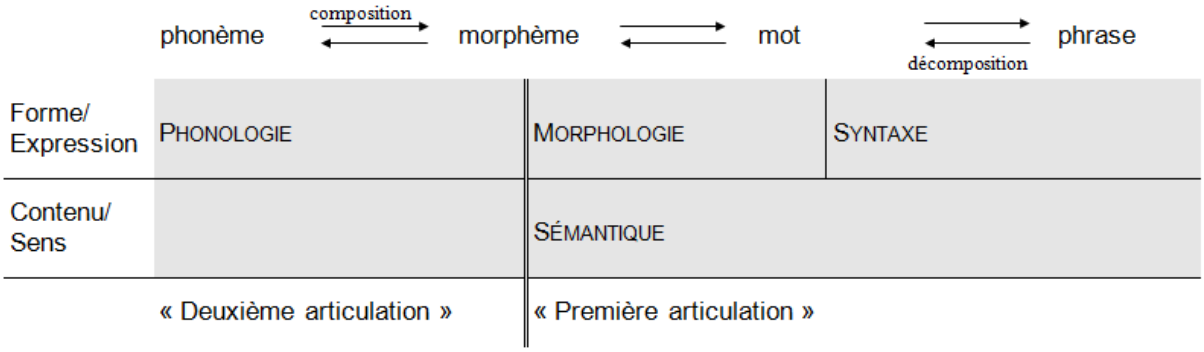

FIGURE 2.2 – Domaines de la linguistique selon la représentation dynamique des éléments constitutifs du langage [Martinet and Colin, 1967].

Au niveau de la *première articulation*, l'énoncé s'articule en unités douées de sens dont les plus petites sont les morphèmes. Au niveau de la *deuxième articulation*, chaque morphème s'articule en unités dépourvues de sens dont les plus petites sont les phonèmes. Dans une langue, chaque mot unité de première articulation est composé de la succession dans le temps d'unités minimales de deuxième articulation (les phonèmes).

terrain commun très important.

3 Phonologie

La phonologie[9] s'occupe de la fonction des sons dans la transmission d'un message. En d'autres termes, elle recherche les différences de prononciation qui correspondent à des différences de sens, ce qu'on appelle des oppositions distinctives. De ce fait, il faut comprendre une langue pour étudier sa phonologie. Il existe deux branches principales en phonologie :

— la *prosodie* (à base de syllabe) désigne les règles de prononciation globales qui influent sur la mélodie d'un énoncé. En français, suivant l'intonation qu'on y met, "tu viens demain" peut devenir une affirmation, un ordre ou une question sans que le sens des mots présents ne change pour autant.

— la *phonématique* (à base de phonèmes) comporte l'analyse de l'énoncé qui permet de dégager les phonèmes, leur description avec leurs variantes, leur classement et enfin l'étude de leurs combinaisons.

9. http ://post.queensu.ca/ lessardg/Cours/215/chap3.html, consulté en 16-1-2017

3.1 Phonème

Les éléments de l'alphabet qui constituent les unités de base de l'analyse du langage ne sont pas tous pertinents pour une langue donnée. En fait, chaque langue opère une sélection dans la liste des sons que la physiologie rend possibles. Parmi ces sons adoptés, elle opère un regroupement en classes d'équivalences. La plupart des langues naturelles sont composées à partir d'un nombre très limité de phonèmes (normalement inférieur à cinquante), voir les tableaux 2.1 et 2.2. Certaines langues possèdent plus de phonèmes que d'autres mais toutes ont en commun le fait qu'elles construisent leurs mots à partir d'un nombre limité de phonèmes : 36 pour l'arabe, 37 pour le français, 44 pour l'anglais, 48 pour l'allemand, 26 pour l'espagnol, etc.

Dans une approche structurale, l'identification des phonèmes se fait par la méthode de la paire minimale : deux sons élémentaires appartiennent à des classes différentes de phonèmes, s'il est possible de trouver deux unités lexicales différentes associées à des signifiés différents qui ne diffèrent d'un point de vue acoustique que par ces deux sons. Le fait de remplacer un son par un autre dans une paire minimale s'appelle la commutation. Si la commutation change le sens, nous tirons la conclusion que les deux sons appartiennent à deux classes distinctes. La substitution d'un phonème par un autre dans un mot peut faire changer ce mot soit vers une forme qui n'appartient pas au lexique soit vers un autre mot [Virole, 1999]. A titre d'exemple : en français, *père* et *mère* forment une paire minimale

I:	see	/sI:/
I	sit	/sIt/
e	ten	/ten/
&	hat	/h&t/
A:	arm	/a:m/
Q	got	/gQt/
O:	saw	/sO:/
U	put	/pUt/
u:	too	/tu:/
V	cup	/kVp/
3:	fur	/f3:(r)/
@	ago	/@"g@U/
eI	page	/peIdZ/
@U	home	/h@Um/
aI	five	/faIv/
aU	now	/naU/
OI	join	/dZOIn/
I@	near	/nI@(r)/
e@	hair	/he@(r)/
U@	pure	/pjU@(r)/

(a)

p	pen	/pen/
b	bad	/b&d/
t	tea	/ti:/
d	did	/dId/
k	cat	/k&t/
g	got	/gQt/
tS	chin	/tSIn/
dZ	June	/dZu:n/
f	fall	/fO:l/
v	voice	/vOIs/
T	thin	/TIn/
D	then	/Den/
s	so	/s@U/
z	zoo	/zu:/
S	she	/Si:/
Z	vision	/"vIZn/
h	how	/haU/
m	man	/m&n/
n	no	/n@U/
N	sing	/sIN/
l	leg	/leg/
r	red	/red/
j	yes	/jes/
w	wet	/wet/

(b)

TABLE 2.1 – Les 20 Voyelles & diphtongues et les 24 Consonnes de l'Anglais avec des mots exemples [Ficquet and Mbodj-Pouye, 2009].

qui permettent d'identifier /p/ et /m/ comme des phonèmes distincts. De même, pour *rouge* et *bouge* : /R/ et /b/ s'opposent entre eux puisque le seul élément qui change entre les deux mots est la consonne initiale.

Les phonèmes peuvent être classés en fonction de différents critères essentiels dit *traits distinctifs* : le mode articulatoire, le voisement, le lieu d'articulation, la labilité, etc. En fait, les phonèmes qui sont des éléments abstraits associés à des sons élémentaires ne sont

a	patte	/pat/
A	pâte	/pAt/
~A	clan	/kl~A/
e	dé	/de/
E	belle	/bEl/
~E	lin	/l~E/
@	demain	/d@m~E/
i	gris	/gli/
o	gros	/glo/
O	corps	/kOr/
~O	long	/l~O/
9	leur	/l9r/
~9	brun	/bl~9/
{	deux	/d{/
u	fou	/fu/
y	pur	/pyl/
j	fille	/fij/
H	huit	/Hit/
w	oui	/wi/

(a)

b	bal	/bal/
d	dent	/d~A/
f	foire	/fwar/
g	gomme	/gOm/
k	clé	/kle/
l	lien	/lj~E/
m	mer	/mE1/
n	nage	/naZ/
J	gnon	/J~O/
N	dancing	/d~AsiN/
p	porte	/pOrt/
r	rire	/1i1/
s	sang	/s~A/
S	chien	/Sj~E/
t	train	/tr~E/
v	voile	/vwal/
z	zèbre	/zEb1/
Z	jeune	/Z9n/

(b)

TABLE 2.2 – Les 19 Voyelles & Semi-voyelles et les 18 Consonnes du Français avec des mots exemples [Ficquet and Mbodj-Pouye, 2009].

pas identiques pour chaque langue : le /a/ du français (comme par exemple dans le mot *Paris*) n'est pas vraiment équivalent au /a/ de l'anglais (dans le mot *cat*). De ce fait est née l'idée de définir un Alphabet Phonétique International (API) [10] qui permet de décrire les sons et leurs prononciations de manière compacte et universelle (voir le tableau 2.3). Lorsqu'il n'est pas possible d'utiliser l'API pour des raisons techniques, il existe d'autres méthodes de transcription des corpus oraux comme le code SAMPA (*Speech Assessment Methods Phonetic Alphabet*), son code étendu XSAMPA, etc.

3.2 Allophone

Une difficulté majeure à affronter est celle de la segmentation du flux continu de paroles en unités discrètes. Une fois ce découpage est réalisé, identifier le phonème correspondant à chaque unité n'est pas évident. Un même phonème peut être prononcé de façons très différentes, suivant son voisinage avec les autres phonèmes. Par exemple, un /a/ en début ou en fin de mot ne se prononce pas de la même façon. On dit alors qu'un phonème peut avoir plusieurs allophones. Certains phonèmes ont tendance aussi à disparaître dans une prononciation courante.

Par exemple [11] en français, *père* peut être prononcer de trois façons différentes selon la consonne /r/qui sont interprétées comme des variantes de prononciation ou des allophones de l'unité phonologique. Le fait de rouler le /r/ (le produit avec le bout de la langue), de le grasseyer (le produit avec le dos de la langue dans la gorge) ou de le prononcer normalement ne provoquent pas de changement de sens malgré qu'ils sont bien différents du point de vue de la production : ils sont phonétiquement distincts et phonologiquement semblables.

10. https ://www.internationalphoneticassociation.org/, consulté en mars 2017
11. https ://fr.wikipedia.org/wiki/Allophone_%28phonologie%29, consulté en mars 2017

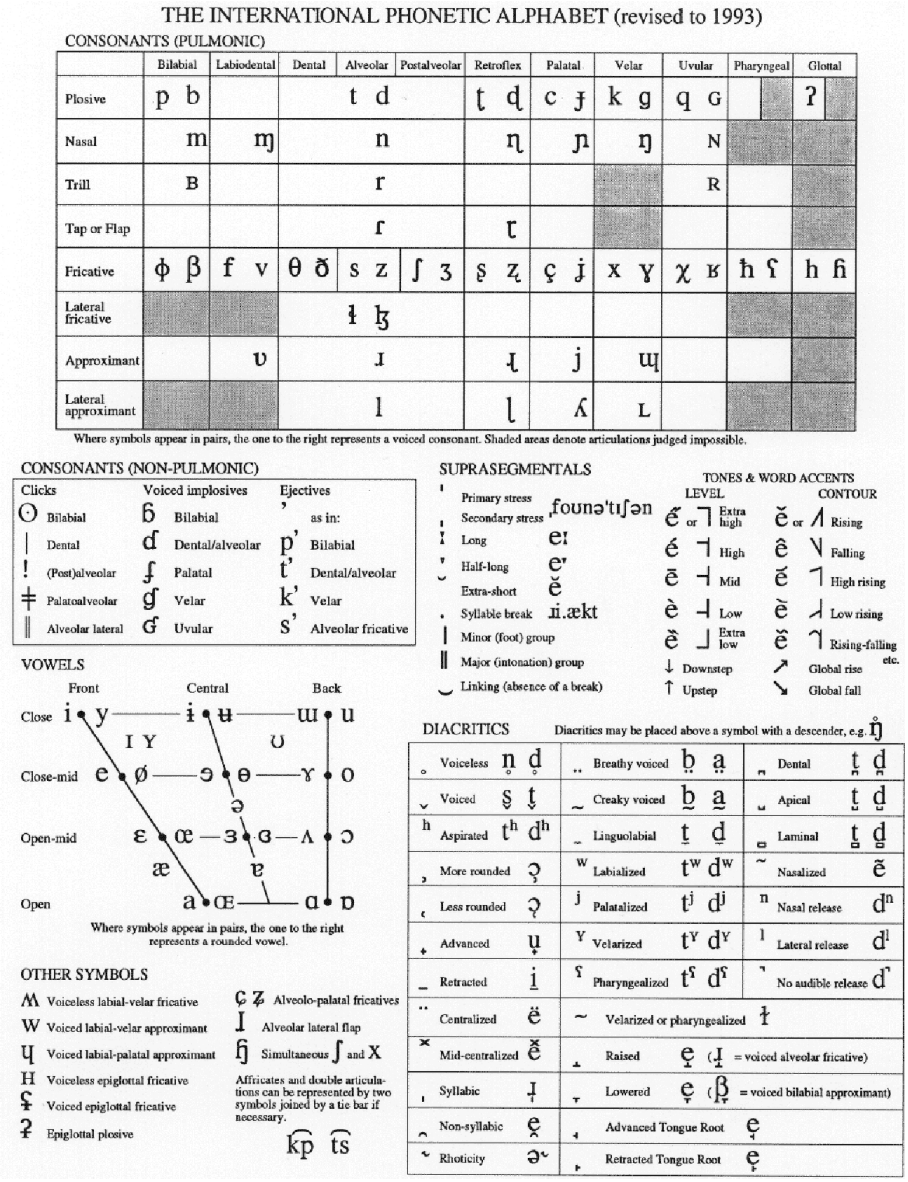

TABLE 2.3 – Tableau complet de l'alphabet phonétique international.

4 Phonétique

La phonétique s'occupe de l'expression linguistique et non pas du contenu. L'étude phonétique d'une langue peut se faire sans faire appel au sens. Il est possible d'étudier les caractéristiques phonétiques d'une langue qu'on ne comprend même pas. Aussi, la phonétique étudie la production des sons de parole, leur transmission sous forme d'ondes sonores ainsi que leur réception. Ce domaine d'étude des sons de la parole exclut les autres sons produits par les êtres humains, même s'ils servent parfois à communiquer comme les toux, les raclements de gorge, etc. Elle exclut aussi les sons non humains. On distingue trois branches principales correspondant au trois étapes de la communication (la production de la parole, sa transmission et sa perception [12]) :

12. Il existe d'autres branches de la phonétique dont on ne s'intéresse pas dans ce contexte :
— la phonétique comparative, entre différentes langues ;
— la phonétique didactique (apprentissage d'une langue étrangère) ;
— la phonétique historique (évolution des sons d'une langue au cours du temps) ;

— *Phonétique articulatoire* étudie les organes de la production des sons.
— *Phonétique acoustique* étudie les propriétés physiques des sons.
— *Phonétique auditive (ou perceptive)* étudie l'appareil auditif et le décodage des sons.

4.1 Phonétique articulatoire

La phonétique articulatoire s'intéresse à la production du son de la parole et s'occupe des activités des organes qui la rendent possible. Le mécanisme phonatoire se compose de deux composantes principales : (1) l'*appareil respiratoire* qui fournit l'énergie de départ sous forme d'un souffle d'air et (2) le *système phonatoire* qui est composé à son tour de plusieurs cavités, comme il est illustré dans la figure 2.3.

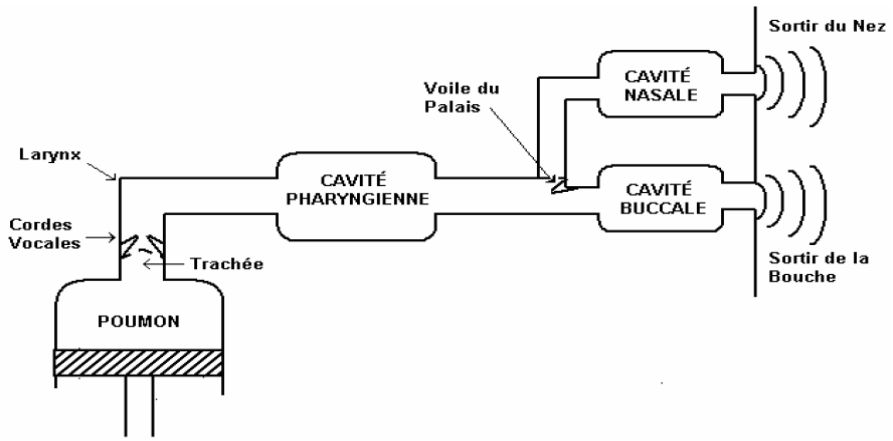

FIGURE 2.3 – Modèle mécanique de la production de la parole.

Ce schéma représente les différentes parties de l'appareil phonatoire : (1) les poumons (niveau respiratoire), (2) le larynx avec les cordes vocales (niveau phonatoire) (3) les cavités pharyngienne, nasale et buccale (niveau articulatoire). Dans la phonation, les poumons libèrent à un rythme qui est sous le contrôle volontaire du locuteur un souffle d'air qui passe par la trachée puis traverse le larynx, la cavité pharyngale (le pharynx) et sort par la bouche ou par le nez.

Système respiratoire

L'appareil respiratoire est constitué par les poumons et les voies aériennes supérieures. La fonction première de ce système est d'oxygéner le corps humain, toutefois il nécessaire pour la phonation. Lors de l'inspiration, l'action du diaphragme (qui se contracte et s'abaisse) et les muscles intercostaux permettent de vider les poumons et de les remplir d'air. Lors de l'expiration, le diaphragme se détend et laisse échapper l'air pénétré qui est ensuite utilisé pour produire des sons.

Système phonatoire

Le Système Phonatoire ou vocal (*vocal tract*) est composé d'une suite de cavités [Hézard, 2013] : le larynx, la cavité supra-glottique, la cavité pharyngale, la cavité nasale et la cavité orale (buccale), comme illustré dans la figure 2.4.

— la phonétique clinique (pathologie vs. norme), etc.
— la psychophonétique (perception métaphorique des sons) ;
— la sociophonétique (influence du milieu, de l'âge sur la façon de parler), etc.

Pendant le discours, les cordes vocales s'ouvrent et se ferment rapidement et font, pour ainsi dire, des sections dans la colonne d'air qui est divisée en bouffées successives. Ces bouffées en se succédant rapidement forment un bourdonnement audible dont la fréquence s'élève à mesure que s'accélèrent les vibrations des cordes vocales. Le caractère de ce bourdonnement est ensuite modifié par les propriétés acoustiques du conduit vocal. Ces propriétés dépendent de la forme que prend le conduit vocal en fonction du mouvement de la langue, des lèvres, de la mâchoire inférieure, etc. Ce sont les mouvements de ces résonateurs (voir figure 2.4) qui en modifiant les propriétés du conduit vocal permettent l'émission des différents sons intelligibles du langage. De ce fait, les phonèmes peuvent se

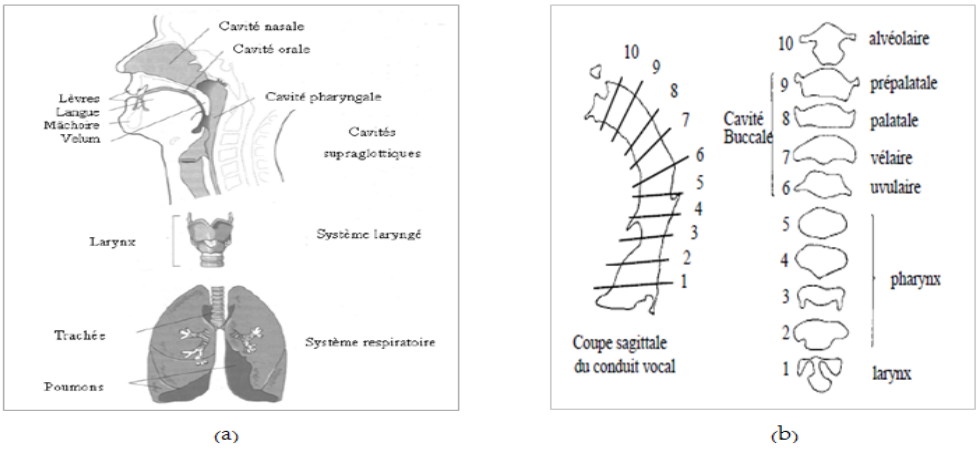

FIGURE 2.4 – Vue de l'appareil phonatoire.

L'appareil phonatoire est composé d'une suite de cavités : le larynx, la cavité supra-glottique, la cavité pharyngale, la cavité nasale et la cavité buccal. Les modifications rapides de la configuration des articulateurs dans le conduit vocal donnent lieu à différents sons de la parole.

regrouper en classes dont les éléments partagent des caractéristiques communes [LE Manh, 2007] relevant des *traits distinctifs*. Pour les voyelles, on peut les classer selon les traits distinctifs suivants :

- la *nasalité* : la voyelle est prononcée à l'aide du conduit vocal ou du conduit nasal suite à l'ouverture du velum ;
- le *degré d'ouverture du conduit vocal* ;
- le *lieu d'articulation* : la position de la constriction principale du conduit vocal réalisée entre la langue et le palais ;
- la *protrusion des lèvres* ou dite la labilité.

De même, les consonnes peuvent être classées à l'aide de trois traits distinctifs :

- le *voisement* : la consonne est prononcée avec une vibration des cordes vocales ;
- le *mode d'articulation* : tel que les modes occlusif, fricatif, nasal, glissant ou liquide ;
- le *lieu d'articulation* qui contrairement aux voyelles n'est pas nécessairement réalisé avec le corps de la langue.

4.2 Phonétique acoustique

La phonétique acoustique étudie la transmission du signal acoustique et examine ses caractéristiques en s'appuyant sur le traitement de signal [Meunier, 2007]. Cet apport de la physique acoustique permet un classement fin des sons en fonction de leur hauteur, leur intensité et leur timbre. Ces trois notions pouvent être traduites en variables physiques : la fréquence, l'amplitude de la vibration et l'audibilité des harmoniques.

La source vocale est créée par les vibrations des cordes vocales qui sont aéro-dynamiquement pilotées. Ces sources, qui peuvent être voisée ou/et non voisée, subissent des modifications spectrales selon la forme du conduit vocal qui agit comme un filtre acoustique. Chaque phonème est constitué par un faisceau de traits distinctifs acoustiques qui correspondent à des traits articulatoires pour décrire la production de la parole.

Du point de vue perceptif, la description en traits acoustiques présente plus de légitimité que les traits articulatoires, même si en pratique et par commodité de langage on désigne parfois les phonèmes par leurs traits articulatoires. Ainsi, il est plus en usage de parler d' *occlusive sourde dentale* que de phonème *discontinu, non voisé et aigu* alors qu'il s'agit du même phonème.

Modèle source-filtre pour le Production de la parole

Grâce aux travaux précurseurs de Chiba et Kajiyama (1941) et Fant (1960), suivis par une succession de contributions telles que : Stevens, 1971 ; Flanagan, 1972 ; Ishizaka et Flanagan, 1972 ; Alessandro, 2010) [Hézard, 2013], les aspects acoustiques de la parole sont devenus plus clairs. Un modèle standard et intéressant décrivant la structure du signal de parole est le modèle *source-filtre* proposé par Gunnar Fant et illustré par la figure 2.5.

Ce modèle permet une séparation de la production de la voix (source) et de sa mise en forme (filtre), il se compose d'un signal d'excitation qui modélise la source sonore $e(n)$ passant par le filtre $h(n)$ pour produire le son de la parole $s(n)$ [Alcaraz Meseguer, 2009] : La parole est un signal physique produit par la mise en mouvement d'un nombre consi-

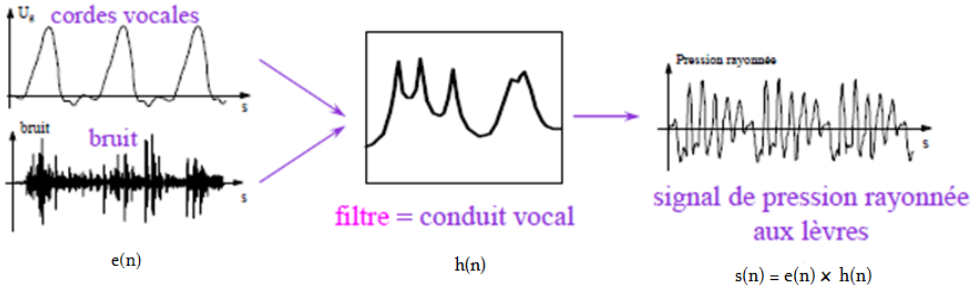

FIGURE 2.5 – Principe du modèle source –filtre de Fant.

D'un point de vue physique, l'appareil phonatoire est constitué d'un générateur sonore (qui permet production de trains d'impulsions ou du bruit dû à un écoulement d'air turbulent), d'un ensemble de cavités réglables (les résonateurs buccal, labial et nasal) et des articulateurs permettant la modification de ces cavités. L'ensemble est modélisé comme la réponse d'un filtre à un train d'impulsions de la source glottique, soit un modèle source-filtre.

dérable d'organes. Les modifications rapides de la configuration des articulateurs dans le conduit vocal donnent lieu à des frottements qui selon le modèle source-filtre produisent une onde. La répercussion de cette onde sur les diverses parois (buccales et nasales) du conduit vocal provoque l'amplification de certaines parties du spectre, ce qui équivaut à l'application d'un filtre. Lorsqu'un excitateur entre en vibration, il fournit un signal dont le résonateur va amplifier certaines composantes. On obtient alors des formants qui permettent la caractérisation du timbre. On appelle *formants*, les composantes fréquentielles de la voix qui subissent une amplification. Pour un conduit vocal dont la longueur est de l'ordre de 17 cm, on peut observer 3 ou 4 formants entre 100 et 5000 Hz. Les trois premiers formants sont indispensables pour caractériser le spectre vocal, les deux suivants sont utiles pour une synthèse de qualité [Boite, 2000]. Ces régions formantiques apparaissent très clairement sur les spectrogrammes.

Certains phonèmes peuvent correspondre à une amplification de la quasi-totalité du spectre perceptible par l'humain, c'est le cas de la plupart des consonnes. Les voyelles quant à elles sont caractérisées par une structure spectrale relativement stable dans le temps. Le signal acoustique résultant de ces phénomènes mécaniques se caractérise par une organisation spectro-temporelle complexe qui est transmise au système auditif périphérique par des phénomènes de transduction mécanico-électriques.

Les sons de la parole peuvent être présentés selon trois cas :

— *Silence* : pas de parole produite ;
— *Sons non voisés* : les cordes vocales ne vibrent pas, l'air passe librement à travers la glotte sans provoquer de vibration des cordes vocales ayant pour résultat la forme d'une onde aléatoire périodique de la parole ;
— *Sons voisés* : les cordes vocales sont tendues et vibrantes périodiquement ayant pour résultat une forme d'onde quasi-périodique (sur une courte période de 5-100 ms où le signal vocal est considéré stationnaire). L'excitation est une vibration périodique des cordes vocales suite à la pression exercée par l'air provenant de l'appareil respiratoire. Ce mouvement vibratoire correspond à une succession de cycles d'ouverture et de fermeture de la glotte. Le nombre de ces cycles par seconde correspond à la fréquence fondamentale (noté $F0$).

Rappelons que, le modèle source-filtre donne une idée du mécanisme acoustico-physique qui participe à la production de la parole, mais il est insuffisant pour décrire précisément un *être communiquant*. Les sons que l'homme produit ne sont pas n'importe quel type de sons : la parole s'articule autour d'unités de bases (les phonèmes) pour donner naissance à la langue. Ainsi, la production de la parole est le lieu d'un double codage, acoustique et syntaxico-sémantique [Obin, 2006].

4.3 Phonétique auditive

La phonétique auditive est une branche de la phonétique qui se préoccupe de la façon dont l'être humain entend, perçoit et reconnaît les sons de la parole par l'oreille. Elle examine les phénomènes de perception des sons du langage par les êtres humains, décrit l'appareil auditif et le décodage des sons.

La parole comme phénomène acoustique se déploie dans des limites énergétiques, spectrale et temporelles compatibles avec les capacités des capteurs cochléaires. L'intensité moyenne de la parole est aux alentours de 60 dB [13] avec une variation de plus ou moins 15 dB selon les circonstances de l'énonciation. Les plages fréquentielles entre 500 et 4000 Hz sont prédéterminées par les capacités sélectives optimales de l'oreille. Il s'agit là de la parole comme objet purement acoustique. La particularité et le grand mystère du langage est que sur cet objet physique sont extraits par des processus encore mal compris, des éléments catégorisateurs, nommés indices phonétiques qui permettent de transformer le flux physique du signal de parole en un flux phonologique permettant la génération du sens et donc l'intelligibilité de la parole [Virole, 1999].

Propriétés de l'ouïe

La capacité des humains et des animaux à répondre de manière adaptée aux stimuli de leur environnement dépend étroitement de la perception des signaux acoustiques. Notre oreille est sensible aux vibrations entre 16 Hz et 20000 Hz. En dessous de 16 Hz ce

13. Le décibel (dB) ne mesure pas des grandeurs, mais des rapports entre des grandeurs de même nature : pressions ou puissances acoustiques, etc.

sont des infra-sons que nous pouvons percevoir par la paroi abdominale. Au-dessus de 20000 Hz, il s'agit d'ultra-sons que certains animaux perçoivent (les chiens, les dauphins, les chauves-souris, etc.). Notre récepteur auditif délicat n'est pas linéaire en fréquence, ni en sensibilité [Besson, 2004].

L'oreille découpe son spectre en bandes critiques. Elle examine le niveau de chaque bande et le compare par rapport au seuil d'audition. Si le niveau est supérieur au seuil, la bande est audible et un certain stimulus est transmis au cerveau. Dans le cas contraire, la bande est inaudible et est dite masquée [Boite, 2000]. Les courbes classiques de Fletcher et Munson présentées dans la figure 2.6 donnent la sensibilité moyenne de l'oreille en fonction de la fréquence.

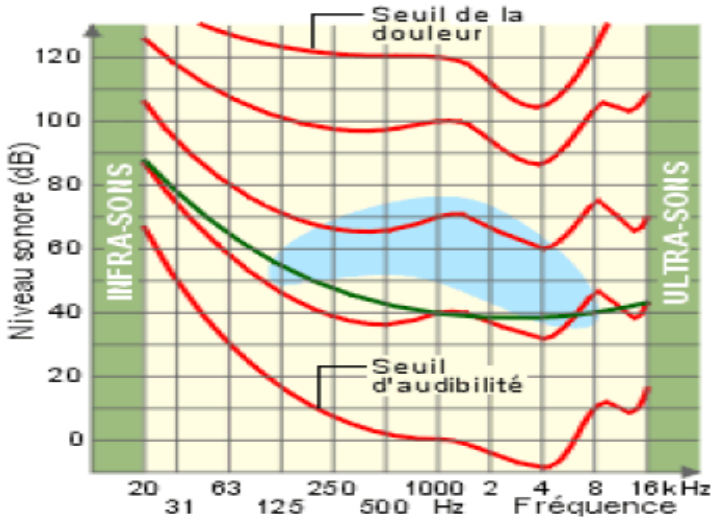

FIGURE 2.6 – Courbes d'égale sensation sonore d'après Fletcher et Munson [Mercier, 2010].

Les courbes de Fletcher et Munson montrent que l'oreille n'a pas une sensibilité linéaire mais que la sensation varie comme le logarithme de l'excitation. Afin de doubler la sensation acoustique, il faut multiplier par dix la puissance de la source pour multiplier par dix l'intensité acoustique en Pa (Pa : l'unité de pression est le newton par mètre carré (N/m^2), nommée en Europe le pascal (Pa).). La figure présente également les courbes d'iso-pression acoustique en fonction de la fréquence tout les 20 dB.

Système auditif humain

La partie visible de l'oreille qui représente le pavillon de l'oreille externe ne constitue qu'une infime partie de notre système auditif. Celui-ci se décompose en deux sous-ensembles : le système auditif périphérique et le système auditif central [Grognuz, 2012].

1. Le **système auditif périphérique** s'étend du pavillon pour s'arrêter aux premiers neurones du nerf auditif, responsable du codage de l'information sonore en potentiels électriques. Il est composé de trois parties qui sont l'oreille externe, l'oreille moyenne et l'oreille interne comme le montre le schéma de la figure 2.7. L'*oreille externe* comprend le pavillon, le conduit auditif et le tympan. Son rôle est de rassembler le son, de l'amplifier (+ 20 dB) et de le transmettre à l'oreille moyenne. L'*oreille moyenne* comprend trois petits os (marteau, enclume et étrier), la fenêtre ovale et la fenêtre ronde. Elle est remplie d'air. Son rôle est d'amplifier le son (+ 33 dB) et de le transmettre à l'oreille interne. La chaîne ossiculaire exerce une fonction de levier. La pression exercée sur le tympan par l'onde sonore est

reportée sur une membrane plus petite dite la fenêtre ovale. Le changement d'impédance se fait entre l'oreille moyenne et l'oreille interne. L'amplification a pour but d'adapter le niveau de pression sonore à la différence d'impédance. L'*oreille interne* comprend le centre de l'équilibre et la cochlée. Elle est remplie de liquide et contient les cellules sensorielles. Son rôle est la transmission hydromécanique au niveau de la membrane basilaire ainsi que la transmission électro-chimique au niveau des cellules ciliées de l'organe de Corti. Les cellules ciliées de l'organe de Corti se divisent en deux parties. On distingue les cellules ciliées internes et les cellules ciliées externes, appelées ainsi en raison de leurs positions respectives au sein de l'organe de Corti. C'est depuis les cellules ciliées internes, excitées par les mouvements de la membrane basilaire, que part l'information afférente. L'information efférente envoyée en retour depuis le tronc cérébral est réceptionnée par les cellules ciliées externes qui jouent majoritairement un rôle d'amplification.

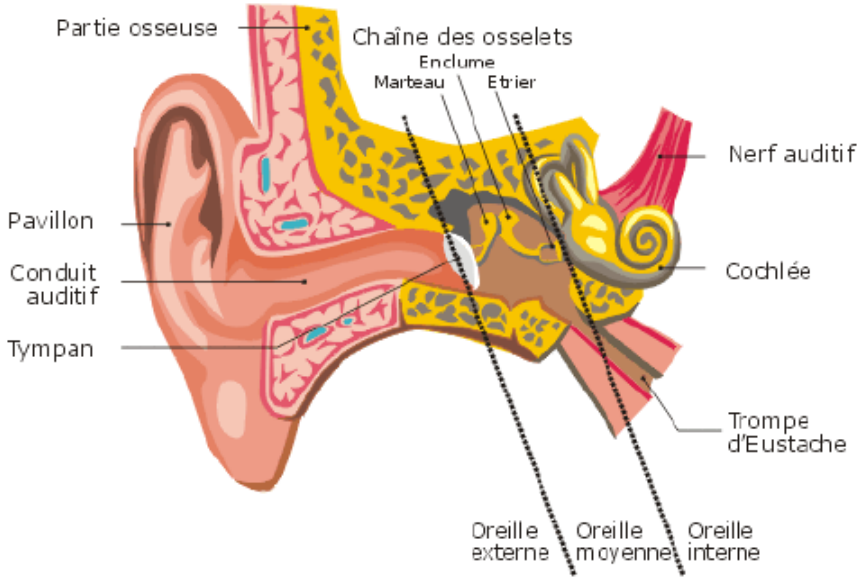

FIGURE 2.7 – Anatomie de l'oreille humaine.

Chacun de ces trois sous-systèmes (oreille externe, moyenne et interne) assure une étape dans la transmission du signal. L'oreille externe est responsable de la transmission aérienne à travers du conduit auditif externe. L'oreille moyenne assure au moyen des trois osselets le marteau, l'enclume et l'étrier, la transmission mécanique du tympan jusqu'à la fenêtre ovale. Tandis que l'oreille interne permet la transduction d'un signal mécanique en un signal bioélectrique qui sera transmis au cerveau via le nerf auditif.

2. Le **système auditif central** s'étend des premiers neurones du nerf auditif pour aller jusqu'au cerveau, responsable de l'interprétation de l'information sonore. Les 4000 cellules ciliées sensorielles internes contenues dans la cochlée sont reliées à 45000 fibres nerveuses. Les 30000 cellules ciliées externes sont, elles, reliées à 5000 fibres nerveuses. Cet ensemble se regroupe pour former les deux nerfs auditifs (correspondant à la VIIIe paire de nerfs crâniens) qui envoient l'information au cerveau, notamment dans le cortex auditif du lobe temporal [14].

14. Chez l'Homme, le lobe temporal est une zone importante du cerveau pour de nombreuses fonctions cognitives, dont notamment l'audition, le langage, la mémoire et la vision des formes complexes.

5 Outils d'analyse de la parole

Le développement de la chirurgie médicale et l'introduction de différents dispositifs électro-acoustiques ont rendue service à la phonétique et à son développement. Pour servir la phonétique acoustique différents outils sont utilisés dans les laboratoires de parole tels que [15] (Munot, 2002) : l'oscilloscope, le spectrographe IBM PC AT, etc. Pour la phonétique auditive des outils de synthèse de la parole comme : *Speech synthesizer*, *concatenated Speech synthesizer*, etc. [Alghamdi, 2000] sont utiles. Un état de l'art sur les plus importantes techniques de l'analyse de la parole sont exposées dans ce qui suit :

1. **Kymographie** :
   La kymographie est parmi les ancêtres des méthodes phonétiques acoustiques. Le kymographe permet de dessiner sur un papier (préalablement noirci à la fumée) la trace d'une onde sonore, fournissant des indications sur la segmentation de l'énoncé en éléments phonétiques, le mode articulatoire, la sonorité, la nasalité, la durée des articulations et l'influence du voisinage. Cet appareil proposé par *Abbé pierre-jean Rousselot*, *Antoine Grégoire* et *Maurice Gramment* est souvent gardé dans les laboratoires de phonétiques à des fins pédagogiques. Il est actuellement dépassé, à cause du frottement l'inertie de l'appareillage électromécanique qui donne à l'ensemble une imprécision non négligeable. D'ailleurs, le kymographe travaille dans une gamme de fréquences qui ne dépasse guère 500Hz (les sons de fréquences élevées n'apparaissent pas sur les tracés). Aussi, il présente l'inconvénient de mettre mal à l'aise l'utilisateur : un embouchoir avec une pastille de micro sur la bouche, un micro de contact placé sur la pomme d'Adam, une olive de verre avec un micro placée dans une narine, etc.
2. **Palatographie**, **Radio-cinématographie** et **Scanners** :
   La méthode kymographique a été complétée par la *palatographie*. L'idée de recueillir l'empreinte laissée par la langue sur le palais au cours de l'articulation qui est le but de la palatographie est très ancienne. Il y a peu de temps, on obtenait les palatogrammes à l'aide d'un palais artificiel placé dans la bouche du locuteur. Après que celui-ci a prononcé le son ou le groupe de sons voulus, on éloigne le palais artificiel et on peut déterminer immédiatement les parties qui ont été touchées par la langue. Cette méthode qui présentait certains inconvénients (problèmes de conservation des traces et gênes pour le locuteur) est aujourd'hui remplacée par un procédé photographique.

   L'appareil à photographier le palais dite *palatographe* fonctionne grâce à un jeu de quatre miroirs, dont l'un est introduit dans la bouche le au-dessous du palais et les trois autres projetant la lumière (d'une source lumineuse réglable) dans la cavité buccale. On photographie avec l'appareil fixé en face du miroir introduit dans la bouche, la trace est laissée après l'articulation par la langue badigeonnée avec du chocolat liquide. Les palatogrammes obtenus renseignent sur les faits de face et vus d'en bas : le mode, le lieu d'articulation, le comportement des bords latéraux de la langue, la largeur du contact lingual et le degré d'aperture des sons.

   La *palatographie* est une méthode précieuse mais incomplète, elle est enrichie ou remplacée par la radiographie et la radio cinématographie. La réalisation des radiocinématigrammes ou radiofilms nécessite l'utilisation des rayons X, ce qui permet de visualiser les mouvements des lèvres, des mâchoires, de la langue, le voile de palais, etc. au cours de la phonation. Malheureusement, cette méthode d'investigation

15. https ://books.google.com/booksisbn=287130114X, consulté en 12 Décembre 2015.

a deux reproches principales : le danger pour le sujet s'il doit être exposé longtemps ou fréquemment puisqu'il faut donner aux rayons une intensité considérable et le coût très élevé de l'installation. Mais les progrès techniques permettent désormais de recourir pour visualiser la production de sons en direct à des méthodes plus légères et plus sûres, grâce aux principes de l'échographie.

Les *scanners* médicaux (ou scanners, ou encore tomodensitomètres) autorise une visualisation précise au millimètre près des articulations en cours. Les documents obtenus peuvent être conservés et mesurés aisément. Parmi les inconvénients de cette méthode, on peut citer son coût lié à la valeur de l'appareil et l'intervention nécessaire des spécialistes.

3. **Oscillographe** :
   L'oscillographe qui est un outil électronique déjà utilisé dans divers domaines scientifiques et industriels a été depuis 1930 utilisé pour des besoins phonétiques. Les résultats obtenus sont précis et permettent une étude détaillée de certains aspects acoustiques des sons d'un langage (tels que la durée, la hauteur et l'intensité). L'installation oscillographique originale comporte un oscilloscope et une caméra spéciale. L'oscilloscope transmit les vibrations d'un microphone ou d'une autre source sonore à travers un amplificateur muni d'un tube à gaz raréfié. Les vibrations électroniques synchrones avec des vibrations de la voix se déplacent sur un cardan au fond du tube, sous forme de taches lumineuses (ou spots). L'inertie étant nulle, le spot peut se déplacer à la vitesse de toutes les impulsions par le microphone. La caméra à déroulement continu dans le temps permet de filmer à des vitesses variables (20 cm à 10 m à la seconde) le spot en mouvement et d'en saisir les courbes successives.

   Malgré que l'oscillographie a donné lieu à de nombreux appareillages plus sophistiqués, le spectrographe qui est plus facile à régler, plus souple et comprend lui-même des programmes d'analyse de nature oscillographique tend à les remplacer.

4. **Spectrographie** :
   Cependant, l'oscilloscope se prêtait assez mal à l'étude de la structure acoustique des sons, et leurs décompositions en fondamental et en harmoniques. C'est alors que, pendant la seconde guerre mondiale, fut inventée par la compagnie de téléphones Bell, aux États-Unis, un remarquable instrument d'analyse acoustique : le *spectrographe*. En 1948, cet appareil fut commercialisé par une firme américaine *Kay Electric Company* sous le nom de *Sona-Graph*.

   A l'origine comme son nom l'indique, le spectrographe (*Sona-Graph*) récupère l'onde sonore complexe survenue d'un microphone et la traite par plusieurs filtres (ou un filtre variable) qui analyse sa composition en harmonique pour enfin obtenir le spectre acoustique des sons, sur un feuille de papier spécial sensible à l'étincelle électrique. Un sonagraphe est essentiellement un filtre passe-bande, le choix de la fréquence de filtrage joue un rôle fondamental. Ce dernier est en trois dimensions : il renseigne sur la fréquence des divers harmoniques (en ordonnée, de bas en haut), sur l'intensité relative des divers harmoniques (degré de foncé) et sur la durée (en abscisses, de gauche à droite).

   L'interprétation du spectrogramme (voir un exemple dans la figure 2.8) permet donc de déterminer : le fondamental, les harmoniques et les formants (qui sont des zones de fréquences, réparties sur plusieurs harmoniques qui forment l'identité même du son, particulièrement les voyelles). Même si ces appareils nous rendent un grand service pour le traitement de la parole, la présence d'un expert ou un

phonéticien pour l'interprétation des tracés est importante.

L'analyse sonagraphique dont les fondements sont posés aux Etats-Unis à la fin des années trente était spécialement dédiée à l'étude de la parole. Cette technique été repris en France, par *A. Moles* (1966) et *Emile Leipp* (1971) et été appliquée à l'étude du phénomène musical.

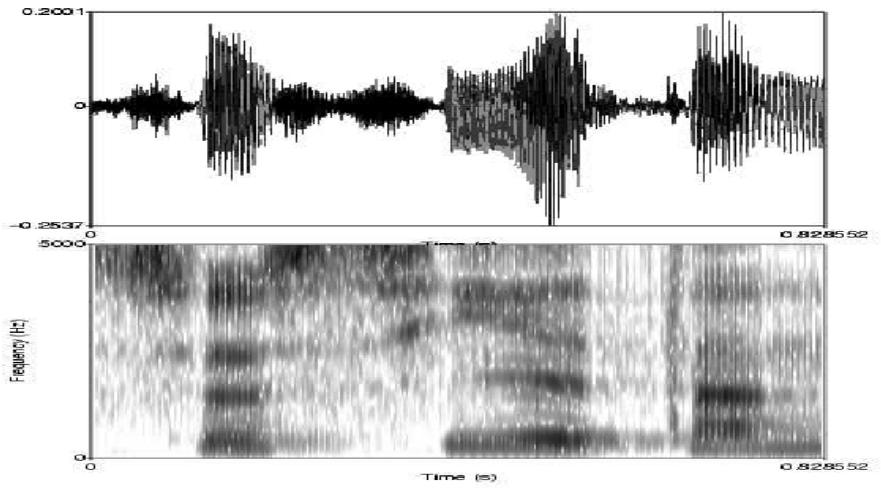

FIGURE 2.8 – Oscillogramme - en haut - et Spectrogramme - en bas - d'une onde acoustique correspondant à la prononciation d'un morceau de phrase.

Dans ces deux types enregistrements, l'axe horizontal marque le déroulement du temps. L'oscillogramme traduit par le mouvement d'une aiguille les variations de l'onde acoustique, tandis que le spectrogramme enregistre pour chaque fréquence sonore l'amplitude (en décibels) de cette fréquence et la traduit dans l'intensité du pixel correspondant. La phonétique acoustique étudie les propriétés de ce genre de diagrammes.

Selon l'analyse mathématique d'un son complexe par le théorème de Fourier, toute courbe complexe est la somme de courbes simples. Le spectrogramme représente le module de la transformée de Fourier appliquée sur le signal sonore initialement numérisé, avec les fréquences en ordonnée, le temps en abscisse et l'amplitude (ou l'énergie) en niveau de gris. Ainsi une zone sombre, indique une forte énergie à la fréquence et au temps correspondants. La répartition des formants sur le spectre, crée une véritable cartographie du son, qui va permettre de caractériser le timbre et/ou la couleur d'un son. On peut privilégier dans cette représentation, comme illustré dans la figure 2.9 [Cours, 2013] soit :

— un spectrogramme à bandes étroites (*narrow band*) de $10 - 45\ Hz$ qui offre une bonne résolution au niveau fréquentiel sur l'axe vertical mais l'analyse temporelle est moins fine ;
— ou un spectrogramme à bandes larges (*wide band)* de $150 - 300\ Hz$[16] qui offre une meilleure résolution temporelle sur l'axe horizontal et permet de dégager les formants vocaliques mais apporte moins d'éléments pour l'étude du domaine fréquentiel.

5. **Synthétiseur à formants** :
Pour contrôler les hypothèses faites à partir de l'analyse des formants, on a recourt à la synthèse de la parole. A partir des spectres acoustiques obtenus au moyen du spectrographe, on réalise par un procédé inverse des sons artificiels (langage

16. Hertz : le nombre de vibrations des cordes vocales (ouverture-fermeture de la glotte) en une seconde constitue la fréquence laryngée, qui s'exprime en hertz (Hz) tel que 1 hertz = 1 vibration par seconde.

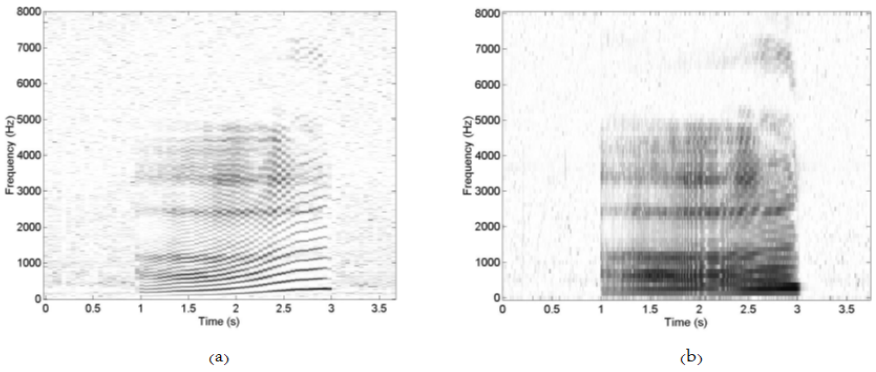

FIGURE 2.9 – Spectrogramme bande étroite (a) et spectrogramme bande large (b) d'une voyelle \a\ prononcée avec une fréquence fondamentale augmentant avec le temps.

Les harmoniques sont clairement identifiées sur le spectrogramme à bande étroite. Les formants sont plus particulièrement visibles sur les spectrogrammes à large bande. Ils sont matérialisés par des zones sombres indiquant des zones fréquentielles à forte énergie.

synthétique). On fait passer un spectre dessiné à base des connaissances qu'on a sur les sons de la parole dans une machine spéciale dite *synthétiseur de la parole (ou des formants)* où le spectre est lu et traduit par des circuits électriques qui émettent des résonances analogues à celles que produisent les cavités de l'appareil phonatoire humain.

6. **Programmes informatiques pour l'analyse** :
   Il existe différents logiciels paramétrables qui permettent l'enregistrement et le traitement de la parole tels que VisArtico de LORIA [17], Audacity [18], Transcriber, Praat, etc. La liste présentée n'est pas exhaustive mais permet de citer les quelques plus répandus dans la littérature.

   Transcriber : est logiciel pour l'analyse de la parole, dont son développement [Barras et al., 1998] [19] a été financé par la DGA (Délégation Générale pour l'Armement) et LDC [20] (*Linguistic Data Consortium*). Ce logiciel a été utilisé pour transcrire de nombreux corpus du catalogue LDC. Transcriber est un logiciel optimisé pour la transcription et l'annotation de corpus volumineux. Il propose quatre niveaux d'annotation (texte, locuteur, thème et bruits de fond). Il offre une gestion des locuteurs permettant d'indiquer, en plus de leur identité, des informations telles que leur sexe, le degré de spontanéité (parole préparée ou spontanée), le canal d'expression, la qualité de l'enregistrement, etc. Par ailleurs, un nombre important de balises est intégré pour représenter les événements sonores (bruit, respiration, toux, reniflement, etc), les prononciations particulières ou encore des particularités lexicales. Aussi, l'utilisateur peut créer ses propres balises. Il offre la possibilité d'aligner le signal audio avec la transcription, et de présenter un affichage permettant d'accéder directement aux segments du fichier son que l'on souhaite écouter. Ce programme est gratuit, open source, ergonomique, simple d'accès et capable de traiter de nombreux formats, en entrée comme en sortie. Il peut gérer des fichiers audio de plusieurs heures.

17. http ://visartico.loria.fr/
18. https ://www.audacityteam.org
19. http ://trans.sourceforge.net
20. https ://www.ldc.upenn.edu/

Praat [21] : est un logiciel d'analyse de la parole, distribué gratuitement, développé à l'Institut des Sciences Phonétiques de l'Université d'Amsterdam (Pays-Bas) par *Paul Boersma* [Boersma and van Heuven, 2001] et *David Weenink*. Ce programme fonctionne sur plusieurs plates-formes et offre la possibilité d'enregistrer des fichiers audio qui peuvent être ensuite analysés sous Praat. Ces fichiers peuvent être codés selon une multitude de formats audio. Praat permet de segmenter et de transcrire des fichiers audio, d'effectuer des analyses phonétiques et acoustiques, de manipuler le signal sonore, etc. Le logiciel permet de réaliser des traitements prosodiques standard et d'éditer des annotations indépendantes. Pour une même zone de la transcription, il est donc possible de disposer d'une annotation orthographique, d'une annotation phonétique, etc. Cela permet également de gérer la parole superposée, en créant des annotations par locuteur.

XTrans : est un outil de transcription de la nouvelle génération de LDC. Comme son prédécesseur Transcriber, il est distribué gratuitement [22]. *XTrans* intègre un système permettant de transcrire la parole superposée. Il est possible d'avoir plusieurs locuteurs qui parlent sur un même canal, ou bien d'avoir différents canaux pour représenter les différents locuteurs. Le logiciel utilise le concept de *Virtual Speaker Channels* (VSC). Chaque VSC correspond à un locuteur. Plusieurs VSC peuvent être associés à un seul canal audio si plusieurs locuteurs sont enregistrés sur ce canal. Un VSC peut être associé à plusieurs canaux audio dans un même fichier son ou à plusieurs fichiers sons (un locuteur - un canal, un locuteur -plusieurs canaux, plusieurs locuteurs - plusieurs canaux). *XTrans* offre également la possibilité de transcrire du texte dont la lecture se fait de la droite vers la gauche, comme l'arabe ou l'hébreu. Le logiciel inclut aussi des fonctions d'annotations dans l'outil de transcription.

VisArtico : est un outil de LORIA Nancy, destiné à visualiser les données articulatoires acquises à l'aide d'un articulographe. Il est destiné aux chercheurs qui ont besoin de visualiser les données acquises à partir de l'articulographe sans traitement excessif. VisArtico est bien adapté aux données acquises à l'aide des AG500 et AG501, développées par Carstens (Carstens Medizinelektronik GmbH). Depuis la version 0.8, VisArtico prend en charge le format NDI Wave, développé par Northern Digital Inc. Dans la dernière version, il est possible d'ouvrir le format adhoc, ce qui permet d'afficher les données de capture de mouvement de base. VisArtico est un logiciel multiplateforme. Il a été testé et travaillé sur Windows, Mac OS et Linux. C'est un logiciel basé sur Java (assurez-vous que Java est installé avant d'utiliser VisArtico).

WaveSurfer : est un logiciel *Open* Source pour la visualisation et la manipulation du son, conçu pour convenir à la fois aux utilisateurs novices et avancés. WaveSurfer a une interface utilisateur simple et logique qui fournit des fonctionnalités d'une façon intuitive et qui peuvent être adaptées aux différentes tâches. WaveSurfer peut être utilisé comme un outil autonome pour une large gamme de tâches dans la recherche de la reconnaissance automatique de la parole et l'éducation. Il peut également servir de plateforme pour des applications plus avancées et

21. http ://www.praat.org/
22. http ://www.ldc.upenn.edu/tools/XTrans/downloads/, consulté en mars 2017.

spécialisés. Ceci est accompli soit par extension de l'application WaveSurfer avec la coutume de nouveaux plug-ins [Salvi and Vanhainen, 2014] ou en intégrant des composants de visualisation WaveSurfer dans d'autres applications.

6 Conclusion

Les langues naturelles sont avant tout orales, beaucoup d'eux n'ont pas de transcription écrite. Il est donc naturel que de nombreuses propriétés de ces langues découlent de considérations acoustiques.

Dans ce chapitre nous avons présenté la linguistique qui est une science ancienne et qui étudie les langages naturels. La linguistique moderne regroupe un certain nombre de domaines. La phonétique étudie la prononciation réelle alors que la phonologie étudie le système qui sous-entend cette prononciation. La phonétique englobe trois principales disciplines relatives à l'articulation, l'acoustique et l'audition. La phonétique articulatoire étudie les organes de la production des sons. La phonétique acoustique à son tour étudie les propriétés physiques des sons. Enfin, la phonétique auditive (ou perceptive) étudie l'appareil auditif et le décodage des sons. La reconnaissance automatique de la parole qui est sujet du prochain chapitre se base sur tout ces concepts divers.

Chapitre 3

Reconnaissance Automatique de la Parole

1 Introduction

La Reconnaissance Automatique de la Parole (RAP) vise à convertir le signal acoustique en texte. La variabilité phonétique observée dans le signal acoustique devrait pouvoir être traitée et atténuée afin de présenter la suite de mots prononcée par une prononciation standard. Le système de la reconnaissance automatique de la parole permet de traiter et de réaliser cette tâche : transcrire la parole humaine en texte écrit.

Cette problématique s'apparente au moins partiellement à celle des premiers phonologues de la fin du XIXe et au début du XXe siècle, comme Saussure ou Troubetzkoy. Ils visaient à distinguer deux « phonétiques » descriptives distinctes suivant qu'on veut étudier les sons phoniques comme des signaux physiques (phonétique) ou comme des éléments abstraits de sons distinctifs d'un système linguistique (phonologie).

Donc quels sont les problèmes posés par la parole en tant que signal physique continu (le signifiant acoustique) avant de pouvoir être transformé en signe linguistique graphémique discret (ou signifiant écrit, d'après la terminologie de Saussure) ? Il est communément admis que la parole est très variable et les différents facteurs responsables peuvent être résumés et cités :

- De manière générale, il n'y a pas de frontières détectables entre les mots et les frontières de phonèmes. La réalisation acoustique d'un phonème dépend fortement de son contexte phonémique gauche-droite.
- Le signal de parole varie en fonction du locuteur (sexe, âge, émotions, accent, etc.).
- Les conditions d'enregistrement et le bruit de fond se superposent à la parole lors de l'enregistrement.
- Le style de parole (lue, préparée, spontanée, etc.) influe fortement sur le débit, la prosodie, la précision de l'articulation, les variantes de prononciations, etc. De même la situation (parole publique ou privée ; monologue ou dialogue, familier ou formel) et l'information portée par le contexte jouent un rôle important sur le choix des mots et leur prononciation. Le contexte général et le sujet abordé influent surtout sur le vocabulaire utilisé [Adda-Decker, 2007] dans le système de reconnaissance.

2 Contexte historique

La naissance de la reconnaissance automatique de la parole (RAP) comme domaine scientifique remonte aux années quarante ou même aux années trente [Rabiner and Juang, 2008]. C'est dans cette période au USA, que les premières tentatives de création d'une machine capable de comprendre le discours humain a eu lieu. Leurs principaux objectifs étaient d'interpréter les messages russes interceptés. Les premiers systèmes de la RAP étaient très rudimentaires et reconnaissaient juste quelques sons prononcés de façon isolée. Ils utilisaient les moyens de l'électronique analogique de l'époque et se fondaient sur le traitement du signal exclusivement.

L'apparition des premiers ordinateurs dans les années soixante a permis de numériser le signal, d'automatiser le processus de reconnaissance et de traiter des problèmes plus complexes : plus grand nombre de sons ou des mots à reconnaître, plus grand nombre de locuteurs pris en compte, etc. A cette époque, l'approche expert et à base de connaissances de l'Intelligence Artificielle (IA) étaient largement utilisées.

Dans les années quatre-vingts, après un passage par la programmation dynamique, l'utilisation de l'approche stochastique devient de plus en plus prépondérante dans la reconnaissance. Cela est motivé en grande partie par la possibilité d'automatiser complètement le processus de reconnaissance. De plus l'approche probabiliste est bien formalisée et justifiée mathématiquement grâce à différentes méthodes d'apprentissage de modèles, preuves de convergence, etc. Cette approche a été utilisée tant au niveau acoustique qu'au niveau de la modélisation du langage. Malgré une charge de calcul importante, elle a permis de passer à la reconnaissance de la parole continue, c'est-à-dire à la parole prononcée sans pauses entre les mots. Cette approche reste jusqu'à nos jours la plus utilisée.

Le passage de l'approche à base de connaissance vers l'approche stochastique a été effectuée de façon assez radicale : les systèmes experts ont été remplacés radicalement par des modèles stochastiques à base de modèles de Markov cachés à tous les niveaux du processus de reconnaissance. Actuellement, l'approche probabiliste est considérée comme mûre et elle représente l'état de l'art du domaine mais quelques limites sont signalées. Un retour flagrant vers l'approche connexionniste est dans l'horizon, ce sujet est traité partialement dans le chapitre 4.

Ces dernières années, les principaux progrès dans le domaine de la RAP peuvent être classés en trois catégories :

1. La modélisation est devenue très détaillée au niveau acoustique et au niveau du modèle de langage. Une quantité importante de logiciels libres sont disponibles pour mettre en place les mélanges de gaussiennes pour les modèles acoustiques et les n-grammes pour les modèles de langage. Des bases de données de plus en plus volumineuses sont disponibles pour apprendre ces modèles.
2. La modélisation est devenue *adaptative*. Plusieures méthodes d'adaptation au bruit et au locuteur ont été proposées. Ces méthodes sont efficaces même pour l'adaptation incrémentale et avec une petite quantité de données d'adaptation.
3. Pour apprendre les différents modèles, différentes méthodes d'apprentissage et de *modélisation discriminante* sont souvent utilisées. Elles permettent d'améliorer la séparabilité entre les modèles et donc d'augmenter la performance des systèmes de la RAP.

Les meilleurs systèmes de la RAP testés durant des compagnes d'évaluations sont capables de reconnaître la parole radiophonique et télévisée avec un taux d'erreur de l'ordre de 12% à 27% [Galliano et al., 2009] et les résultats sont arrivés même à un taux de 10.9%

[Galibert et al., 2014]. Ce genre de transcription de parole est difficile puisqu'il s'agit de traiter un très grand vocabulaire (plusieurs centaines de milliers de mots), des phrases qui ne sont pas toujours grammaticalement correctes et de paroles prononcées parfois sur fond musical, avec des bruits ou par téléphone ou pendant des réunions. Le système de la RAP d'IBM Watson de Tom Sercu et ses compagnons a atteint un taux d'erreur de 6.9% avec des modèles acoustiques et linguistiques avancés. Dans les grands laboratoires, le but ultime est d'atteindre ou de dépasser la précision humaine qui est estimée à un taux d'erreur d'environ 4%, sur cette tâche.

3 Formulation probabiliste d'un système de RAP

Actuellement, la recherche dans le domaine de la RAP concerne la reconnaissance de parole continue spontanée à grand vocabulaire (*Large Vocabulary Continous Speech Recognition* - LVCSR) avec des dizaines ou des centaines de milliers d'entrées.

Considérant le signal acoustique prononcé par un utilisateur comme une observation notée O et W est la séquence de mots correspondante. Le système de la RAP recherche la séquence de mots la plus vraisemblable par rapport au signal acoustique en entrée. Il s'agit donc de trouver la $\overset{\smile}{W}$ qui maximise la probabilité *à posteriori* $P(W|O)$ de la séquence de mots sachant le signal acoustique. Ceci revient à résoudre l'équation suivante :

$$\overset{\smile}{W} = \underset{W \in L}{argmax}\ P(W|O) \tag{3.1}$$

tel que : L désigne le langage considéré.

L'utilisation de cette formule 3.1 est difficile à cause de l'estimation de la probabilité $P(W|O)$. Cette difficulté réside dans la grande variabilité dans l'ensemble de départ des observations acoustiques. Il est plus facile d'estimer la probabilité d'avoir une certaine séquence d'observations acoustiques O sachant une séquence de mots W. La formule de Bayes permet de décomposer le terme $P(W|O)$ en obtenant l'équation 3.2 :

$$\overset{\smile}{W} = \underset{W \in L}{argmax}\ \frac{P(O|W).P(W)}{P(O)} \tag{3.2}$$

Le problème est réduit à une tâche d'optimisation par rapport à la séquence de mots W. Grossièrement, la probabilité de la séquence des observations acoustiques $P(O)$ ne dépend pas de la séquence de mots W, ce qui induit à la formule 3.3 reliant seulement le modèle acoustique vraisemblant (*the acoustic model likelihood*) $P(O|W)$ et le modèle de langage à priori (*the language model prior*) $P(W)$.

$$\overset{\smile}{W} = \underset{W \in L}{argmax}\ P(O|W).P(W) \tag{3.3}$$

Les deux probabilités doivent être maximiser ensemble pour avoir la meilleure séquence de mots. Pour calculer $P(O|W)$, les mots peuvent être découpés en petites unités qui peut être présentées par des modèles de Markov cachés (Hidden Markov Model- HMM). Ces modèles permettent la modélisation de n'importe quel mot de la langue présent dans le *vocabulaire*, pourvu que sa prononciation (les différents phonèmes le constituant) soit connue et présente dans le *dictionnaire de prononciation.*

4 Composantes d'un système de RAP

Les systèmes de RAP à base de modélisation statistique sont constitués de plusieurs modules. La figure 3.1 illustre les principaux éléments de tels systèmes : l'**analyse acoustique** (*front-end*), le **modèle acoustique**, le **modèle de langage** et l'**algorithme de recherche** [Jelinek, 1997].

La partie de l'analyse acoustique est la fonction d'extraction des paramètres significatives d'un signal de parole entrant dans le système de RAP. Les modèles acoustiques sont habituellement estimés *à priori* sur des corpus audio par des outils statistiques. Les modèles de langage ont pour but de capter les propriétés d'une langue par l'estimation de la probabilité d'une suite de mot dans une séquence vocale. Un dictionnaire de prononciation est une liste de mots auxquels sont associées leurs différentes prononciations tout en respectant les séquences des unités acoustiques. Ce dictionnaire est le composant qui relie la modélisation acoustique à la modélisation linguistique. Le vocabulaire à son tour comprend tout les mots pouvant être reconnu par le système de la RAP et qui existent dans le dictionnaire de prononciation. Chaque mot du dictionnaire de prononciation est présent dans le modèle de langage et chaque unité phonétique utilisée dans la prononciation des mots est présente dans le modèle acoustique. L'algorithme de recherche est utilisé pour rechercher la (ou les) séquence des mots la plus probable qui a été prononcé par le locuteur.

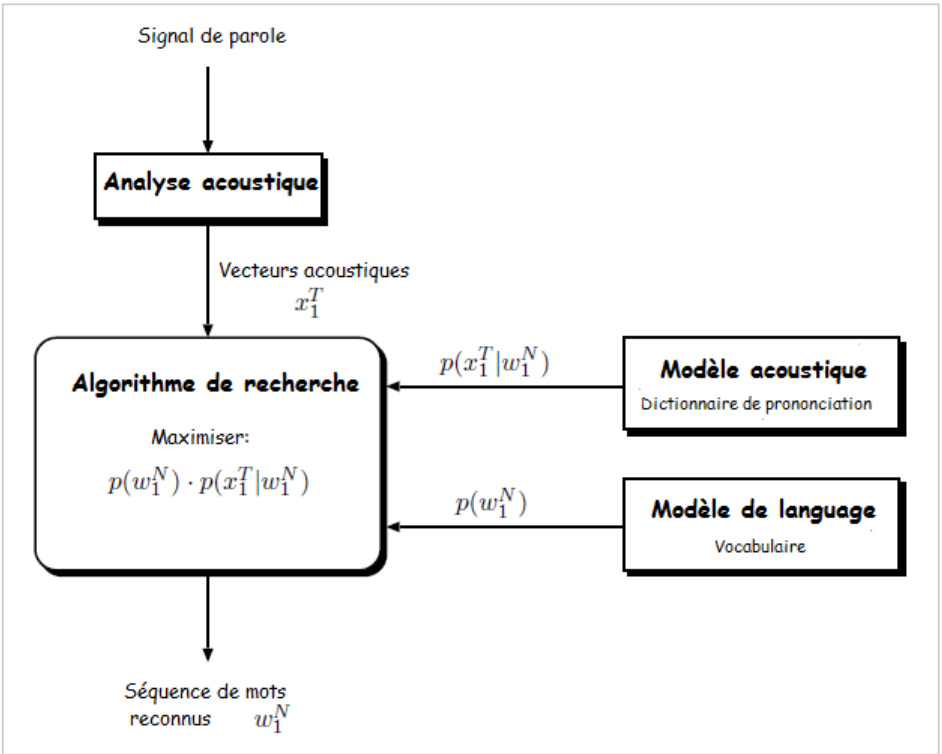

FIGURE 3.1 – Architecture des composantes d'un système de RAP.

Les ressources nécessaires pour la construction d'un système de la RAP sont : un *corpus textuel* pour développer des modèles de langages et le *vocabulaire* relatif, un *corpus de parole* pour développer des modèles acoustiques et le *dictionnaire de prononciation* relatif au vocabulaire. Une fois la phase de modélisation est terminée et le système est opérationnel, on peut incorporer comme entrée des signaux audio pour obtenir en sortie leurs hypothèses de transcriptions textuelles.

4.1 Analyse acoustique

Le signal de parole ne peut pas directement être transformé en hypothèse de séquences de mots. L'extraction des paramètres acoustiques (ou l'analyse acoustique - *front-end problem*) est une étape importante puisqu'elle s'occupe de déterminer les caractéristiques pertinentes du signal audio variant dans le temps. Pour effectuer une telle paramétrisation avec le minimum de perte possible, il est nécessaire de l'effectuer à intervalles suffisamment réguliers pour capturer chaque variation du signal. Il est généralement admis que le signal de parole peut être considéré comme quasi-stationnaire sur une fenêtre de l'ordre de 10 millisecondes. C'est pour cela que la paramétrisation est généralement effectuée sur des fenêtres glissantes de 30 millisecondes décalées de 10 millisecondes. Pour réduire les discontinuités dans le signal et ainsi améliorer la qualité de l'analyse, les fenêtres de 30 millisecondes sont ensuite pondérées par une fenêtre temporelle. Ces fenêtres sont utilisées par quasiment toutes les approches de paramétrisation. C'est la manière d'extraire les paramètres ainsi que leur nature qui fait la différence entre les approches. Un vecteur de paramètres représentant une observation acoustique est donc extrait pour chaque trame au moyen d'une des techniques de paramétrisation existantes. Ces paramètres sont ensuite envoyés au module de reconnaissance acoustique qui identifie les sons présents dans le signal phonétique considéré.

Coefficients acoustiques

Les paramètres acoustiques jouent un rôle essentiel dans un système de la RAP. Ils sont porteurs de l'information nécessaire à la reconnaissance puisqu'ils sont à la base de la construction des modèles acoustiques. Selon [Barrault, 2009], la modélisation de l'espace acoustique varie selon l'architecture du modèle acoustique et le type de paramètre acoustiques utilisés. Ainsi, deux modèles de même topologie fourniront des résultats différents s'ils utilisent des paramètres acoustiques différents.

Nous présentons une liste non exhaustive des approches de paramétrisation acoustique les plus utilisées :

1. Après la phase de numérisation et de quantification, le paramètre intuitif pour caractériser le signal est l'*énergie*. Cette énergie correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames successives du signal pour pouvoir mettre en évidence ses variations, en sommant des valeurs absolues des amplitudes des différents échantillons.
2. Les coefficients *MFCCs* pour *Mel-scaled Frequency Cepstral Coefficients* : Domaine cepstral) [Davis and Mermelstein, 1980] s'obtiennent en considérant pour le calcul du cepstre une représentation fréquentielle selon l'échelle perceptive des fréquences de Mel. Cette échelle est linéaire jusqu'à 100 Hz et logarithmique au-delà : c'est une échelle proche du traitement non linéaire de l'oreille humaine.
3. Le codage *LPC* (*Linear Predictive Coding* : Domaine temporel) [Markel and Gray, 1976] est basé sur le fait qu'un échantillon $s(t)$ à un instant t d'un signal vocal peut être approximé par une combinaison linéaire des échantillons précédents.
4. Le codage *PLP* (*Perceptual Linear Predictive* : Domaine spectral) [Hermansky and Cox Jr, 1991] vise à introduire des connaissances issues de la psycho-acoustique dans l'estimation des modèles auto-régressifs semblables à ceux utilisés dans l'analyse de prédiction linéaire. Ce codage repose sur l'utilisation de la transformation cepstrale issue d'une analyse en banc de filtre à échelle Bark. Un modèle auto-régressif est ensuite estimé sur la racine cubique des énergies de sortie. Cette échelle

réduite tend aussi à se rapprocher de l'oreille humaine qui possède une bonne résolution spectrale en basse fréquence et une mauvaise résolution en haute fréquence.

5. Le *taux de passage par zéro* (*zero crossing rate*) représente le nombre de fois que le signal passe par la valeur centrale de l'amplitude (généralement zéro) dans sa représentation amplitude/temps. Ce paramètre est fréquemment employé dans les algorithmes de détection de section voisée/non voisée pour un signal audio. En effet, la nature aléatoire du bruit possède généralement un taux de passage par zéro supérieur à celui des parties voisées. Cette mesure se montre aussi très intéressante dans le cadre d'une détection de parole en amont d'un système de reconnaissance [Gaillard et al., 1997].

Un état de l'art plus détaillé sur les approches d'extraction des paramètres acoustiques peut être consulté dans [Essid, 2005], [Barrault, 2009]), [Zolnay and Haeb-Umbach, 2006] et [Dave, 2013].

Post-paramétrisation

Sachant que les paramètres acoustiques peuvent être sensibles aux changements du canal ou de l'environnement, il est nécessaire d'appliquer d'autres traitement pour pouvoir gérer aux mieux les contraintes liées aux conditions provoquant la distorsion du signal. Les modifications que subi le signal acoustique lors de sa transmission sont souvent regroupées sous le terme *effet de canal de transmission*. Si l'on prend comme exemple l'enregistrement via un microphone, la réponse en fréquence de ce dernier introduit une distorsion qui modifie les fréquences identifiables dans le signal. Si l'enregistrement de la voix est réalisé par le biais d'une ligne téléphonique, la réduction fréquentielle est encore plus forte. En effet dans ce cas la bande passante se situe entre 300 et 3400 Hertz, ce qui élimine toutes les autres fréquences. Avec l'utilisation des serveurs de reconnaissance distribuée, le canal peut aussi comporter une transmission via le réseau Internet dite transmission de Voix sur IP (*Voice over IP* - VoIP) comme dans le cas des applications de visioconférence. Le canal provoque non seulement une distorsion dûe au codage de la voix mais aussi du fait que l'implémentation des protocoles est basée sur UDP/IP (*User Datagram Protocol*) [1], une perte de paquets et de données dans le signal à reconnaître est possible. Aussi, les systèmes de RAP doivent faire face au fait de devoir traiter un grand nombre de *Locuteurs différents* dans ces conditions variables d'enregistrement.

Les paramètres acoustiques peuvent être normalisés de façon à être rendus plus robustes aux distorsions dues aux canaux de transmissions. Il est possible de réduire la différence entre les données servant à apprendre les modèles de reconnaissance en utilisant différentes méthodes. Une liste non exhaustive des méthodes les plus utilisées pour la post-paramétrisation est présentée :

1. L'adaptation jacobienne (JA) est une approximation linéaire de la méthode de Combinaison parallèle des modèles (*Parallel Model Combination* - PMC) dans le domaine cepstral.
2. Le Lissage du vecteur champs (*Vector Field Smoothing* - VSF) est une extension de l'adaptation jacobienne.
3. L'analyse *RelAtive SpecTrAl* (RASTA) [Hermansky and Morgan, 1994] est un traitement de post-paramétrisation ayant pour but de supprimer les composantes spec-

1. Le protocole de datagramme utilisateur (*User Datagram Protocol* -UDP) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI, il appartient à la couche 4, comme TCP.

trales dont l'évolution temporelle est plus rapide ou plus lente que celle du conduit vocal humain. La méthode RASTA est intégrée à une analyse PLP. En effet, après avoir effectué la transformée de Fourier discrète à court terme, on calcule le spectre d'amplitude en bandes critiques. On applique le logarithme pour récupérer l'enveloppe spectrale du signal comme pour une analyse cepstrale. On effectue ensuite un filtre passe bande qui a pour conséquence de supprimer les composantes constantes ou lentes du signal. On réalise après une compression de l'amplitude par l'application d'une racine cubique. Enfin, on calcule les coefficients selon la méthode LPC classique.

4. La *Normalisation moyenne et variance des paramètres cepstraux* (CMVN) est une technique de post-paramétrisation qui consiste à retirer la moyenne de la distribution de chacun des paramètres cepstraux (la composante continue), et à ramener la variance à une variance unitaire en les divisant par l'écart type global des paramètres acoustiques. Quand seule la moyenne est normalisée, on parle alors de la méthode *Cepstral Mean Normalisation-CMN* [Acero, 1990] [Liu et al., 1993]. Quand la différence de la moyenne est normalisée, on parle alors de la soustraction de la moyenne cepstrale *Cepstral Mean Subtraction-CMS*. Ces méthodes nécessitent une connaissance du canal pour pouvoir construire des bases acoustiques pour l'apprentissage des modèles acoustiques. Ces normalisations du cepstre moyen sont des techniques qui agissent dans le domaine cepstral et permettent de réduire l'influence du canal de transmission.
5. La *Normalisation de la longueur de conduit vocal* (VTLN) [Zhan and Waibel, 1997], [Cohen et al., 1995] est une technique de normalisation de locuteur intervenant au niveau des paramètres acoustiques. Elle est largement répandue dans les systèmes de reconnaissance à grands vocabulaires. Cette normalisation repose sur une modification linéaire par morceaux de l'échelle des fréquences afin de compenser les différences de longueur de conduit vocal entre les locuteurs, leurs accents, leurs styles de vocabulaires, etc. qui peuvent influencer les performances globales du système de la RAP. Le coefficient de normalisation est sélectionné parmi un ensemble de valeurs (0,8 à 1,25) pour maximiser la vraisemblance des données de test, en utilisant une transcription obtenue avec un décodage rapide.
6. L'apprentissage fMLLR (*feature Maximum Likelihood Linear Regression*) [Gales, 1998] s'applique sur les paramètres directement issus de l'observation (comme la technique VTLN). Ainsi les paramètres acoustiques sont rapprochés de ceux de l'apprentissage, en modifiant leur espace de représentation et en débruitant les caractéristiques particulières du signal (locuteur, bruit, etc.).

Dérivées première et seconde

Les trames du signal de parole sont corrélées. La corrélation existe dans le domaine temporel (entre plusieures trames) et dans le domaine spatial (entre différents coefficients de la même trame). Pour avoir un modèle acoustique précis et performant, il est nécessaire de prendre en compte implicitement ou explicitement la corrélation dans le modèle. La plupart des systèmes de la RAP utilisent des paramètres MFCC, qui sont faiblement corrélés. L'ajout des coefficients de régression du premier et du deuxième ordre [Furui, 1981] permet de prendre en compte la corrélation temporelle. Or, il est important de noter que l'utilisation des coefficients dynamiques de régression contredit une des hypothèses fondamentales du HMM disant que les observations sont conditionnellement indépendantes entre elles, étant donné l'état du modèle. Heureusement, les évaluations expérimentales

montrent que cette contradiction ne détériore pas les résultats de la reconnaissance [Illina, 2005].

Le but final de l'extraction des paramètres est de modéliser la parole. Vu la variabilité de cette dernière, la valeur de l'énergie est importante mais elle n'est pas suffisante seule pour donner toute l'information portée par ce paramètre. Il est donc souvent nécessaire et utile comme déjà mentionné, de recourir à des paramètres acoustiques et des informations sur leur évolution dans le temps. Les dérivées première et seconde sont calculées pour représenter la variation ainsi que l'accélération de chacun des paramètres (voir la figure 3.2). La robustesse de la représentation obtenue avec l'accumulation de tout ces paramètres est accrue mais cela entraîne la multiplication par 3 l'espace de représentation.

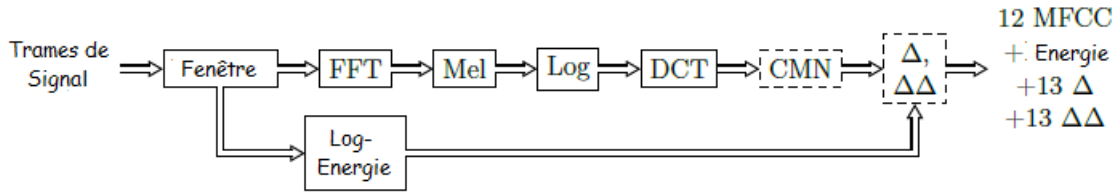

FIGURE 3.2 – Processus d'extraction des 39 coefficients MFCC.

Pour calculer les paramètres cepstraux, un fenêtrage de *Hamming* est appliqué sur le signal. Le signal est converti du domaine temporel au domaine spectral en appliquant une transformée de Fourier discrète (DFT) ou une transformée de Fourier rapide (FFT). Pour filtrer les bandes spectrales les plus pertinentes, un banc de filtres Mel est appliquée, suivie d'une transformation logarithmique et d'une transformée en cosinus discrète (DCT), pour pouvoir enfin déduire les caractéristiques cepstraux. Ces derniers peuvent être normalisées en appliquant une *soustraction du cepstre moyen* (*Cepstral Mean Subtraction* - CMS) ou une *normalisation du cepstre moyen* (*Cepstral Mean Normalization* - CMN). En plus des 12 cepstraux obtenus, on ajoute le log-énergie du signal pour avoir au total 13 paramètres. Enfin, les dérivées temporelles première et deuxième des 13 coefficients obtenus sont calculées et ajoutées afin de mieux saisir les propriétés dynamiques de la parole, cela permet d'avoir des vecteurs acoustiques de 39 paramètres.

En effet dans [Mezzoudj, 2011] et [Mezzoudj and Benyettou, 2012], l'ajout des dérivées première et seconde aux vecteurs acoustiques pour des données phonétiques extraites du corpus TIMIT [2]) a apporté un gain considérable de 15% en terme de taux de reconnaissance. Un passage de 58.3% avec les 13 paramètres MFCC à 73.02% avec les 39 paramètres. Sachant que la taille des vecteurs acoustiques est devenu plus importante (un passage de 416 caractéristiques à 1677 au maximum), l'algorithme génétique est adapté puis combiné avec un classifieur à vecteurs de support (SVM) [3] pour extraire les paramètres acoustiques considérés pertinents. Une sélection de 62% des données acoustiques a permet d'obtenir un gain de 1.6% en taux de reconnaissance.

Réduction de l'espace de représentation

Vu que l'espace de représentation du signal acoustique est souvent de taille conséquente qui peut atteindre des dizaines de paramètres. Plusieurs méthodes pour la réduction de la dimension sont proposées dans la littérature : l'analyse discriminante linéaire (*Linear Discriminant Analysis* - LDA)[Haeb-Umbach and Ney, 1992] permet de pallier certaines catégories de bruits [Siohan, 1995]. Cette technique est proche à l'analyse en composantes principales (*Principal Component Analysis* - PCA) [Pinkowski, 1997]. Une étude comparative entre les deux méthodes LDA et PCA, qui montre leurs performances et leurs complémentarités, peut être consultée dans [Hung et al., 2001] et [Zolnay and Haeb-Umbach, 2006].

2. https ://catalog.ldc.upenn.edu/LDC93S1
3. http ://svmlight.joachims.org/, consulté en mars 2017

L'analyse discriminante linéaire (LDA) est une méthode utilisé pour les statistiques, la reconnaissance de formes et l'apprentissage automatique afin de trouver une combinaison linéaire qui caractérise ou sépare deux ou plusieurs classes d'objets ou d'événements. La combinaison résultante peut être utilisée comme un classificateur linéaire, ou plus couramment pour la réduction de la dimensionnalité avant la classification. Elle permet l'obtention de paramètres considérés comme discriminants en appliquant une transformation linéaire de l'espace d'entrée (vecteur acoustique initial) vers un espace de taille réduite. Son principe consiste tout d'abord à calculer les vecteurs moyens et les matrices de covariances pour les differents classes possibles. Ensuite on calcule les deux matrices de covariance : inter-classe et intra-classe pour les classes. Enfin, on cherche une transformation permettant de maximiser la variabilité intra-classes tout en minimisant la variabilité inter-classes.

Le problème avec la LDA, c'est qu'elle permet une classification sur les moyennes des classes sans prendre en compte l'information discriminante présente dans la différence des matrice de covariance de ces classes. Cette technique est incapable de traiter les données dans le cas *hétéroscédastique*, càd le cas dans lequel les classes n'ont pas de matrices de covariance égale. Cette limitation devient très évidente dans le cas de deux classes avec deux matrice de covariance différentes. D'où l'introduction de la méthode HLDA (*heteroscedastic linear discriminant analysis*) [Kumar and Andreou, 1997] et d'autres variantes SHLDA [Karafiát et al.], MAP-SHLDA, etc. pour traiter ces types de problèmes.

4.2 Modéle acoustique

Etant donné que le vocabulaire des mots possibles à reconnaître par un système de RAP peut être très grand, chaque mot est décomposé en une séquence d'unités de sons élémentaires appelés *phones*. Un phone n'est autre que la réalisation acoustique d'un phonème. Vue la nature stochastique du signal de parole, les locuteurs ne prononcent pas souvent un mot de la même manière. Cette variation de prononciation se manifeste par la durée du son emis et par son contenu spectral (observations). En plus, les phonèmes, quant ils sont prononcés dans des contextes différents peuvent produire des variations du contenu spectral suite à la co-articulation.

Les modèles acoustiques sont sensés de modéliser la totalité de l'espace acoustique. Ils sont obtenus grâce à l'ensemble des enregistrements audio des locuteurs. Pour la reconnaissance de la parole continue à grand vocabulaire (*Large Vocabulary Continous Speech Recognition* - LVCSR), le choix d'unité de reconnaissance souvent employé est le *phone* ou le phonème. En effet, la modélisation d'un nombre constant de phonèmes est plus appropriée que la modélisation d'un nombre de mots. Ce dernier principe peut être utile pour un système de la RAP à petit vocabulaire relativement à la taille du vocabulaire considéré.

En général et selon une notation phonologique économique, il est possible d'associer à chaque mot une prononciation canonique. La linéarité du signifiant acoustique (déroulement dans le temps) entraîne que les phonèmes se présentent a priori les uns après les autres en formant une chaîne (un mot). A cette représentation phonologique est associé une modélisation dite acoustique. Cette modélisation repose sur la paramétrisation du signal, qui consiste souvent à calculer de coefficients cepstraux de Mel MFCC. La réalisation statistique des paramètres acoustiques de chaque phonème est représentée par un modèle de Markov Caché (*Hidden Markov Model* - HMM) : chaque phonème est représenté par 3 états (ou 5 états), à chaque état est associé une densité multigaussienne (*Gaussian Mixture Model* - GMM). Les densités GMM avec un grand nombre de composantes visent

à tenir compte de la variation phonétique effectivement observée dans le signal physique de la parole en relation avec le sexe, l'âge, l'accent du locuteur et les bruits. Un schéma illustrant cette modélisation est présenté dans la figure 3.3.

Chaque phonème de la langue sera modélisé par un jeu de HMM-GMM distincts représentant des allophones de ce phonème, afin de tenir compte des effets de coarticulation. Si le dictionnaire de prononciation contient des variantes de phonèmes (allophones), le modèle acoustique de mot linéaire peut devenir un graphe intégrant toutes les variantes. Ce dictionnaire permet ainsi d'expliciter des variantes non représentées implicitement dans les modèles acoustiques.

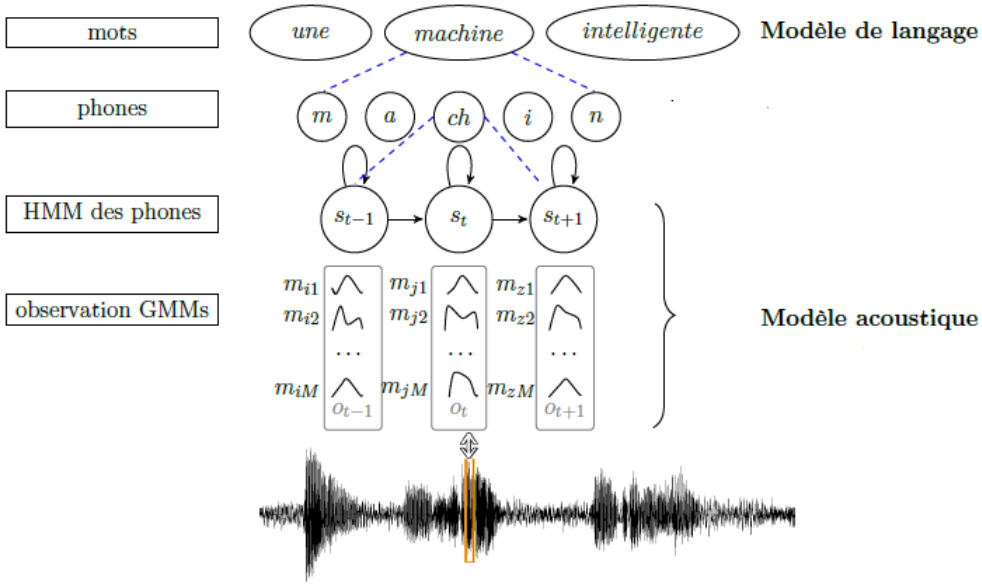

FIGURE 3.3 – Un modèle acoustique de mot est obtenu par concaténation de modèles HMMs élémentaires, homogènes à des phonèmes [Gorin, 2014].

Un système de la RAP utilise trois niveaux d'information pour décoder un message parlé. Au niveau des mots, on utilise le *modèle de langage* qui représente les successions possibles des mots. Cette modélisation est généralement construite à partir de l'analyse de séquences de mots provenant d'un grand corpus textuel. Le lexique spécifie le *vocabulaire* et associe à chaque mot une ou plusieurs séquences de phonèmes correspondant à la ou aux prononciations possibles des mots. Chaque phonème de la langue sera modélisé par un jeu de HMM-GMM distincts représentant des allophones de ce phonème. Le troisième niveau correspond aux *modèles acoustiques* qui traduit la réalisation acoustique de chaque élément modélisé tels que phonème, silence, bruit, etc.

L'utilisation des modèles de Markov cachés (Hidden Model Markov - HMM) dans le domaine de la RAP a débuté par des HMM avec des distributions discrètes [Jelinek, 1976] [4], semi-continues, puis des distributions continues, laplaciennes, gaussiennes et des mélanges de distributions. Pour prendre en compte plusieurs locuteurs, la modélisation du signal de parole est améliorée par une augmentation du nombre de paramètres des HMM. Dans les systèmes de la RAP actuels, les distributions associés à chaque état d'un HMM sont représentées par des mélanges de lois gaussiennes (GMM) [Jouvet et al., 2012b] ou ses différentes variantes [Gorin et al., 2014].

Une autre alternative pour modéliser l'espace acoustique est d'utiliser un réseau de neurones qui peut prendre plusieurs trames acoustiques en entrée et produit des probabilités *à posteriori* des états en sortie [Hinton et al., 2012]. Les détails nécessaires concernant ce types de modèles sont exposés dans le 5^{me} chapitre. Cependant, durant notre travail

4. Frederick Jelinek (18 Novembre 1932 – 14 Septembre 2010) un chercheur dans la théorie de l'information, la reconnaissance automatique de la parole et le traitement automatique du langage naturel.

expérimental, nous nous sommes contenté d'utiliser les modèles acoustiques standard à base d'HMM-GMM et non pas des modèles acoustiques neuronaux.

Modèles de Markov Cachés

La modélisation acoustique effectuée par des HMM [Jouvet, 1988] [Rabiner, 1989] s'appuient sur un formalisme ayant des fondements mathématiques introduit par Baum et ses collègues [Baum and Petrie, 1966] [Baum et al., 1970]. Les HMM sont des automates probabilistes à états finis qui respectent l'hypothèse markovienne d'ordre 1 : la connaissance du passé se résume à celle du dernier état occupé. Ils permettent de calculer la probabilité d'émission d'une séquence d'observations. Pour un système de RAP, les séquences d'observations sont les vecteurs des coefficients acoustiques du signal audio, des MFCC par exemple et leurs dérivées premières et secondes.

Dans la plupart des systèmes de reconnaissance de la parole actuels, les HMM basiques utilisés sont associés à des phonèmes (*phone*). Aussi, pour tenir compte de la variabilité de prononciation d'un phonème, un HMM est construit pour un phonème donné, associé à des contextes particuliers gauche et droit. Un contexte gauche d'un phonème est le phonème qui précède ce phonème et un contexte droit est le phonème qui succède à ce phonème. Ce triplet contexte gauche, phonème et contexte droit est appelé triphone ou *phonème en contexte*. Pour affiner la modélisation d'un phonème en contexte, la position de ce phonème dans un mot (début, milieu, fin ou phonème isolé) est parfois prise en compte.

Afin de réduire la taille du modèle, une factorisation d'états similaires est effectuée pour obtenir des *états partagés* (*shared triphone* ou *senone*) [Young et al., 1994]. En pratique, la modélisation des phonèmes en contexte pose des problèmes d'estimation. En effet, le nombre de modèles augmente exponentiellement avec la taille du contexte. Par exemple, avec 33 unités phonétiques de base, il faut théoriquement estimer $33^3 = 35937$ modèles acoustiques si l'on veut modéliser des tri-phones. Aussi, le fait que la quantité des données d'apprentissage est fixe : plus il y a de modèles à estimer, moins il y a de données pour chaque modèle. Il est aussi possible que certains contextes ne soient pas présents dans le corpus d'apprentissage. Pour résoudre ce problème et réduire la complexité des modèles, on utilise souvent des modèles contextuels avec partage d'états. Le principe est de regrouper les états des modèles qui sont proches pour pouvoir les estimer sur un grand volume de données disponibles plutôt que d'estimer des modèles séparés sur de petites quantités de données. Une solution proposée par Young et al. (1994) consiste à utiliser un arbre de décision phonétique pour regrouper les états qui peuvent être considérés comme contextuellement équivalents. Ainsi toutes les observations acoustiques affectées initialement aux différents états d'un même groupe seront utilisées pour estimer l'état représentatif du groupe.

La figure 3.4 présente un exemple d'HMM gauche-droit, avec un saut d'état possible.

Formellement, un HMM est définit par l'ensemble des paramètres $\lambda = (\pi, A, B)$ suivants :

1. La sequence des états $Q = (q_1, ..., q_t, ..., q_T)$, tel que $q_t \in \{1, ..., N\}$ est un état représentant la trame du temps t.
2. La distribution de l'état initial non associé à aucune observation, $\pi_i = P(q_0 = i)$ pour chaque état initial q_0, sinon $\pi_i = 0$.
3. La matrice des probabilités de transition $A = \{a_{ij}\}$ liées à la topologie du HMM en indiquant les probabilités de transition : $a_{ij} = P(q_t = j | q_{t-1} = i)$ d'un état vers un autre, $i, j \in \{1, .., N\}$ et $\sum_{j=1}^{N} a_{ij} = 1$.

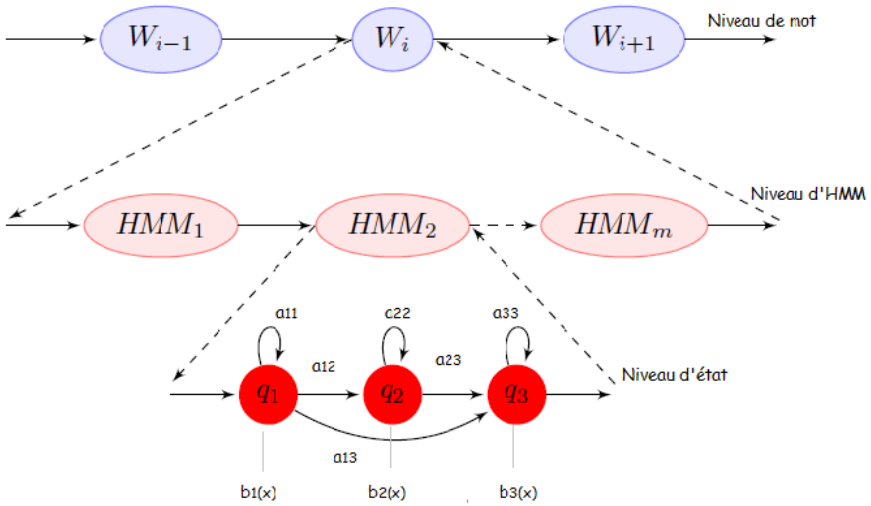

FIGURE 3.4 – Un exemple d'HMM gauche-droit.

Une observation correspond à une trame de parole. La structure la plus utilisée de HMM est de type gauche-droite à trois états : les états sont représentés par des ronds et les transitions par des flèches. La topologie utilisée des modèles suppose qu'on commence par le premier état. Par ailleurs, les transitions du HMM sont présentées par des probabilités a_{ij} d'un état i à un état j et la durée est modélisée implicitement par les transitions-boucles dans chaque état avec $j \geq i$: un état peut boucler sur lui-même. A chaque instant t un état j est donc atteint, et une émission o_t est générée et associée à une densité de probabilité $b_j(o_t)$.

4. Les fonctions d'émissions $B = b_j(o_t) = P(o_t|q_t = j)$ des observations, appelées aussi les fonctions de vraisemblance des états (*state likelihood function*) ou encore les densités d'observation pour chaque état j, tel que $o_t \in \mathbb{R}^d$ est un vecteur d'observations de dimension d. Il s'agit en pratique de mélanges de densités de probabilités gaussiennes définies par leurs vecteurs de moyennes, leurs matrices de covariances (des matrices diagonales) et une pondération associée à chaque densité de probabilité.

En général, la réalisation d'un système de la RAP à base d'HMM s'effectue en trois phases :

— Décrire la topologie du HMM (- GMM) qui reflète les unités élémentaires à traiter.
— Réaliser un apprentissage des paramètres du modèle(s) $\lambda = (\pi, A, B)$.
— Effectuer la reconnaissance de la parole par le calcul de la vraisemblance des séquences de mots.

Revenons au modèle statistique définit précédemment par l'équation 3.3 qui permet de calculer la vraisemblance de séquence de mots $P(O|W)$, on la reformule en utilisant la séquence des états du HMM $P(O|Q)$, où chaque état est associé à un vecteur d'observations d'une trame. Le but se redirige vers la recherche de séquence des états qui maximise la séquence des observations O. L'utilisation de la formule de Bayes induit à l'équation 3.4 :

$$\overset{\frown}{Q} = \underset{Q}{argmax}\ \frac{P(O|Q).P(Q)}{P(O)} = \underset{Q}{argmax}\ P(O|Q).P(Q) \tag{3.4}$$

Quelques simplification seront considérées :

1. L'hypothèse de la chaîne de Markov de premier ordre : chaque observation dépend seulement de l'état précédent, selon l'équation 3.5

$$P(q_t|q_1, ..., q_{t-1}, q_{t+1}, ..., q_T) = P(q_t|q_{t-1}) \tag{3.5}$$

2. L'hypothèse de l'indépendance des sorties : chaque observation dépend seulement de l'état qui produit cette observation, selon l'équation 3.6

$$P(o_t|q_1, ..., q_{t-1}, q_t, ..., q_T, o_1, ..., o_{t-1}, o_{t+1}, ..., o_T) = P(o_t|q_t) \tag{3.6}$$

En respectant ces deux hypothèses, l'équation 3.4 est reformulée par les paramètres du HMM et simplifiée comme formulé dans l'équation 3.7 :

$$\overset{\frown}{Q} = \underset{Q}{argmax} \prod_{t=1}^{T} P(o_t|q_t)P(q_t|q_{t-1}) = \underset{Q}{argmax}\ \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \tag{3.7}$$

En pratique, un seul état initial est supposé d'où la distribution initiale est exclue des dérivations $\pi_{q_0} = 1$.

Modèle de Mélanges de Gaussiennes

Une partie importante du HMM est la fonction d'émission des observations (*state likelihood function*) $b_j(o_t) = P(o_t|q_t = j)$ qui représente la probabilité d'observer le vecteur o_t en considérant l'état j au temps t. Initialement dans les anciens systèmes de la RAP, des HMM discrets été exploités : une fonction de vraisemblance discrète est associée à des vecteurs discrets. En général, pour assurer plus de précision, des fonctions à densité continue sous forme de Modèles de Mélanges de Gaussiennes (*Gaussian Mixture Model*-GMM) sont considérées. Dans un Modèle de Markov Caché avec densité d'observation à mélanges de gaussiennes noté HMM-GMM ou simplement HMM (*Hidden Markov Model with Gaussian Mixture Observation density - HMM-GMM or HMM*), la fonction de densité de probabilité d'une observation pour l'état du modèle j est définie comme une somme pondérée des M composantes du GMM par l'équation 3.8 :

$$b_j(o_t) = P(o_t|q_t = j) = \sum_{l=1}^{M} \omega_{jl} \mathcal{N}(o_t|\mu_{jl}, \Sigma_{jl}) \tag{3.8}$$

telle que : $\mathcal{N}(o_t|\mu_{jl}, \Sigma)_{jl}$ est une densité gaussienne avec un vecteur de moyenne μ_{jl}, une matrice de covariance Σ_{jl} où seulement les éléments de la diagonale sont utilisés et un vecteur d'observation o_t de taille n est formulé selon l'équation 3.9 :

$$\mathcal{N}(o_t|\mu_{jl}, \Sigma_{jl}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{jl}|}}\ exp\{-\frac{1}{2}(o_t - \mu_{jl})^T \Sigma_{jl}^{-1}(o_t - \mu_{jl})\} \tag{3.9}$$

Chaque composante l de l'état j a un poids ω_{jl} tel que $\sum_{l=1}^{M} \omega_{jl} = 1$. Différemment d'une simple gaussienne, une GMM a plusieurs pics (voir la figure 3.5). Par conséquent, les groupes séparés de points de données peuvent être modélisés par les différentes composantes de densité. Cette propriété est très importante pour la RAP (voir la figure 3.6), où différentes répartitions des caractéristiques acoustiques peuvent correspondre à la même unité phonétique en raison de la variabilité du signal de parole.

Apprentissage HMM : Algorithme Forward-Backward

La tâche d'apprentissage des HMM consiste à estimer leurs paramètres, connaissant un signal de parole et la transcription correspondante. La structure ou la topologie du modèle qui est représentée par le nombre d'états des modèles et les transitions autorisées entre ces états est fixée *a priori*.

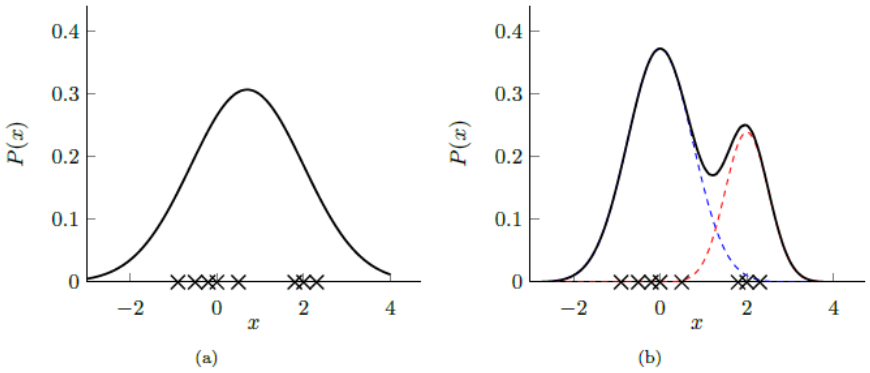

FIGURE 3.5 – Estimation de la distribution des données avec une simple gaussienne et un mélange de deux gaussiennes.

A titre d'exemple, cette figure compare les estimations d'une simple et unique fonction gaussienne et un mélange de deux fonctions gaussiennes pour un certain ensemble de points de données dans un espace à une dimension. Les données sont bien paramétrées par le mélange, tandis qu'une simple gaussienne conduit à une moyenne avec une grande variance.

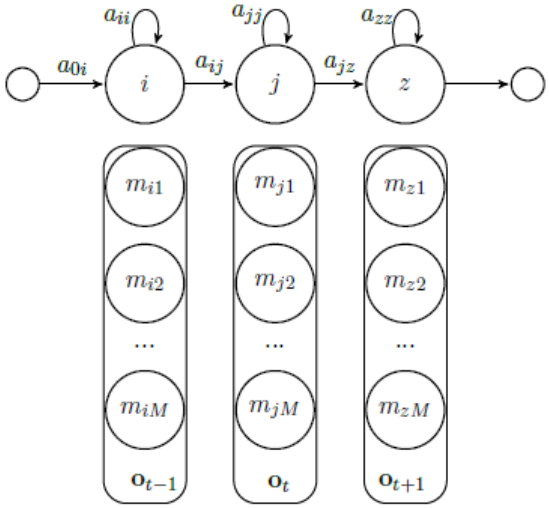

FIGURE 3.6 – Exemple d'un HMM avec des fonctions de densités GMM.

Une représentation de HMM-GMM en forme graphique : les composantes GMM peuvent être eux-mêmes considérés comme des états avec des sorties de valeurs continues. Les valeurs de sortie de ces états dépendent du vecteur d'observation et des paramètres de chaque composante GMM associée à des états HMM. On considère un chemin à travers une séquence Q particulière d'états et la séquence des composantes GMM $M = (m_1, ..., m_t, ..., m_T)$, où chaque $m_t \in \{1, ..., M\}$ désigne l'indice de composante de la densité associé à la trame de temps t.

Connaissant une suite d'observations émises par une suite de modèles, il est possible de modifier les paramètres de ces modèles de manière à rendre plus probable l'émission des observations. Il s'agit d'une estimation par le critère du maximum de vraisemblance (*Maximum Likelihood Estimation* - MLE). L'algorithme d'apprentissage est l'algorithme itératif d'Espérance-Maximisation (*Expectation-Maximization* - EM) avec une implémentation efficace de la *programmation dynamique*, connu sous le nom de l'algorithme de *Baum-Welch*. L'algorithme de *Baum-Welch*[5] est un algorithme de maximisation permettant de trouver le maximum de vraisemblance des paramètres des modèles probabilistes lorsque le modèle dépend de variables latentes non observables. L'algorithme *EM* alterne des étapes d'évaluation de l'espérance (*Expectation* - E), où (1) la vraisemblance

5. L'algorithme de Baum-Welch est introduit initialement pour la RAP dans [Baum and Petrie, 1966] puis étendu en *Expectation-Maximization* pour l'apprentissage automatique [Dempster et al., 1977].

est maximisée en optimisant une fonction qui est l'espérance de la log-vraisemblance sous une distribution conditionnelle sachant les observations et (2) une étape de maximisation (*Maximisation* - M) estimant le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape *E*. Les paramètres trouvés en *M* sont réutilisés comme point de départ d'une nouvelle phase d'évaluation de l'espérance. Les deux étapes sont réitérées jusqu'à l'obtention de la convergence.

Si on considère un modèle HMM initial décrit par l'ensemble des paramètres $\acute{\lambda}$, l'algorithme de *Baum-Welch* permet lors de l'apprentissage l'estimation de nouveaux paramètres $\breve{\lambda}$ qui maximisent la probabilité d'émission des observations $P(O|\lambda)$ selon l'équation 3.10

$$\breve{\lambda} = \underset{\lambda}{argmax}\ P(O|\lambda) \tag{3.10}$$

Une résolution analytique directe du problème d'apprentissage des HMM n'est pas possible mais l'algorithme de *Baum-Welch* permet une ré-estimation locale et itérative des paramètres du modèle [Baum, 1972] [Rabiner, 1989].

A la suite de la réestimation $(n+1)$ des paramètres du modèle λ_n, le nouveau modèle λ_{n+1} vérifie l'équation 3.11

$$Q(\lambda, \acute{\lambda}) = \sum_{Q} P(Q|O, \acute{\lambda}) log P(O, Q|\lambda) \tag{3.11}$$

L'algorithme *Baum-Welch* se base sur le calcul des variables *forward* et *backward* séparément qui représentent les parties gauche et droite de la séquence des états. La variable *forward* est définie comme la probabilité de la séquence d'observation partielle $\{o_1, ..., o_t$ de l'état j jusqu'au temps t par l'équation 3.12 :

$$\alpha_t(j) = P(o_1, o_2, ..., o_t, q_t = j|\acute{\lambda}) = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}] b_j(o_t) \tag{3.12}$$

De façon similaire, la variable *backward* est la probabilité de la séquence d'observation à partir du temps $t+1$ au temps-fin T sachant l'état i au temps t du modèle $\acute{\lambda}$, définie par l'équation 3.13 :

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T|q_t = i, \acute{\lambda}) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{3.13}$$

Quelques importantes variables sont définies :

1. La probabilité d'être à l'état i au temps t et à l'état j au temps $t+1$, en considérant le modèle initial $\acute{\lambda}$ et la séquence d'observation O (nombre attendu des transitions de l'état i à l'état j observant O) est définie par l'équation 3.14

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|O, \acute{\lambda}) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_t(n) a_{nm} b_m(o_{t+1}) \beta_{t+1}(m)} \tag{3.14}$$

2. La probabilité d'être à l'état i au temps t, en considérant la séquence des observations et le modèle (avec le nombre attendu des transitions de l'état i observant O) est définie par l'équation 3.15

$$\gamma_t(i) = P(q_t = i|O, \acute{\lambda}) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{n=1}^{N} \alpha_t(n) \beta_t(n)} = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.15}$$

En utilisant les deux équations des variables *forward* et *backward*, 3.14 et 3.15 respectivement, la re-estimation des probabilités des transitions est définit par l'équation 3.16 :

$$a_{ij} = \frac{\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, O|\acute{\lambda})}{\sum_{t=1}^{T} P(q_{t-1} = i, O|\acute{\lambda})} = \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.16}$$

Pour définir les formules de re-estimation des paramètres de la densité des gaussiennes, une autre variable aléatoire est introduite : $m_t \in 1, ..., M$ qui représente un composant particulier de la densité observée au temps t. Ensuite, la probabilité d'observer la composante l de l'état j à l'instant t est définie par l'équation 3.17

$$\gamma_t(j,l) = P(q_t = j, m_t = l|O, \acute{\lambda}) = \frac{\omega_{jl}\mathcal{N}(o_t|\mu_{jl}, \Sigma_{jl})}{\sum_{k=1}^{M} \omega_{jk}\mathcal{N}(o_t|\mu_{jk}, \Sigma_{jk})}\gamma_t(j) \tag{3.17}$$

Les formules de re-estimation pour les valeurs des poids des mélanges de gaussienne, la moyenne et la variance pour un état j et le composant l sont définies par les équations 3.18, 3.19 et 3.20 respectivement :

$$\omega_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, O|\acute{\lambda})}{\sum_{t=1}^{T} P(q_t = j, O|\acute{\lambda})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.18}$$

$$\mu_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, O|\acute{\lambda}).o_t}{\sum_{t=1}^{T} P(q_t = j, m_t = l, O|\acute{\lambda})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l).o_t}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{3.19}$$

$$\Sigma_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, O|\acute{\lambda}).(o_t - \acute{\mu}_{jl})(o_t - \acute{\mu}_{jl})^T}{\sum_{t=1}^{T} P(q_t = j, m_t = l, O|\acute{\lambda})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l).(o_t - \acute{\mu}_{jl})(o_t - \acute{\mu}_{jl})^T}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{3.20}$$

Généralement, la ré-estimation itérative des HMM réalisé par l'algorithme de *Baum-Walch* [Baum and Petrie, 1966] vise à maximiser la vraisemblance des modèles probabilistes a posteriori. L'apprentissage des modèles acoustiques consiste à estimer les vecteurs de moyennes et les matrices de covariances d'un ensemble de gaussiennes, ainsi que les pondérations permettant d'établir des mélanges à partir de ces gaussiennes. L'apprentissage étant itératif, les paramètres du modèle initial vont converger vers un jeu de paramètres optimal. Ces paramètres permettent de calculer des densités de probabilités qui constituent des valeurs de vraisemblance associées à l'émission d'une observation de l'état d'un HMM. À cela s'ajoute l'estimation des probabilités discrètes associées aux transitions entre les différents états des HMM. Évidement, les choix concernant la définition des modèles ont une influence sur la qualité de l'apprentissage [Laurent, 2010].

L'apprentissage des modèles acoustique peut être réalisé par l'algorithme de Baum-Welch [Baum and Petrie, 1966] comme déjà mentionné ou par la version de l'algorithme du forward-backward [Baum, 1972] ou même simplement à l'aide de l'algorithme *Viterbi* comme l'indique [Forney Jr, 1973]. Il existe d'autres techniques d'apprentissage discriminantes à utiliser au lieu de l'algorithme EM et ses variantes :

1. La méthode MMI (*Maximal Mutual Information*) introduite par Bahl (1986) afin d'adapter les paramètres de modèles de Markov pour la RAP. La méthode MMIE (*Maximal Mutual Information Estimation)* a ensuite été développée pour les systèmes de la RAP [Valtchev et al., 1997]. Cette méthode vise à maximiser les probabilités *a posteriori* des phrases qui cherchent à maximiser la probabilité *a priori*

du modèle correspondant aux données. L'idée est d'extraire les caractéristiques essentielles de chaque modèle, afin de pouvoir le différencier par rapport aux autres modèles.

2. L'algorithme à base des erreurs minimums des phonèmes (*Minimum Phone Error* - MPE) [Povey and Woodland, 2002] fonctionne sur un principe similaire à celui de MMI mais tend à minimiser le taux d'erreurs par phonèmes.
3. L'algorithme à base des erreurs minimums des mots (*Minimum Word Error* - MWE) [Heigold et al., 2005] fonctionnent aussi sur un principe similaire à celui de MMI mais tendent à minimiser le taux d'erreurs par mots.
4. Le critère de discrimination de trame (*Frame Discrimination Criterion* - FD) [Kapadia, 1998].
5. Le critère de maximum de vraisemblance conditionnelle (*Conditional maximum likelihood Criterion* - CML) [Nádas et al., 1988].
6. Le critère d'erreur de classification minimum (Minimum Classification Error Criterion - MCE) [Juang and Katagiri, 1992] qui tend aussi à minimiser le taux d'erreur au niveau des phrases.

Un état de l'art sur les différentes techniques discriminatives dédiées à la parole peut être consulté dans [Vertanen, 2004] et [Rigoll, 2010].

Adaptation acoustique

Les modèles acoustiques sont habituellement estimés a priori sur les données d'apprentissage du système de RAP. Généralement, afin de ne pas cumuler toutes les sources de variabilité de la parole des HMMs spécifiques sont estimés en fonction des caractéristiques des locuteurs (hommes, femmes) et en fonction des enregistrements (bande large, bande téléphonique). Si les conditions acoustiques des documents à transcrire changent, il faut théoriquement estimer de nouveaux modèles pour ces données. Cependant, lorsque peu de nouvelles données sont disponibles ou que le lourd processus de ré-estimation des modèles ne peut être effectué, il est possible d'adapter les anciens modèles aux nouvelles données.

L'idée générale de l'adaptation acoustique est d'adapter les modèles acoustiques déjà appris pour en créer des nouveaux beaucoup plus proches des conditions de test, en utilisant les modèles initiaux et une quantité restreinte de nouvelles données. Ces techniques d'adaptation sont efficaces, par exemple, lorsque le système de RAP doit traiter un locuteur inconnu (voir la figure 3.7), des données extraites des données d'apprentissage permettent d'adapter les modèles existants en modèles spécialisés.

Afin de pouvoir réaliser ce processus d'adaptation, une transcription des nouvelles données audio doit être fournie. Ces méthodes d'adaptation sont alors dites supervisées. Cependant, il est possible que la transcription de référence ne soit pas disponible alors un décodage au moyen du système de RAP non-adapté permet d'obtenir une transcription initiale.

Plusieures techniques d'adaptation des modèles acoustique sont proposées dans la littérature [Shinoda et al., 2001]. Nous exposons brièvement les techniques les plus utilisées :

1. **Adaptation par maximum *a posteriori*** :
   La méthode bayésienne d'estimation du *Maximum a posteriori* (MAP) introduite pour la RAP dans [Gauvain and Lee, 1994] permet d'établir des contraintes probabilistes sur les paramètres de modèles acoustiques. Le critère MAP est appliqué aux modèles ayant fait l'objet d'un apprentissage préalable et pour lesquels on

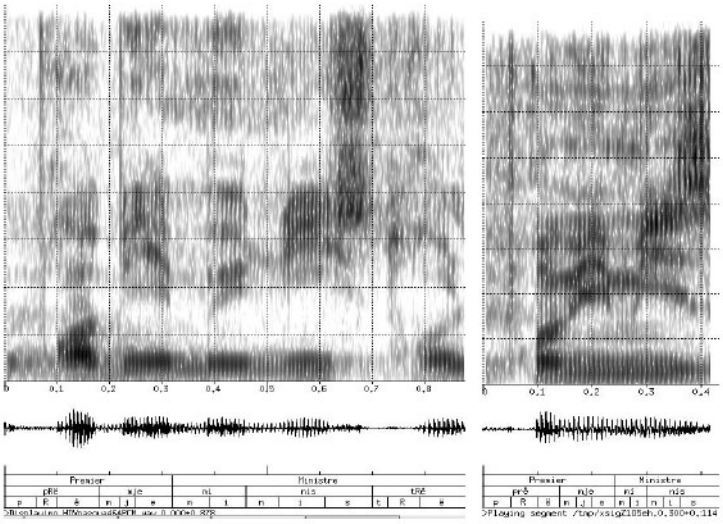

FIGURE 3.7 – Variantes de prononciation pour la séquence de mots *Premier Ministre*.

Un modèle acoustique de mot est obtenu par concaténation de modèles HMMs élémentaires de phonèmes. Ce spectrogramme represente deux variantes de prononciation pour la séquence de mots *Premier Ministre* produite par deux hommes de politiques dans le même contexte d'une émission d'interviews télévisée [Adda-Decker, 2007].

dispose de données *a priori*. Les modèles markoviens sont toujours estimés avec l'algorithme EM mais en maximisant la vraisemblance a posteriori (MAP) au lieu de la vraisemblance des données.

Cette technique permet d'obtenir de nouveaux modèles en réduisant la variance des modèles initiaux, en fournissant alors des modèles plus spécifiques grâce à l'utilisation d'une quantité restreinte de données d'adaptation, par exemple des modèles spécifiques au sexe du locuteur ou encore à des conditions acoustiques particulières.

On part du jeu de paramètres initial des modèles et on effectue quelques itérations de cette approche pour obtenir des paramètres qui modélisent mieux les données de test. L'adaptation MAP obtient de bons résultats et la quantité d'informations nécessaire à l'apprentissage est raisonnable en comparaison avec d'autres approche, comme le maximum de vraisemblance.

2. **Adaptation par régression linéaire** :
   L'adaptation par régression linéaire (*Maximum Likelihood Linear Regression*- MLLR) [Leggetter and Woodland, 1995], [Gales, 1998] est une technique d'adaptation des modèles acoustiques par régression linéaire, très efficace lorsque peu de données sont disponibles. Les transformations linéaires sont utilisées dans la procédure d'adaptation des modèles indépendants du locuteur. Empiriquement, les modèles dépendants d'un locuteur obtiennent de meilleurs résultats qu'un modèle indépendant du locuteur. La technique consiste alors à adapter le modèle indépendant du locuteur appris sur un grand volume de données à un locuteur précis, en essayant d'obtenir des performances les plus proches possibles de celles obtenues au moyen d'un modèle dépendant du locuteur. Des transformations linéaires sont utilisées pour l'adaptation à un locuteur. Elles permettent d'estimer les transformations propres à chacun des locuteurs du corpus d'apprentissage ainsi que les paramètres des HMM. La force de cette méthode réside dans le fait que l'adaptation peut facilement être réalisée pour un nouveau locuteur. En effet, une ou plusieurs transformations sont réalisées au moyen de cette méthode qui peuvent être ensuite

appliquées sur tout les modèles. Avec la méthode MLLR, les transformations sur les moyennes et variances des gaussiennes sont décorrélées les unes des autres.

3. **Adaptation par CMLLR** :
   Au contraire de l'adaptation MLLR, l'adaptation par régression linéaire par contraintes [Gopinath, 1998] (*Constrained Maximum Likelihood Linear Regression* - CMLLR) lie les transformations de la variance et de la moyenne. Au moyen de l'algorithme EM, les paramètres de l'adaptation sont optimisés selon le maximum de vraisemblance sur les données d'adaptation. Si des transformations linéaires identiques sont utilisées pour apprendre les modèles indépendants du locuteur, il est possible d'estimer conjointement les transformations propres à chacun des locuteurs de ce corpus et les paramètres de modèles markoviens. Les modèles résultants sont ensuite facilement adaptés à un nouveau locuteur.

4. **Adaptation SAT** :
   L'apprentissage adaptatif du locuteur (*Speaker Adaptative Training* - SAT) [Anastasakos et al., 1996] vise à réduire l'impact des variations inter-locuteurs lors de l'estimation des modèles acoustiques. Pour ce faire une transformation linéaire [Gales, 1998], des contraintes sont estimées pour chaque locuteur du corpus d'apprentissage afin de rapprocher les données du locuteur au modèle multi-locuteur. Un nouveau modèle est alors construit en utilisant les données transformées d'apprentissage. Ce modèle canonique est utilisé lors du décodage pour faciliter l'adaptation non-supervisée.

Par exemple, la transcription de conversations téléphoniques est une tâche bien plus complexe que la transcription d'émissions radiophoniques ou télédiffusées. Un système de transcription d'émissions d'information (*Broadcast News* - BN) peut évoluer vers un système de transcription de conversations en utilisant les différentes techniques d'adaptation [Gauvain et al., 2005].

Limitations des modèles de Markov cachés

Il est important de noter que, les HMM restent des modèles standard dans plusieurs systèmes de RAP de l'état de l'art malgré le nombre de limitations connues mais allégées par différentes adaptations.

Les HMM reposent sur un ensemble d'hypothèses simplificatrices. Tout d'abord, les données à l'entrée d'un HMM sont supposées être statistiquement indépendantes, et donc la probabilité qu'un vecteur soit émis au temps t ne dépend pas des vecteurs précédemment émis. Cette hypothèse est irréaliste. En effet, les vecteurs de paramètres acoustiques sont calculés sur des portions de signal d'une durée très petite (en général 30 ms.), il est donc incorrect de penser que deux trames successives ne possèdent aucune corrélation statistique. L'utilisation des dérivées premières et secondes des paramètres acoustiques comblent en partie cette imprécision mais les systèmes n'intègrent pas complètement la corrélation entre les trames de manière efficace.

Une autre limitation réside dans la modélisation de la durée, qui est implicite dans un HMM. Elle est déterminée par le critère visant a maximiser la probabilité a posteriori. Les HMM-GMM ne sont pas capables de représenter précisément des distributions très hétérogènes de paramètres acoustiques. Par exemple, le traitement de la variabilité inter-locuteur reste un problème pour les systèmes de RAP. Autrement, les limitations des HMM-GMM sont expliquées par les hypothèses d'indépendance conditionnelle assez fortes. Lors du décodage, il n'y a aucune garantie de cohérence de trajectoire (i.e. le che-

min optimal peut être associé à des composantes correspondant à des locuteurs ou à des conditions très différentes d'un état à un autre) [Gorin, 2014].

L'utilisation de modèles de trajectoire qui permet de prendre en compte l'évolution temporelle du signal de parole été proposé par Gong et Haton (1994) et amélioré dans [Gorin and Jouvet, 2014]. Aussi, une façon traditionnelle pour pallier ce problème consiste à utiliser des modèles acoustiques adaptés à la voix de groupes homogènes de locuteurs (ayant même sexe, âge, etc). Une classification automatique peut alors être appliquée sur chaque phrase, en supposant que le locuteur ne change pas au cours de même phrase. Dans [Gorin and Jouvet, 2014], deux approches différentes sont étudiées : (1) l'approche basée sur une classification non supervisée et une adaptation des modèles acoustiques pour chaque classe et (2) une nouvelle approche de modélisation acoustique basée sur la classification des données d'apprentissage pour structurer les composantes gaussiennes des densités GMM.

Une alternative utilisant différents architectures de réseaux de neurones pour la modélisation acoustiques est aussi exploitée dans la littérature.

Ressources pour les modèles acoustiques

En général, les ressources nécessaires pour l'apprentissage des modèles acoustiques sont un corpus audio et un dictionnaire de prononciation.

1. **Corpus audio** :
   Le recueil de signaux de parole est souvent réalisé en faisant l'enregistrement des textes prononcés par des locuteurs professionnels dans un studio. Cette tâche est fastidieuse et coûteuse. Cependant, il est intéressent d'avoir des ressources orales en grande quantité (par exemple, des dizaines ou des centaines d'heures de signaux vocaux) pour une bonne modélisation acoustique.

   Pour obtenir rapidement un corpus de parole, on peut enregistrer des émissions de type bulletin d'information des chaînes de radio mais il sera difficile d'obtenir les fichiers textes correspondants, sachant que la transcription manuelle de ce genre de fichiers est très coûteuse. En 1995, et afin de répondre à ce besoin qui grandissait de plus en plus, l'intention du NIST (*National Institute of Standards and Technology*) a été attirée par les journaux radiophoniques (*Broadcast News*) qui étaient relativement faciles à collecter (par rapport à un enregistrement de corpus classique). C'est ainsi que le LDC (*Linguistic Data Consortium*) s'est chargé d'obtenir les droits pour enregistrer, transcrire et distribuer ces données à la communauté du traitement automatique des langues.

   Parmi les corpora audio utilisés dans les expérémentations des systèmes de RAP de la littérature, on trouve le corpus *Resource Management* (RM) [6], le corpus *GlobalPhone* (GP) [Schultz, 2002], le corpus Timit [7], le corpus *Wall Street Journal* (WSJ) [8] [9]. Malheureusement, ces corpora distribués par le LDC (*Linguistic Data Consortium*) sont payants.

   En 1992, le corpus de conversations téléphoniques *Switchboard Corpus* [10] été enregistré (Godfreyet al., 1992) dans le but d'étudier ce type de parole qui s'est avéré plus difficile à traiter que la parole préparée. Ainsi, 2500 conversations pour 500

6. https ://catalog.ldc.upenn.edu/LDC93S3A
7. https ://catalog.ldc.upenn.edu/ldc93s1
8. https ://catalog.ldc.upenn.edu/ldc93s6a
9. https ://catalog.ldc.upenn.edu/docs/LDC94S13A/wsj1.txt
10. https ://catalog.ldc.upenn.edu/ldc97s62

locuteurs été enregistrées. Les conversations portaient sur divers sujets imposés à l'avance à des locuteurs qui ne se connaissaient pas. Il s'est avéré que ces conversations n'étaient pas naturelles.

Dans le but d'obtenir des données provenant de conversations spontanées, un programme d'enregistrement d'appels téléphoniques réels été mis en place. La procédure consistait à offrir des appels gratuits aux personnes qui acceptaient que leurs conversations soient enregistrées anonymement. Ceci a permis de construire le corpus *Callhome* (Canavan et al., 1997), qui contenait des sous-corpus en Anglais, Espagnol (avec dialectes variés), Arabe (Egyptien), Mandarin, Japonais et Allemand.

2. **Dictionnaire de prononciation** :
   On suppose que le système de RAP connaît le vocabulaire des locuteurs et la recherche est restreinte aux séquences de mots présents dans le dictionnaire. Le dictionnaire de prononciation (ou de phonétisation) [Strik and Cucchiarini, 1999] [Tang et al., 2012] liste les mots et leurs prononciations sous forme de suites de phones. Il est utilisé lors de l'apprentissage des modèles acoustiques. Aussi, il fournit le lien entre les séquences des unités acoustiques et les mots représentés dans le modèle de langage.

   Le décodeur du système de RAP est capable de fournir seulement les suites de mots connus et présents dans son vocabulaire. Ce vocabulaire étant de taille finie, il ne couvre pas tout les mots du langage. Il arrive donc qu'un mot prononcé n'y soit pas présent dans le vocabulaire, il est alors désigné sous le terme de *mot hors-vocabulaire* (*Out Of Vocabulary* - OOV). La performance du système de RAP est directement liée au taux de mots hors vocabulaire.

   Alors que les corpora de parole peuvent être collectés, le dictionnaire de prononciation n'est pas directement disponible. La création d'un tel dictionnaire nécessite un jeu de phonèmes et un vocabulaire. Pour obtenir le dictionnaire de phonétisation, plusieurs approches sont possibles. La première approche envisagée est de créer le lexique manuellement. L'avantage de cette approche réside dans le fait que les prononciations possibles de chaque mot sont fiables puisqu'elles sont vérifiées par un expert humain. Cependant, générer un lexique complet est très coûteux en ressources, et il est très difficile de couvrir la totalité des mots d'une langue. Des projets se sont notamment penchés sur la vérification manuelle de ces données tel que le projet BDLex [et de Calmès M., 1987], [Pérennou, 1998]. Une autre approche possible consiste à phonétiser les mots de manière automatique. Le système à base de règles LIA-PHON [Béchet, 2001] propose la phonétisation automatique pour transcrire les graphèmes en phonèmes. Généralement, ces deux approches sont utilisées conjointement en considérant les dictionnaires créés manuellement, puis de les enrichir par des mots manquants au moyen d'une technique de phonétisation automatique. Dans la littérature, l'approche simple générer automatiquement le dictionnaire de prononciation à base des graphèmes comme unité de modélisation été bien validée dans [Billa et al., 2002],[Seng et al., 2010].

4.3 Modèle de Langage

Pour la reconnaissance automatique de la parole continue, la seule information acoustique ne suffit pas pour transcrire correctement les suites de mots. La modélisation acoustique permet de réaliser la transcription phonétique d'une phrase. Or, en absence de

contraintes linguistiques, il est possible que cette suite soit très différente de la chaîne attendue. Par conséquent, il est nécessaire d'introduire des connaissances sur les niveaux supérieurs du langage. Le modèle de langage qui fournit la probabilité $P(W)$ de l'équation (3.3) est un élément clé dans le système de reconnaissance. Les modèles de langage (*Language Models* -LMs) entraînés sur des larges corpus textuels permettent de résumer statistiquement et introduire les contraintes linguistiques, syntaxiques ou sémantiques liées à une langue naturelle au sein du processus de décodage.

Les modèles de langage statistiques (dit aussi probabilistes) n'exigent pas des restrictions de type grammaticales, ce qui leur permettent d'être souples et d'accepter toute construction syntaxique pouvant apparaître dans le langage parlé. Les modèles de langage statistiques les plus utilisés dans les systèmes de RAP actuels sont à base de n-grammes [Rosenfeld, 2000] [Chen and Goodman, 1999]. Une alternative à base de réseaux de neurones artificiels est proposé par [Bengio et al., 2003] [Schwenk, 2007] [Mikolov, 2012] pour la modélisation du langage.

Les travaux présentés dans cette thèse ont pour objectif d'exploiter et d'améliorer la modélisation du langage. Le chapitre 4 est intégralement dédié à ce module des systèmes de la RAP et plus de détails y seront fournis.

Combinaison des modèles acoustiques et linguistique

La formule 3.3 suggère que la probabilité du modèle acoustique et la probabilité du modèle de langage peuvent être combinées à travers une simple multiplication. Cependant, il est nécessaire en pratique d'effectuer une pondération. Sans cela, la participation d'un des modèles est négligeable à cause de la différence d'ordre de grandeur des variations des deux distributions : lorsque la différence d'ordre de grandeur est trop importante, un des modèles prend l'ascendant sur l'autre en terme de puissance de discrimination. Ainsi, seul un des modèles est alors effectivement utilisé pour la prise de décision finale. Ceci s'ajoute à la différence de nature des deux probabilités : le modèle de langage fournit de probabilité de valeur discrète, alors que le modèle acoustique manipule des densités de probabilités. La solution la plus couramment utilisée pour atténuer ce problème consiste à ajouter un poids, noté lw (*linguistic weight*) et souvent appelé *fudge factor*, au modèle de langage. On a alors l'équation 3.21 :

$$\overset{\frown}{W} = \underset{W \in L}{argmax}\ P(O|W).P(W)^{lw} \tag{3.21}$$

où le poids lw est déterminé empiriquement à partir d'expériences effectuées sur un corpus de développement, la valeur choisie est celle qui optimise les performances du système de reconnaissance. Généralement $lw > 1$.

Les multiplications successives de probabilités mènent à manipuler des valeurs de plus en plus proches de 0. En pratique, les systèmes de RAP ne manipulent pas directement les probabilités mais ce sont les logarithmes de ces probabilités qui sont utilisés. Le passage aux logarithmes entraîne l'utilisation d'addition plutôt que de multiplication, ce type d'opérations conforte la propriété intéressante des logarithmes qui changent très lentement d'ordre de grandeur. Ainsi, la Formule 3.21 se ré-écrit en utilisant l'équation 3.22 :

$$\overset{\frown}{W} = \underset{W \in L}{argmax}\ log\ P(O|W) + lwlogP(W) \tag{3.22}$$

4.4 Algorithme de décodage

A partir de l'observation d'événements acoustiques et de connaissances a priori (vocabulaire, modèles acoustiques, modèles de langage), un système de RAP génère un ensemble d'hypothèses de séquences de mots. Cet ensemble est appelé espace de recherche. L'espace de recherche est généralement représenté sous la forme d'un graphe, appelé graphe de recherche, qui intègre les informations utilisées pour la génération des hypothèses de transcription des mots. Cet ensemble peut être codé sous la forme d'un graphe ou treillis de mots.

L'objectif de l'algorithme de 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 graphe ou un treillis de mots (dit espace de recherche) où chaque nœud représente un etat de phone donné à un temps t.

Ce graphe est exploré afin de trouver le chemin qui satisfait la fonction de coût 3.22, qui regroupe les hypothèses linguistiques et acoustiques. L'algorithme de décodage doit en extraire la (ou les) meilleure phrase qui maximise au mieux cette equation. Comme une exploration complète du graphe est irréaliste, il est nécessaire de limiter l'espace de recherche, qui croit de manière exponentielle, pour obtenir le bon résultat dans un temps de traitement acceptable.

Les algorithmes de décodage (ou dit algorithmes de recherche, ou de reconnaissance (en anglais *Hypothesis search*) incluent souvent des heuristiques réduisant l'espace de recherche en éliminant les chemins peu probables. Cela leur permet de choisir un nombre limité d'hypothèses à chaque instant et ainsi de n'explorer que l'espace suffisant ou nécessaire pour trouver la meilleure solution. Parmi les algorithmes de recherche [Vaufreydaz, 2002a], on distingue l'algorithme de Viterbi, l'algorithme A^*, l'algorithme à base de modélisation arborescente, l'algorithme de résolution de graphe (ou treillis) de mots, le décodage à pile (*stack decoding*) mis en œuvre par Jelinek (1969), etc. Formellement, deux grandes familles d'algorithmes existent : les algorithmes synchrones et asynchrones [11].

Les algorithmes synchrone sont les plus utilisés dans les systèmes de RAP, le plus répandu dans cette famille est l'algorithme de Viterbi en faisceau (*beam search*). Le principe de ces algorithmes est d'explorer le graphe d'hypothèses de manière synchronisée avec le signal de parole. Avec de tels algorithmes, il est difficile d'intégrer des informations contextuelles dans la fonction de coût utilisée. Il est essentiel de pouvoir utiliser un score linguistique contextuel fourni par le modèle de langage à la fonction de coût. Il est possible de modifier l'algorithme pour intégrer cette information en créant artificiellement des chemins du graphe dépendant de l'historique. Cette modification est lourde, c'est pour cela que les systèmes de RAP basés sur ce type d'algorithmes effectuent en général un décodage initial avec des modèles de langage simples (généralement des bi-grammes).

Les algorithmes asynchrones se basent sur l'idée d'explorer en profondeur le graphe. Les hypothèses ayant le score le plus élevé sont explorées en premier. L'algorithme est dit asynchrone car il peut, après avoir longuement exploré un chemin du graphe, décider de revenir en arrière et de prendre un autre chemin. Cette approche a été décrit par Jelinek (1969). Une implémentation utilisant l'algorithme $A*$ a été proposée par Paul (1992) est la plus couramment utilisée.

Un état de l'art sur sur l'ensemble des techniques de décodage peut être consulté dans [Schwartz and Austin, 1991], [Lacouture, 1995], [Aubert, 2000] et [Aubert, 2002]. Dans

11. L'algorithme synchrone au temps de Viterbi et l'algorithme A^* qui est asynchrone sont les plus utilisés.

cette section, les algorithmes de recherche les plus utilisés dans les systèmes de RAP sont présentés :

Algorithme de Viterbi :
En théorie, l'algorithme de Viterbi [Viterbi, 1967] est un algorithme de programmation dynamique permettant de trouver parmi tous les chemins possibles d'un graphe de reconnaissance celui ayant le coût le plus bas, comme il illustré dans la figure 3.8. En RAP, on l'utilise pour trouver le chemin ayant le score le plus élevé. Cet algorithme est à la base, sous une forme ou une autre, de presque tout les systèmes de la RAP utilisant comme modèles acoustiques des HMM [Lacouture, 1995].

En utilisant le dictionnaire de prononciations et l'algorithme de *Viterbi*, il est possible de procéder à la phase de décodage permettant d'aligner des phonèmes sur le signal. Pour obtenir des modèles acoustiques performants, il est nécessaire que la phonétisation de chaque transcription soit la plus proche possible de la prononciation effective de la phrase correspondante. En considérant un HMM ayant les paramètres λ et une séquence d'observations O, la tâche de décodage consiste à calculer la séquence des états Q la plus probable. Soit le score du chemin de *Viterbi* représentant la probabilité la plus élevée au long du meilleur chemin qui se termine à l'instant t dans l'état j, donné par l'équation 3.23 :

$$v_t(j) = \max_{q_1,q_2,\ldots,qt-1} P(q_1,\ldots,q_{t-1},q_t=j,o_1,\ldots,o_t|\lambda) = \max_i [v_{t-1}(i)a_{ij}]b_j(o_t) \tag{3.23}$$

L'équation 3.23 définit la meilleure probabilité du chemin, mais en effet pour récupérer la séquence des états, les valeurs du pointeur secondaire (*backpointer*) doivent être calculés en prenant l'*argmax* au lieu de *max* dans la même équation :

$$bp_t(j) = \underset{i}{argmax}\ [v_{t-1}(i)a_{ij}] \tag{3.24}$$

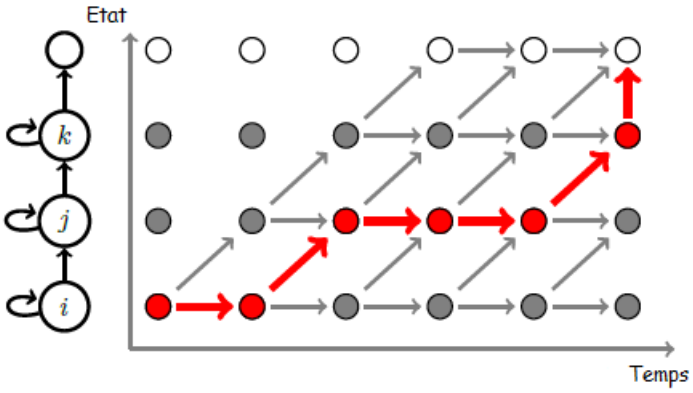

FIGURE 3.8 – Exemple de recherche d'un phonème par l'algorithme de *Viterbi*.

L'algorithme de Viterbi propose de simplifier le graphe de recherche au fur et à mesure à sa construction. En effet, lors de son déroulement on se retrouve rapidement avec des branches proposant les mêmes substitutions, mais avec des probabilités différentes. Il n'est pas nécessaire de dérouler celles ayant les plus faibles probabilités puisqu'elles ne peuvent plus être candidates pour décrire le message le plus probable [Gorin, 2014].

Pour appliquer effectivement l'algorithme de *Viterbi* pour l'ASR, les HMM de mots sont construits par concaténation des HMM phonétiques utilisant les états non-émetteurs. Les états non émetteurs sont connectés aux états finaux des mots avec les états initiaux

des mots candidats suivants. Ces transitions à mots-croisés sont également utilisées pour inclure des probabilités du modèle de langage. Le nombre d'opérations nécessaires pour le décodeur de *Viterbi* est au plus égal à N^2T (si tous les états sont connectés à tous les états), où N est le nombre d'états et T est le nombre des trames. En général, si K est un nombre moyen de transitions qui se terminent dans un état (habituellement $K \ll N$), l'algorithme de *Viterbi* nécessite KNT opérations. Néanmoins, comme le nombre total des états à évaluer peut être très grand dans un système de RAP a grand vocabulaire LVCSR (*Large Vocabulary Continuous Speech Recognition)*, la recherche heuristique faisceau (*beam search*) est mis en œuvre. L'idée de cette dernière est d'étendre le chemin pour la trame suivante seulement à partir des états, pour lesquels les probabilités de chemin sont supérieures à un certain seuil qui est sélectionné à chaque trame en fonction de la largeur du faisceau [Lowerre, 1990].

Algoritghme A^* :
Jelinek est le premier à appliquer l'algorithme A^* [Jelinek, 1969] qui une technique de l'intelligence artificielle pour le domaine de la RAP. Le principe dans cet algorithme est de conserver les hypothèses les plus prometteuses dans une pile. Le degrés de promesse d'une hypothèse est une valeur numérique appelée score total. Cette valeur est égale à la somme du score courant de l'hypothèse et d'une valeur heuristique représentant un estimé du score associé au meilleur chemin possible entre l'état courant et un état final. L'heuristique sur-estime toujours le potentiel d'un chemin et donne la meilleure solution.

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

- L'algorithme débute son parcours sur un nœud donné du graphe, le premier nœud pour la condition initiale. Il applique à ce nœud un coût qui représente généralement un score combinant une partie linguistique et une partie acoustique, puis il estime la distance séparant ce nœud du nœud terminal. La somme du coût et de l'évaluation représente le coût estimé du chemin menant à ce nœud. Le nœud est alors ajouté à une file d'attente prioritaire appelée la liste ouverte (*open list*).
- L'algorithme A^* retire le premier nœud de la file d'attente prioritaire. Si cette dernière est vide, il n'y a aucun chemin du nœud initial au nœud d'arrivée, ce qui est une condition d'arrêt de l'algorithme. Si le nœud retenu est le nœud d'arrivée, l'algorithme reconstruit le chemin complet à partir des informations sauvegardées dans la liste fermée *closed list* et s'arrête.
- Si le nœud n'est pas le nœud d'arrivée, tout les nœuds adjacents sont explorés. Pour chaque nœud successif, A^* calcule et stocke son coût. Celui-ci est calculé à partir de la somme du coût de son ancêtre et du coût de l'opération pour atteindre ce nouveau nœud.
- L'algorithme met également à jour la liste des nœuds qui ont été vérifiés, dans la liste fermée. Si un nouveau nœud existe déjà dans cette liste avec un coût égal ou inférieur, aucune opération n'est faite sur ce nœud ni sur son jumeau s'il se trouvant.
- La distance évaluée entre le nouveau nœud et le nœud d'arrivée est ajoutée au coût du nœud. Ce nœud est alors ajouté à la la liste ouverte, à moins qu'un nœud identique dans cette liste ne possède déjà un coût inférieur ou égal.
- Une fois ces trois étapes réalisées pour chaque nouveau nœud adjacent, le nœud original pris dans la liste ouverte est ajouté à la liste des nœuds explorés. Le nœud suivant est alors retiré de la liste ouverte et le processus recommence.

Stratégies Multi-passes

Afin d'accélérer la reconnaissance, le moteur de RAP effectue plusieurs passes pour déterminer le résultat final, appelé souvent des stratégies multi-passes. Une première passe rapide et grossière avec des modèles moins précis permet de restreindre l'espace de recherches des solutions en éliminant les mots du vocabulaire qui sont très différents du mot à reconnaître. La première passe génère une transcription qui sera réutilisée pour adapter les modèles acoustiques en fonction des locuteurs ou de la qualité d'enregistrement. Les premières passes permettent également de générer des graphes de mots qui peuvent être ré-explorés *a posteriori* avec des modèles de langages plus importants.

Ces stratégies en plusieurs passes permettent ainsi d'introduire à chaque itération une information supplémentaire en utilisant des modèles plus précis. Généralement, les informations rajoutées n'auraient pu l'être à l'itération précédente, car la quantité d'hypothèses en concurrence était trop importante. Dans ce cas, le but n'est pas seulement d'obtenir le mot correct mais aussi d'éliminer autant de mots-candidats que possible (sans éliminer le bon). En cas de autres passes successives, elles utilisent des modèles plus précis et des informations supplémentaires.

A titre d'exemple dans [Vaufreydaz, 2002a], lors d'une première passe sur un graphe de mots ou de phonèmes, les meilleurs chemins au niveau acoustique sont retenus et ordonnés (en général, via un algorithme de type *Viterbi*). Pour chaque instant τ où τ varie entre 1 et T, sachant que T est le nombre de trames dans le signal de parole. Aussi, pour chaque état q_i du graphe tel que i varie de 1 à N_Q, la probabilité maximale d'observation des τ premières trames le long du meilleur des chemins atteignant l'état q_i à l'instant τ. La phase de retour est réalisée par l'algorithme A^* en remontant le chemin à partir de l'état final du graphe (T, N_Q). Cet algorithme explore le graphe en y rajoutant ses contraintes (linguistiques ou autres) et en estimant le chemin restant (appelé sonde) grâce aux chemins pré-estimés sur le score acoustique.

5 Evaluation d'un système de RAP

Les sorties que génère un système de la RAP sont de trois types : les n-meilleures phrases transcrites, le graphe de mots reconnus ou le graphe de confusion[Razik, 2007].

1. Les *n-meilleures phrases* sont un sous ensemble de toutes les phrases qu'il est possible de générer suivant le vocabulaire et les modèles de langage. En général, la plupart des phrases sont similaires. Ainsi pour une phrase longue, il faut considérer un nombre important de séquences pour obtenir une variété suffisantes de phrases.
2. Le *graphe de mots reconnus* permet de représenter de manière plus précise et plus complète les informations issues généralement de la première passe. Un graphe de mots, comme illustré par la figure 3.9, inclut les multiples chemins possibles qui vont du début à la fin de la phrase. Les informations stockées sont le score acoustique et leurs instants du début à la fin.
3. Le *réseau de confusion* est une forme simplifiée du graphe de mots. Après le décodage, il est possible de modifier le graphe de mots de telle sorte que tout les arcs partant d'un nœud aient le même nœud de destination. On obtient alors un réseau de confusion dont tout les arcs sont pondérés par une probabilité calculée à partir des probabilités des mots du graphe initial. Les mots n'y sont plus localisés suivant leurs instants de début et de fin, mais suivant leurs positions dans la phrase.

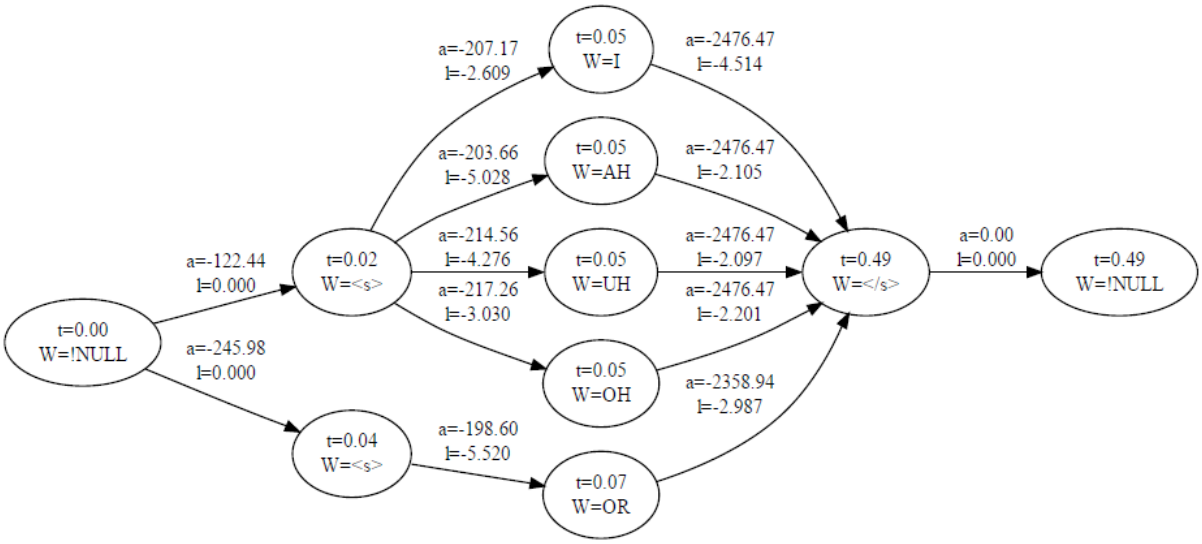

FIGURE 3.9 – Un exemple de graphe de mots.

L'exemple de graphe de mots pour un court énoncé : des mots, des informations de synchronisation ainsi que des log-vraisemblances acoustiques et même des scores des modèle de langage sont présentés [Seigel, 2013].

5.1 Mesures d'erreurs

La mesure la plus souvent employée pour évaluer les sorties un système de RAP est le *taux d'erreur mot* (*Word Error Rate* - WER). Ce taux est obtenu par un comptage du nombre de mots mal reconnus par le système réalisé, en utilisant une comparaison entre la meilleure solution du système (la séquence de mots reconnue) et la transcription manuelle du signal de parole (la séquence de référence).

Pour calculer le WER, les deux séquences de mots sont alignées et on mesure la distance de *Levensthein* (appelée aussi distance d'édition) qui permet de mesurer la similarité entre deux chaînes de caractères. Elle est égale au nombre minimal de mots qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre. Cette distance est d'autant plus grande que le nombre de différences entre les deux chaînes est grand.

Trois types d'erreurs possibles interviennent dans le calcul du WER :

— Les *omissions* sont les mots de la référence qui n'ont pas été reconnus par le système. Ils ne se retrouvent donc pas dans la solution fournie.

— Les *insertions* sont des mots reconnus par le système qui ont été insérés dans la solution en plus des mots de la référence.

— Les *substitutions* sont les mots qui ont été reconnus à la place d'autres mots de la référence.

Le WER est calculé en sommant les trois types d'erreurs et en normalisant par le nombre total des mots dans la référence, en utilisant l'équation 3.25 :

$$WER = \frac{Nombre\ d'omissions\ +\ Nombre\ d'insertions\ +\ Nombre\ de\ substitutions}{Nombre\ de\ mots\_reference} \tag{3.25}$$

Selon la littérature, d'autres mesures d'évaluation des systèmes de la RAP peuvent être utilisées :

— La *précision* et le *rappel* qui sont empruntés au domaine de la recherche d'informations. La précision mesure la capacité du système de rejeter les hypothèses de mots incorrectes, tandis que le rappel mesure la capacité du système à accepter

les hypothèses de mots correctes. Ces deux mesures sont définies par les équations 3.26 et 3.27 respectivement :

$$Prcision = \frac{Nombre\ de\ mots\ correctement\ reconnus}{Nombre\ de\ mots\ reconnus} \tag{3.26}$$

$$Rappel = \frac{Nombre\ de\ mots\ incorrectement\ reconnus}{Nombre\ de\ mots\ reconnus} \tag{3.27}$$

— Le *taux d'erreur mot Oracle* (*WER Oracle*) est obtenu en recherchant dans le graphe de mots la séquence de mots la plus proche de la référence du point de vue de la distance d'édition. Le *WER Oracle* peut être vu comme la limite théorique du WER qui pourrait être obtenu en choisissant idéalement les séquences de mots dans le graphe.

5.2 Intervalle de confiance

Afin de connaître la fiabilité des résultats du système de RAP en fonction du nombre d'échantillons utilisés pour les tests, il convient de calculer un intervalle de confiance pour chacun des résultats. Cet intervalle de confiance est calculé en considérant que l'apparition d'une erreur de reconnaissance sur un mot ou sa non-apparition est associée à une variable aléatoire binomiale, dont la distribution dépend des couples (mot reconnu, mot prononcé).

Le taux d'erreurs sur les mots, noté wer_f, est la proportion d'erreurs observées sur le corpus de test. Le paramètre k est le nombre de mots qui constituent le corpus de test. L'intervalle de confiance permet d'estimer, à partir de l'observation de wer_f sur les k échantillons disponibles, l'intervalle de valeurs dans laquelle se situe la proportion wer_p d'erreurs sur les mots pour une population infinie de mots répondant aux critères de l'application. Un intervalle de confiance est également défini par un niveau $1-\alpha$ qui permet de déterminer la fiabilité de cet intervalle. Généralement, l'intervalle de confiance est un intervalle à risques symétriques $\frac{\alpha}{2}$, c'est-à-dire que la valeur wer_f observée sur les k échantillons se trouve au centre de cet intervalle. Pour un nombre d'échantillons k suffisamment grand (habituellement, si $k > 100$), l'équation 3.28 définit l'intervalle de confiance de niveau $1-\alpha$ de wer_p [Estève, 2002] :

$$wer_f - \mu_{\alpha/2}\sqrt{\frac{wer_f(1-wer_f)}{k}} < wer_p < wer_f + \mu_2\sqrt{\frac{wer_f(1-wer_f)}{k}} \tag{3.28}$$

où la valeur de $\alpha/2$ dépend de α qui est disponible dans le tableau de la loi de *Student*[12]. L'intervalle de confiance le plus couramment utilisé est l'intervalle de niveau $1-0.95 = 0.05$, appelé aussi intervalle de confiance à 95% (dans ce cas, $\mu_{0.425} = 1.96$).

6 Transcription de la parole

Avec le développement rapide des capacités de stockage, de la puissance de calcul des machines, ainsi que les performances exigeantes en RAP basées sur les approches d'apprentissage statistique, les systèmes de RAP sont devenus très gourmands en termes de données. Ainsi, de nombreuses applications sont devenues de plus en plus envisageables

12. La loi de Student est une loi de probabilité, faisant intervenir le quotient entre une variable suivant une loi normale centrée réduite et la racine carrée d'une variable distribuée suivant la loi du χ^2.

comme la transcription automatique d'émissions radiophoniques, la transcription d'émissions télévisées [Illina, 2005], la transcription pour la traduction en une autre langue, la transcription de spectacles, l'apprentissage des langues, les systèmes de dialogue, etc.

Les données de type d'**émissions radiophoniques** (*Broadcast News*) présentent énormément de diversités. Les enregistrements obtenus peuvent contenir de la parole préparée, de la lecture, de la parole spontanée, des disfluences, des niveaux très variables de bruits suivant le type d'émission et les locuteurs intervenants, musique, parole + musique, jingles, silence, etc. Suivant le type d'application, différentes segmentations sont envisageables : (1) la segmentation musique/non-musique pour le traitement de la musique (classification par genre ou par instrument, etc), (2) la séparation parole/fond musical des segments de parole + musique pour le mixage audio ou la séparation des sources, etc. et (3) la segmentation parole/non-parole pour la transcription orthographique et éventuellement la recherche d'information.

En général, la transcription automatique d'émissions radiophoniques [Gauvain et al., 2002], [Fousek et al., 2008] est une tâche difficile qui ouvre différentes directions de recherche intéressantes liées au traitement et à la reconnaissance de la parole : la segmentation de la parole (parole téléphonique/non téléphonique) et/ou non-parole (musique/-bruits/jingles [13]) [14], l'identification du locuteur [15], etc.), la détection des changements de locuteurs, la détection de la superposition de la parole et de la musique ou de la parole simultanée, etc.

La transcription d'émissions radiophoniques consiste à fournir la transcription textuelle complète d'une émission radiophonique. La transcription fournie peut être enrichie par les informations structurelles suivantes : l'identification des segments de parole et des segments de non parole, l'identification des locuteurs, la liste de mots clés, les titres d'émissions, les sections (nouvelles, sport, météo, etc.).

De nombreuses utilisations de la transcription des émissions radiophoniques sont possibles, telles que (1) l'archivage de ces émissions radiophoniques, (2) la recherche d'information via Internet, (3) la veille technologique pour un client industriel dans une émission radiophonique, où le système de transcription détecte les mots transcrits interessants, les extrait et les communique au client.

La transcription d'**émissions télévisées** [Gauvain et al., 1999], [Zhu et al., 2006], [Allauzen and Gauvain, 2004], [Allauzen and Gauvain, 2002], [Brousseau et al., 2003], [Mareüil et al., 2013] est similaire à la transcription d'émissions radiophoniques. La particularité ici est que le son diffusé par la télévision est de qualité médiocre par rapport à la radio. Cela peut dégrader les performances du système de transcription. Il est donc nécessaire de traiter le son de façon particulière. De plus, dans ce type d'applications la quantité de parole spontanée est plus importante que pour les émissions radiophoniques. Une autre particularité est que dans certains cas, il peut être nécessaire de synchroniser l'image et la transcription textuelle. Pour l'instant, la reconnaissance d'images n'est pas utilisée dans ce type d'applications. A long terme, l'ajout du système de reconnaissance d'images au système de transcription de la parole pourrait augmenter les performances finales de la transcription.

13. Le Jingle est une mélodie courte et accrocheuse servant en général d'annonce musicale, généralement associée à un slogan, accompagnant une publicité, ou à une marque. Les jingles servent aussi, placés entre deux publicités, rubriques d'une émission ou lors de deux interventions d'un animateur, à rappeler l'identité sonore d'une station de radio ou d'une chaîne de télévision.

14. A titre d'exemple, la segmentation parole-musique permet de segmenter le signal en parties correspondant à la parole et en parties correspondant à la musique

15. L'identification du locuteur détecte dans le flux de parole, la personne qui a parlé et quand.

La transcription d'émissions télévisées peut être utilisée pour les mêmes applications que la transcription d'émissions radiophoniques : (1) l'archivage, (2) la recherche d'information ou (3) la veille technologique. Les résultats de la transcription peuvent également être utilisés sous forme de sous-titrage pour aider et assister les spectateurs d'origine étrangère ou les malentendants.

La transcription de **conversations téléphoniques** [Gauvain et al., 2005] [Matsoukas et al., 2006] est aussi une tâche complexe. La principale difficulté de la modélisation linguistique pour ce type parole réside dans la faible quantité de données textuelles d'apprentissage disponible. La transcription de conversations téléphoniques (en les sous-titrant) pourrait être utilisée pour traiter les appels téléphoniques des clients et pour aider les personnes malentendantes à lire les paroles que prononcent leurs interlocuteurs.

Une autre application qui devient de plus en plus demandée est la **transcription des réunions** [Hain et al., 2012]. Les problèmes à affronter ici les différentes conditions d'enregistrement des réunions (où le fond est bruité), les différentes configuration d'enregistrement (par exemple la distance par rapport au microphone qui diffère d'un locuteur à autre ou les différents types de microphones, etc.), la parole est spontanée et souvent simultanée, l'écho, etc.

La transcription est un domaine intéressant dans la communauté scientifiques et industriels. Il faut noter que les Etats-Unis d'Amérique sont très avancé dans ce sujet. Cela pourrait être expliqué par le fait qu'ils ont réussi à trouver les moyens humains et financiers nécessaires à leurs réalisations. De plus, le NIST (*National Institute of Standards and Technology*) et le DARPA (*Defense Advanced Research Projects Agency*) ont lancé plusieurs campagnes d'évaluation, ce qui a permis d'avoir une avancée importante dans ce domaine.

Plusieurs sessions de conférences internationales telles que (*International Conference on Spoken Language Processing* - ICSLP) ou (*European Conference on Speech Communication and Technology* - Eurospeech) traitent le sujet de la transcription. De plus, quelques ateliers de travail (*Workshops*) ont été organisés sur ce sujet tels que le *NIST Speech Transcription Workshop*, le *DARPA Workshop on Automatic Transcription of Broadcast News*, etc.

7 Compagnes d'évaluation de la RAP

Les premières campagnes internationales en RAP étaient organisé aux Etats-Unis d'Amérique vers la fin de l'année 1993, avec les *benchmark tests* proposés par la DARPA (*Defense Advanced Research Projects Agency*), auxquels ont participé plusieurs universités américaines mais aussi des laboratoires provenant du Canada, de France, d'Allemagne ou encore du Royaume-Uni. À cette époque, le but de la tâche proposée était d'améliorer les performances de base des systèmes sur des données considérées comme propres. Les tâches considérées étaient sur la transcription, la détection thématique et la détection automatique d'entités (*Automatic Content Extraction* - ACE). Elle consistait en un décodage de 200 segments issus de 10 locuteurs (20 segments par locuteur) grâce à un système statique (i.e. non-adaptatif) à l'aide d'un vocabulaire fermé commun à tous les participants, afin que tout les systèmes soient comparables entre eux.

Les résultats (en termes de WER) variaient entre 16.8% et 12.2% pour les meilleur systèmes, ce qui représente de bons scores à l'époque. Cependant, il est nécessaire de considérer la simplicité de la tâche, qui est très différente des campagnes qui sont menées par la suite. Une liste non exhaustive des plus importantes compagnes d'évaluation est

présentée dans [Haton et al., 2006]. Par la suite, au fil des années, les corpus de test (ou dits *benchmark tests*) ont évolué vers de la RAP *appliquée*, à savoir des tâches de reconnaissance des journaux ou des nouvelles radiophoniques, qui constituent toujours aujourd'hui la majorité du contenu des campagnes d'évaluation.

1. DARPA/NIST *Wall Street Journal evaluation* **HUB1**, **HUB3** (1993-1995) :cette série d'évaluation des systèmes LVCSR (*Large Vocabulary Continuous Speech Recognition*) utilisait comme base de test des articles lus extraits du *Wall Street Journal*[16].
2. *Broadcast news recognition* **HUB4** (1996-1999) :cette série d'évaluations concernait la transcription automatique d'informations radiodiffusées aux états-Unis.
3. *Conversational telephone recognition* **HUB5**(197-2001) : cette évaluation concernant la transcription automatique de conversations téléphoniques enregistrées essentiellement aux états-unix.
4. *Rich transcription* - ***RT*** (2002-2003) : cette évaluation concernait la transcription automatique de réunions : NIST a organisé des évaluations tendant à produire des transcriptions plus riches et à se focaliser sur des tâches plus ardues telles que la reconnaissance sur de la parole spontanée ou conversationnelle ou encore la transcription de réunions avec de multiples intervenants simultanés. Ces évaluations prennent également en compte le temps de traitement, avec une volonté de se rapprocher du temps réel. Les meilleurs systèmes proches du temps réel obtenaient un score WER de 14.6%, tandis que le meilleur système sans contrainte de temps parvenait à descendre sous la barre des 10%.
5. La campagne d'évaluation *Rich Transcription Fall* **RTF** (2004), organisée aussi par le NIST et qui intégrait de nombreuses tâches de transcription de la parole dans le contexte d'émissions d'information ainsi que des conversations téléphoniques dans différentes langues, a montré le fait qu'une importante chute au niveau des résultats des systèmes de RAP est visible lorsque ceux-ci devaient transcrire de la parole spontanée.
6. NIST *Speaker Recognition Evaluation* **SRE** (2008) ; cette évaluation concernait la reconnaissance du locuteur.

Beaucoup de compagnes d'évaluation internationales à base de corpus standardisé à titre d'exemple GALE[17] (*Global Autonomous Language Exploitation*), Arabic Tree Bank[18] s'intéressent à la langue Arabe standard moderne et à ses dialectes[19].

Concernant la langue française, une première vague de campagnes d'évaluation avait été lancée dans les années 1990.

1. La première évaluation de la RAP était organisée en 1997 avec la campagne ARC (**B1**) et l'Agence Universitaire de la Francophonie (AUF), qui portait sur de la reconnaissance de parole lue (journalistique).
2. En 2003, la première campagne d'évaluation **ESTER** (Évaluation des Systèmes de Transcription enrichie d'Émissions Radiophoniques)[Gravier et al., 2004] etait organisée par l'Association Francophone de la Communication Parlée (AFCP), la Délégation Générale pour l'Armement (DGA) ainsi que l'*Evaluations and Language resources Distribution Agency* (ELDA). Cette campagne reprend le modèle des

16. https ://catalog.ldc.upenn.edu/docs/LDC94S13A/wsj1.txt
17. https ://catalog.ldc.upenn.edu/ldc2016s07, consulté en fev. 2018
18. https ://catalog.ldc.upenn.edu/LDC2016T02, consulté en fev. 2018
19. https ://catalog.ldc.upenn.edu/byyear, consulté en fev. 2018

évaluations NIST américaines. Elle visait à permettre le développement d'un corpus conséquent adapté à la tâche visée ainsi qu'un ensemble de ressources d'évaluation destiné à la communauté scientifique.

La campagne ESTER s'organise autour de trois tâches : la transcription (T), la segmentation (S) et l'extraction d'informations (E). Les tâches T et S constituent le noyau de la campagne tandis que la tâche E regroupe des thèmes prospectifs. Bien que ces tâches ne soient pas indépendantes, les tâches sont évaluées séparèrent avec une métrique propre. La tâche de transcription (T) consiste à produire une transcription à partir du signal audio. Les transcriptions sont évaluées en termes de mots erronés (*Word Error Rate*) calculés à partir des transcriptions de référence après normalisation (même notation des chiffres, majuscule ou minuscule, abréviations, etc.).

Les résultats de la première phase ESTER montrèrent des scores WER à presque 40% pour la plupart des participants. Cela permit de mettre en place une procédure et des conditions d'évaluation clairement définies pour la seconde phase, qui se déroula en janvier 2005 et dont le corpus de test était constitué de dix heures d'audio provenant de six stations de radio. Les résultats à l'issue de cette seconde phase montrèrent des scores très bons, le meilleur système obtenant un WER global de 11.8% sur les six stations évaluées.

3. En janvier 2008, les trois mêmes organismes (AFCP, DGA et ELDA) ont organisé une seconde campagne nommée **ESTER**2 [Galliano et al., 2009] avec pour finalité la mesure des progrès réalisés dans le domaine de la RAP depuis ESTER1 ainsi que le lancement de nouveaux axes de recherche et la production de nouvelles ressources, notamment un corpus [Galliano et al., 2006] annoté de taille conséquente. Les tâches évaluées au niveau de ESTER2 concernaient des tâches de segmentation, de transcription et d'extraction d'informations. Le meilleur score sur la tâche de transcription était de 12.1%.

4. Il est à noter que le projet **EPAC** [20][Esteve et al., 2010](Exploration de masse de documents audio pour l'extraction et le traitement de la PArole Conversationnelle), financé par l'ANR (Agence Nationale de la Recherche Française), de janvier 2007 à août 2010, poursuit ce travail sur la parole spontanée. L'objectif principal de ce projet est d'améliorer les systèmes de RAP sur ce type de parole, et de fournir des informations supplémentaires au niveau des transcriptions automatiques en sortie de ces systèmes (nommer les locuteurs, définir le genre de l'émission, réaliser un découpage syntaxique des segments, etc.). Le corpus, qui est réalisé dans le cadre du projet EPAC, se compose de transcriptions manuelles de 100 heures d'enregistrement audio. enregistrées entre 1998 et 2004 et provenant de six radios : France Inter, France Info, RFI, RTM, France Culture et Radio Classique. Ces transcriptions ont été annotées en partie grâce à une transcription assistée [21], le reste ayant été fait entièrement manuellement. Les enregistrements audio proviennent des 1500 heures d'audio brut diffusées aux participants de la campagne ESTER1. Finalement, les sorties automatiques produites par les différents outils des partenaires du projet EPAC pour l'ensemble des 1500 heures d'audio brut de ESTER1 viennent s'ajouter à ces transcriptions manuelles [Dufour, 2010]. La plupart des émissions contiennent de la parole préparée (reportages informations). Des articles du jour-

20. http ://epac.univ-lemans.fr/, consulté en mars 2016.

21. Les transcriptions dites assistées sont obtenues en deux étapes : une première transcription est obtenue automatiquement au moyen d'un système de RAP, puis une correction est opérée manuellement.

nal *le Monde* de 1987 à 2003 peuvent être utilisés en plus du corpus transcrit pour apprendre le modèle de langage.

5. La campagne d'évaluation **ESTER**2 reprend le corpus qui était déjà fourni pour la campagne ESTER1, et est étendu pour permettre de couvrir de nouveaux types de données. En particulier, ESTER2 inclut plus d'émissions mettant en jeu des locuteurs avec des accents étrangers, et des émissions de parole spontanée. En complément des émissions des six radios françaises, la campagne inclut des débats et des programmes provenant de la radio Africaine Radio Africa No1. Les données ajoutées par la campagne ESTER2 sont les suivantes :
   — 100 heures d'émissions radiophoniques transcrites manuellement, enregistrées entre 1998 et 2006,
   — 6 heures pour le développement et 6 heures pour le test enregistrées en 2007 et 2008,
   — 40 heures d'émissions radiophoniques africaines avec transcriptions manuelles rapides. Les ressources textuelles sont elles aussi étendues avec des articles du journal *le Monde* de $2004 - 2006$ (en complément des articles de 1987 à 2003).

   Il est à noter que le corpus de développement de ESTER2 contient plus de parole dégradée que le corpus de test.

6. Enfin, le projet **ETAPE** (Évaluations en Traitement Automatique de la Parole) [Gravier et al., 2012] [Galibert et al., 2014] a pour objectif de mesurer les performances des technologies vocales appliquées à l'analyse des flux télévisés en langue française. Le projet s'inscrit dans la continuité des campagnes ESTER tout en élargissant les enjeux scientifiques, en particulier, à la parole spontanée, la parole superposée et à la diversité des contenus. Au-delà de la mesure des performances, le projet a vocation à faire progresser le domaine du traitement automatique de la parole par la mise à disposition d'un corpus annoté, par la formalisation de nouveaux problèmes et par son rôle de structuration de la communauté scientifique. Enfin, l'enrichissement du corpus par des annotations phonétiques et syntaxiques vise à l'émergence de nouveaux domaines de recherche interdisciplinaires. En général, la transcription automatique d'émissions radiophoniques est un problème difficile et qui ouvre différentes directions de recherche intéressantes : segmentation de la parole (parole téléphonique/non téléphonique, parole/musique/bruits), détection des changements de locuteurs, détection de la superposition parole et musique, de la parole simultanée, de la parole bruitée, etc. Lorsque l'on traite les émissions radiophoniques, différents styles de parole peuvent apparaître. Il est ainsi possible de rencontrer de la parole proche d'un texte lu (type présentation d'un journal), ou, au contraire, de la parole plus spontanée (lors de débats ou d'interviews). L'objectif est de proposer des méthodes améliorant la RAP sur la parole spontanée, sans dégrader les performances sur la parole préparée [22].

La figure **??** illustre les importantes tâches de la RAP traitées pendant ces dernières années.

Dans cette thèse, le corpus textuel utilisé dans la compagne ETAPE est étendue et utilisé dans nos expérimentations pour la modélisation de langage et la sélection des données textuelles pour une meilleure modélisation.

22. http ://www.afcp-parole.org/etape.html, consulté en 12 Mars 2016

8 Plateformes pour les Systèmes de LVCSR

Il existe différentes plateformes pour la modélisation acoustique de la parole :

1. la plate-forme **HTK** [23] (Hidden Markov Model Tool Kit) [Young et al., 1997] [Young et al., 2009] est un ensemble de modules et d'outils portable, écrits en langage C, permettant la création et la manipulation de modèles de Markov cachés. HTK dispose d'une bonne documentation.
2. **Julius** [25] est une plate-forme de haute performance, qui utilise deux passes lors du décodage. Elle est basée sur des modèles de langage 3-grammes dépendant du contexte HMM. Elle peut effectuer presque en temps réel le décodage de 20k mots dictés, sur la plupart des PC actuels. Des formats standards sont adoptés pour une interaction avec les autres boîtes à outils de modélisation libre. Julius est distribué avec une licence *open To-GETHER* avec des codes source. Elle a été utilisée par beaucoup de chercheurs et de développeurs au Japon [Lee et al., 2001].
3. La **boite à outils RWTH** [26] [Rybach et al., 2009] comprend l'état de l'art de la reconnaissance vocale pour les modèles acoustiques, le décodage, les méthodes d'adaptation, une bibliothèque des automates à états finis, et un arbre de recherche décodeur efficace, etc. Il est fournit avec une ample documentation, des exemples et un tutoriel pour les nouveaux utilisateurs.
4. le système **CMU-Sphinx** est un projet lancé principalement par l'université Carnegie Mellon. Il peut être utilisé comme base pour un système de la RAP. Il existe quatre familles de décodeurs Sphinx, disponibles sous une licence permettant l'utilisation commerciale. Sphinx 2 [Huang et al., 1993], Sphinx 3 [Ravishankar et al., 2000], Sphinx 4 [Walker et al., 2004] (écrit complètement en langage Java), SphinxTrain [27] pour entraîner les modèles acoustiques (HMMs) et Pocket Sphinx [Huggins-Daines et al., 2006].
5. **Kaldi** [28] est une boite à outils écrite en C++ contenant des librairies, des programmes et des scripts destinés pour la RAP. Kaldi utilise l'algorithme de *Viterbi* pour l'entraînement des modèles acoustiques. Pour le cas de l'estimation discriminante de l'adaptation de locuteur, il peut utiliser l'algorithme étendu de *Baum-Welch* [Povey et al., 2011]. La librairie de Kaldi est basée sur la librairie *OpenFST* [Allauzen et al., 2007] et utilise les librairies d'algèbre linéaire comme BLAS [29] (*Basic Linear Algebra Subprograms*) et LAPACK [30] (*Linear Algebra PACKage*).

Des études comparatives des différentes librairies peuvent être consultées [31] [Hamza and Halima], [Gaida et al., 2014], etc.

Selon la littérature, l'outil Kaldi [32] fournit des performances plus avancées. L'outil Sphinx aussi permet de bons résultats en peu de temps. La plateforme HTK fournit des résultats inférieurs et nécessite beaucoup plus de temps. En Septembre 2015, HTK 3.5 a introduit de nouvelles fonctionnalités telles que les réseaux de neurones pour les modèles

23. [24]
25. http ://julius.sourceforge.jp
26. http ://www-i6.informatik.rwth-aachen.de/rwth-asr/
27. http ://cmusphinx.sourceforge.net/html/cmusphinx.php
28. https ://sourceforge.net/projects/kaldi/
29. www.netlib.org/blas/
30. www.netlib.org/lapack/
31. http ://thegrandjanitor.com/tag/kaldi/
32. Projects :2015s1-06 Performance Evaluation of KALDI

acoustiques, RNNLMs pour les modèles de langage, etc. Donc, la concurrence est encore en cours.

9 Systèmes populaires de la RAP

Plusieurs groupes et laboratoires de recherche proposent des systèmes de la RAP à grand vocabulaire, qui sont fonctionnels et compétitives lors des compagnes d'évaluation pour la transcription de la parole avec des décodeurs d'une ou de plusieures passes. Parmi ces systèmes, on peut citer ceux du laboratoire de recherche : *Philips GmbH* de l'université de Aachen- Allemagne [Thelen et al., 1997], le Laboratoires de recherches AT&T Florham Park NJ-USA [Ljolje et al., 1999] [Mohri and Riley, 2001], le groupe d'IBM Watson pour la parole conversationnelle [33], Le laboratoire d'informatique de l'université du Maine (LIUM), le groupe traitement du langage parlé *Spoken language processing* de LIMSI-CNRS Paris-France [Gauvain et al., 2002], le laboratoire lorrain de recherche en informatique et ses applications (LORIA), etc.

Une brève description est présentée pour les trois derniers systèmes. Lors de notre travail, une partie importante des expérimentations sont réalisé avec le système de transcription de LORIA.

9.1 Système de transcription de LIUM

Les paramètres acoustiques utilisés par le système de transcription du Laboratoire d'Informatique de l'Université du Maine (LIUM) sont au nombre de 39 : il s'agit de descripteurs issus d'une analyse du signal de type PLP et d'un descripteur de l'énergie, ainsi que les dérivées premières et secondes de ces descripteurs.

Les modèles acoustiques sont appris sur les $80h$ d'ESTER1 et sur une partie des données d'ESTER2, complétés par $40h$ d'émissions radiophoniques provenant du projet EPAC. Ce corpus est composé d'environ $191h$ de données bande large (BL) et de $40h$ de données bande étroite (BE). Les modèles BL sont appris avec les $191h$ de données large bande puis adaptées avec la méthode MAP pour chacun des genres (homme/femme). Les modèles BE sont appris sur l'ensemble des données ($BL + BE = 231h$), puis adaptés aux genres à l'aide de la méthode MAP. Après l'obtention de ces modèles (BL+BE pour homme/femme), une adaptation aux données de BE est réalisée, encore une fois à l'aide de MAP.

Les phonétisations des mots composant le vocabulaire du système ont été obtenues en utilisant le dictionnaire de phonétisations BDLEX. Pour les mots inexistants dans le dictionnaire BDLEX, le système de phonétisation automatique à base de règles LIA-PHON [Béchet, 2001] a été utilisé.

Pour construire le vocabulaire, un modèle unigramme résultant de l'interpolation linéaire des modèles unigrammes des sources de données disponibles (dans ESTER 2) est réalisé. Les coefficients mis en jeux dans l'interpolation linéaire ont été optimisés sur le corpus de développement d'ESTER 2. Les 122000 mots les plus probables ont ensuite été extraits du modèle de langage interpolé. Ces mots constituent le vocabulaire du Système de LIUM. L'adaptation MAP est utilisée pour adapter les modèles acoustiques au type de bande passante (large/étroite) ainsi qu'au genre des locuteurs (homme/femme).

Le système de la RAP du LIUM est composé de plusieurs passes de décodage. Les modèles acoustiques de la première passe sont composés de 6500 états partagés, chaque

33. https ://developer.ibm.com/watson/

état étant modélisé par une mixture de 22 gaussiennes. Les modèles utilisés lors des passes deux et trois sont composés de 7500 états partagés, toujours modélisés par une mixture de 22 gaussiennes. Ces modèles ont été estimés grâce à un apprentissage de type SAT (*Speaker Adaptive Training*) [Anastasakos et al., 1997] combiné à un apprentissage discriminant de type MPE (*Minimum Phone Error*). Une matrice de transformation *Constrained Maximum Likelihood Linear Regression* (CMLLR) est calculée pour chaque locuteur et appliquée sur les paramètres acoustiques de chacun des locuteurs respectifs.

Les modèles de langage ont été appris à l'aide du toolkit SRILM [Stolcke, 2002], et en utilisant la technique de lissage dite de Kneser-Ney modifié [Chen and Goodman, 1999], [Kneser and Ney, 1995] avec interpolation des n-grams d'ordres inférieurs. Le modèle de langage utilisé pour le décodage à grand vocabulaire, comprend $121K$ 1-grams, $29M$ de 2-grams, 162M de 3-gram et 376M de 4-grams.

Ce système de transcription présenté lors de la campagne ESTER2 a été le meilleur système libre de la RAP, avec 24.2 % de WER sur le corpus de développement et 17.8% sur le corpus de test. Le corpus de développement contient plus de parole dégradée que le corpus de test, ce qui explique la différence de WER entre les résultats obtenus sur ces deux corpus [Laurent, 2010].

9.2 Système de la transcription de LIMSI

Le système de transcription d'émissions radio ou télédiffusées (*Broadcast News transcription*) [Gauvain et al., 2002] du Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI) de l'université Paris-Sud repose sur deux composants principaux : un segmenteur audio et un décodeur lexical. Le décodeur utilise des modèles de Markov cachés avec densités de probabilité continues (sommes pondérées de gaussiennes) pour les modèles acoustiques, et des statistiques n-grammes obtenus sur de grands corpus de textes pour les modèles linguistiques. Les modèles de Markov cachés représentent des allophones contextuels avec une structure gauche-droite à états liés. Ils modélisent des séquences de trames centisecondes avec 39 composantes, 12 coefficients cepstraux (PLP) et le logarithme de l'énergie à court-terme, avec leurs dérivées d'ordre 1 et 2.

Le décodage en mots est effectué en trois passes. La première passe produit une hypothèse qui est utilisée pour réaliser l'adaptation MLLR non supervisée des modèles acoustiques. Les modèles adaptés sont utilisés dans la seconde passe pour générer un graphe de mots. Ces deux passes utilisent un modèle de langage trigramme. L'hypothèse finale est générée avec un modèle de langage quadrigramme et les modèles acoustiques adaptés lors de la seconde passe. La première passe utilise un jeu d'allophones représentant environ 5500 contextes avec 6300 états liés. Les passes 2 et 3 utilisent des modèles plus gros, représentant 11000 contextes phonétiques et 11700 états liés, avec respectivement 16 et 32 gaussiennes par état. L'ensemble du décodage est effectué en moins de 10 fois le temps réel. Le regroupement des états est réalisé en créant un arbre de décisions pour chaque état de chaque phonème de façon à maximiser la vraisemblance des données d'apprentissage pénalisée par le nombre d'états liés.

9.3 Système de la transcription de LORIA

Lors du projet Technolangue EVALDA-ESTER pour l'évaluation des systèmes de transcription automatique des émissions radiophoniques francophones, l'équipe Parole (renommée actulement *Multispeech*) du laboratoire Lorrain de Recherche en Informatique

et ses Applications a développé le système ANTS [34] (*Automatic News Transcription System*). Les détailes de ce système peuvent être consulté dans [Brun et al., 2004].

Un système plus avancé de transcription des émissions radiophoniques est réalisé comme système de base au niveau de LORIA [Jouvet and Fohr, 2013] et [Jouvet and Fohr, 2014]. Ce système de transcription utilise une étape de diarisation, l'outil CmuSphinx [35], la plateforme HTK [36] et le décodeur Julius [37]. La diarisation [Jun, 2012] est le processus de partitionnement d'un flux audio d'entrée en segments homogènes en fonction de l'identité du locuteur. Dans ce cas, pour chaque segment est associé la qualité de parole identifiée automatiquement, le genre du locuteur et son identité. Donc les modèles acoustiques appris sont spécifiques au genre des locuteurs (homme/femme) et la qualité de la parole (studio vs. téléphone).

Les paramètres acoustiques utilisés dans ce système sont au nombre de 39 : il s'agit de descripteurs issus d'une analyse du signal de type HTK MFCC, un descripteur de l'énergie et les dérivées premières et secondes de ces descripteurs. Les unités acoustiques utilisées dans les modèles acoustiques sont des phonèmes à contexte dépendant. Le système de base a 7500 densités partagées (*senones*), chacune d'eux est composée de 64 gaussienne.

Les phonétisations des mots composant le vocabulaire sont obtenues en utilisant le dictionnaire de phonétisations BDLEX et le lexique *in-house pronunciation*. Pour les mots restants, les variantes pronociations sont obtenues automatiquement par les convertisseurs *JMM-based* et *CRF-based Grapheme-to-Phoneme* [Illina et al., 2011].

Les modèles de langage utilisées sont des 3-grammes appris à l'aide du toolkit SRILM [Stolcke, 2002] sur un énorme corpus textuel en utilisant la technique de lissage dite de Kneser-Ney modifié [Chen and Goodman, 1999], [Kneser and Ney, 1995].

Pour la phase de reconnaissance à multi-passes, la première passe de décodage effectue un décodage pour chaque segment audio en utilisant le modèle acoustique le plus adéquat (selon la qualité de la parole estimé et le genre des locuteurs (homme/femme)). La seconde passe de décodage est basée sur l'adaptation non-supervisée VTLN des paramètres acoustiques et l'adaptation MLLR pour les modèles acoustiques. D'autres types d'adaptation peuvent être utilisées.

L'évaluation de la performance du système de la RAP sur les données de développement et de test sont analysées par la boîte à outils sclite [38]. Cet outil de notation et d'évaluation du rendement des systèmes de la RAP fait partie du NIST SCTK Scoring Toolkit.

10 Conclusion

Nous avons consacré ce chapitre à la description des différentes composants d'un système de reconnaissance de la parole, allant de la paramétrisation du signal acoustique jusqu'à son décodage, en évoquant toutes les nouvelles techniques d'adaptation relatives. Nous nous sommes contentés de présenter les systèmes destiné à la parole continue fondés sur l'approche probabiliste (Modèle de Markov Caché avec densité d'observation à mélanges de gaussiennes - HMM-GMM) puisque le système de la transcription de LORIA que nous avons utilisé dans une importante partie de nos expérimentations est de ce type.

34. https ://hal.archives-ouvertes.fr/inria-00107762/document
35. http ://cmusphinx.sourceforge.net/
36. htk.eng.cam.ac.uk/
37. http ://julius.osdn.jp/en_index.php
38. www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

Malgré que la modélisation statistique du langage est une phase primordiale dans un système de la RAP, nous nous sommes contenté dans ce présent chapitre d'introduire brièvement les modèles de langage. Le prochain chapitre traitera en détails ces concepts : les modèles de langage, leurs lissages, leurs types, leurs évaluations, etc.

Chapitre 4

Modèles de Langage

1 Introduction

En général, la seule information acoustique ne suffit pas pour transcrire correctement les suites de mots dans un système de reconnaissance à large vocabulaire. L'information linguistique est aussi considérée dans des applications liées au traitement du langage naturel et plus particulièrement à la reconnaissance automatique de la parole, la traduction automatique, l'identification des langues, etc. Son rôle est de restituer des contraintes linguistiques en guidant la transcription de la parole grâce à l'information quantifiée qu'ils apportent sur la validité ou non d'une hypothèse.

La modélisation acoustique seule permet de réaliser la transcription phonétique d'une phrase. En absence de contraintes d'ordre linguistique, qu'elles soient lexicales, syntaxiques ou sémantiques, il est probable que la suite de phonèmes ainsi obtenue n'aura qu'un lointaine rapport avec la chaîne attendue. Cela est dû à la difficulté de l'identification phonétique dans la parole continue en raison des phénomènes de coarticulation. L'auditeur humain exploite les niveaux supérieurs pour interpréter l'ambiguïté phonétique intrinsèque de nombreux contextes et même compenser une certaine dégradation limitée de l'information acoustique. Mariani [Mariani, 2002] repporte qu'une suite particulière de 9 phonèmes peut être transcrite de Français en 32000 suites de mots différentes orthographiquement correcte.

La modélisation du langage a pour objectif de résumer les contraintes linguistiques liées à une langue naturelle. La modélisation du langage peut être faite par des grammaires probabilistes en se basant sur des règles de production du langage [Cormons, 2014]. Les grammaires [Baker, 1979] engendrent généralement une réponse de type «oui/non» pour l'acceptation ou le refus d'une hypothèse. Aussi, leurs rédactions n'est pas aussi facile, et leur taille augmente avec la taille du vocabulaire. Cependant, la modélisation statistique de la langue, qui est une deuxième alternative, est considérée comme le résultat d'une estimation de modélisation non supervisée sur un corpus d'apprentissage textuel. L'exemple le plus fréquent des modèles statistiques est le modèle à base de n-gramme [Rosenfeld, 2000]. Dans le contexte de la modélisation statistique (dit aussi probabiliste) [Chen, 1996], les phrases possibles dans la langue sont approchées par des distributions probabilistes sous forme de chaînes markoviennes. Les modèles de langage statistiques n'exigent pas des restrictions de type grammaticales, ce qui leur permet d'être souples et d'accepter toute construction syntaxique pouvant apparaître dans le langage parlé. Une troisième alternative aussi de type statistique, à base de réseaux de neurones est utilisée [Bengio et al., 2003] pour la modèlisation du langage.

Dans ce chapitre nous introduisons et nous exploitons la modélisation du langage dans

le contexte de la reconnaissance automatique de la parole par les modèles de langage de type n-gramme et les modèles de langage neuronaux.

2 Traitement du Langage Naturel et la Modélisation du Langage

L'informatique a engendré la problématique de produire de manière automatique des tâches de communication autrefois réalisées seulement par l'homme en se basant sur le *traitement de langues naturelles*. Le terme *naturel* s'oppose alors aux langues dites *formelles* ou *artificielles*. Cependant, si la linguistique est l'étude des langues naturelles et précisément du langage humain, le Traitement Automatique des Langues Naturelles (TALN) (*Naturel Langage Processing* - NLP) en est l'application informatique. Cette discipline étudie en effet un ensemble d'approches ou de techniques permettant de modéliser, d'analyser et d'interpréter le langage humain. Le TALN prend en considération les données textuelles. Ces textes ne sont en effet pas réduits à une succession de mots mais sont considérés d'un point de vue linguistique [Béchet, 2009].

Le TALN utilise des méthodes stochastiques, probabilistes ou statistiques pour résoudre certaines difficultés, en particulièr celles qui surviennent du fait que les phrases très longues sont fortement ambiguës, une fois elles sont traitées avec des grammaires réalistes. Le TALN statistique comporte toutes les approches quantitatives du traitement linguistique automatisé. Par héritage, le TALN utilise les connaissances linguistiques (phonologiques, morphologiques, sémantiques et syntaxiques) en s'intéressant à différentes tâches comme la reconnaissance automatique de la parole, la traduction automatique, le résumé automatique de textes, la synthèse de la parole, la classification et la catégorisation automatique de documents textuels, la recherche documentaire, la recherche d'information, la fouille de textes, la reconnaissance d'entités nommées (les noms propres, tels que des personnes ou des endroits), l'annotation sémantique, etc.

Dans le cadre du TALN, différents types de descripteurs traditionnels sont utilisés pour les traitements des textes tels que le mot ou le lemme, la forme fléchie, le radical, les descripteurs phonétiques, les n-grammes, etc. [Béchet, 2009]. Un des descripteurs permettant de lever en partie les ambiguïtés sémantiques, on trouve les n-grammes qui sont également utilisés pour la modélisation du langage dans un système de reconnaissance automatique de la parole (RAP). Malgré que, la modélisation statistique du langage a été fortement critiquée par les linguistes dès ses premières utilisations. La déclaration de Chomsky : "la notion de Probabilité d'une phrase est complètement inutile" [1] est considérée aujourd'hui erronée dû au succès incontestable des applications qui impliquent des modèles statistiques à base de n-gramme. Aussi, la modélisation statistique s'est étendue vers les techniques d'apprentissage automatique comme les réseaux de neurones. Aujourd'hui, une tendance importante en TALN, qui concentre les efforts de nombreuses équipes de recherche [Collobert and Weston, 2008] [Goldberg, 2015] dans ce sens.

1. In 1969, Chomsky said in regards to statistical language models : "*[...] probability of a sentence" is an entirely useless [concept], under any known interpretation of this term*". see : http ://-norvig.com/chomsky.html

3 Ressources pour les Modèles de Langage

Dans le cadre de la reconnaissance automatique de la parole, l'estimation des modèles de langage statistiques est faite à partir de corpus textuels en se basant sur un vocabulaire.

3.1 Corpus de textes

Les données textuelles d'apprentissage des modèles de langage proviennent en grande partie de sources écrites, mais également de parole transcrite. Dans le cadre de la reconnaissance automatique de la parole spontanée, l'apprentissage des modèles statistiques du langage est réalisé principalement avec des corpora textuels collecté sur des données journalistiques et/ou des transcriptions manuelles de données audio directement liées à l'application visée. Cependant, de telles données ne sont pas disponibles en grandes quantités et elles sont très coûteuses à collecter.

Le *World Wide Web* est devenu une des plus importantes sources d'information disponible de manière électronique. Le Web constitue ainsi un corpus gratuit, riche, énorme et accessible pour de nombreuses langues.

Pour la modélisation statistique du langage, on distingue trois corpora textuels disjoints nécessaires [Haton et al., 2006] :

— Le corpus d'apprentissage : sachant que l'apprentissage des modèles de langage est basé sur le fait que, des suites de mots très fréquentes dans la langue auront une probabilité d'apparition élevée, alors que des séquences a priori très peu probables resteront néanmoins possibles avec une faible probabilité. Il est donc nécessaire que le corpus de texte utilisé doit contenir une quantité assez importante de données textuelles pouvant servir à un apprentissage consistant des modèles de langage. Dans certains travaux, ce corpus peut contenir jusqu'à quelques milliards de mots.
— Le corpus de développement : sert à son tour d'affiner les paramètres du modèle de langage.
— Le corpus de test : sert à évaluer le modèle de langage face à de nouvelles données.

Prétraitement du corpus textuel

Concernant le recueil de données textuelles en grande quantité pour la construction des systèmes de reconnaissance automatique de la parole, une approche intéressante consiste à télécharger un grand nombre de sites Web dans la langue donnée et à filtrer les données brutes récupérées pour les rendre exploitables, comme dans le travail proposé dans [Vaufreydaz, 2002b], [Seng et al., 2010] et [Oger, 2011]. Ces données textuelles peuvent servir d'une part à calculer des modèles statistiques du langage, et d'autre part à obtenir un corpus pouvant ensuite être prononcé par des locuteurs en vue de la constitution d'une base de signaux acoustiques conséquente.

Il est nécessaire de rappeler que, l'encodage pour une langue quelconque permet de coder un caractère. La nécessité d'utiliser une représentation interne unique pour les données textuelles s'est posée. Le consortium Unicode [2] a proposé une norme de codage qui spécifie un numéro unique pour chaque caractère, quelle que soit la plate-forme, le logiciel et la langue. L'UTF-8 [3] est le format de transformation des caractères Unicode en ASCII le plus commun pour les applications liées à l'Internet.

2. http ://www.unicode.org
3. Unicode Transformation Format : http ://www.ietf.org/rfc/rfc2279.txt

Afin de pouvoir construire un corpus de texte utile pour la modélisation du langage, quelques pré-traitements sont nécessaires sur les pages html récupérées (ou les textes brutes en général), selon le schéma suivant :

— transformation des pages html vers du texte ;
— normalisation des tags et restructuration des documents ;
— conversion des encodages (passage au Unicode UTF-8) ;
— séparation en phrases puis en unités lexicales (mots) et/ou unités sous-lexicales (selon la langue) ;
— transcription des caractères spéciaux et des nombres ;
— conversion de la casse du caractère (des majuscules vers des minuscules) ;
— suppression de la ponctuation ;
— filtrage en fonction d'un vocabulaire donné.

Des outils automatiques pour la récupération depuis le web et le traitement d'un corpus de texte peuvent être construits spécifiquement pour chaque langue ou chaque tâche mais ils sont coûteux en temps de développement. Cependant, de simples scripts perl ou python peuvent suffire pour cette tâche.

3.2 Vocabulaire

Le vocabulaire est un paramètre important d'un modèle de langage. Il constitue une liste fermé (close) de mots ou d'unités lexicales simples et composés pouvant être reconnus par le système de reconnaissance. Les systèmes de transcription actuels sont dédiés à des tâches à grands et très grands vocabulaires dans un contexte de dialogue interactif. La taille du vocabulaire et la sélection des unités lexicales dans le vocabulaire influencent fortement les performances du système de transcription automatique.

Le problème majeur dans les systèmes de la RAP est le vocabulaire fermé. En effet, que ce soit acoustiquement ou dans le cadre de modélisation du langage, le nombre de mots est fixe. Dans ce cas, si l'utilisateur prononce un mot qui n'existe pas au sein du vocabulaire, comme des noms propres ou des noms communs non inclus dans la tâche, il est alors substitué à une séquence de mots proche acoustiquement et probable selon le modèle de langage. Cela peut conduire à une ou plusieurs erreurs consécutives influençant la reconnaissance des mots suivants.

Cependant, on peut recueillir un vocabulaire en le récupérant à partir de ressources lexicales existantes pour assurer une bonne couverture lexicale et maximiser l'espérance de l'importance du vocabulaire considéré. Avant 1999, les ressources destinées au traitement automatique du français étaient toutes des ressources payantes, principalement distribuées par ELRA (*European Language Resources Association*) [4]. Depuis, des ressources distribuées gratuitement commencent à apparaître : le lexique de l'*ABU* (1999), la première version de *Lefff* (Clément et al., 2004) et *Morphalou* (Romary et al., 2004). ABU, le plus ancien des lexiques morphosyntaxiques distribué librement sur le Web, comporte environ 300 000 formes et 60 000 lemmes (ou formes de citation). Lefff et Morphalou comptent quant à eux respectivement 500 000 et 525 000 entrées. À côté des ressources lexicales développées par les linguistes informaticiens, il existe *Lexique* et *Brulex* (Content et al., 1990) qui sont des ressources gratuites fournissant des transcriptions phonémiques et un découpage syllabique.

Il existe d'autres ressources avec une couverture plus complète, contenant ces mêmes informations issues des laboratoires de recherche publiques mais qui sont payantes et

4. http ://catalog.elra.info/

sous licences non libres. L'une des plus anciennes et des plus connues est la ressource *BDLex* [Pérennou, 1998]. Citons également le lexique morphosyntaxique *Multext* (Ide et Véronis, 1994), le dictionnaire DELAF [5], *ILPho* (Boula De Mareuil et al., 2000) et *GlobalPhone* (Schultz et al., 2013). Ces dictionnaires ont été produits automatiquement par des outils entraînés sur des transcriptions d'enregistrements de locuteurs et les textes correspondants, puis corrigés manuellement.

Ces dernières années, on connait l'apparition de ressources volumineuses non validables manuellement telles que *WOLF* (Sagot et Fišer, 2008) ou *BabelNet* (Navigli et Ponzetto, 2010) et le lexique *Glàff* [Sajous et al., 2013] [Sajous et al., 2014] construit à partir du dictionnaire libre Wiktionnaire [6]. Ces ressources résultent souvent de l'agrégation, de la traduction automatique d'autres ressources et de l'application de règles d'inférence.

Une solution possible est de générer le vocabulaire automatiquement à partir de données textuelles disponibles dans une langue considérée. La méthode peut être réalisée par un outil qui estime la densité des unités lexicales, qui revient a calculer ou estimer l'occurrence d'une unité lexicale dans un (ou plusieurs grands) corpus textuel. Donc, le nombre d'entrées et les entrées lexicales du vocabulaire sont choisies empiriquement, selon la couverture désirée. Par exemple, on sélectionne simplement les plus fréquentes entrées observées sur un corpus de texte. Si on a plusieurs corpus de texte, une méthode de détermination de combinaison optimale des vocabulaires produits par les différents corpus peut être considérée.

La taille du vocabulaire influence fortement les performances du système de la RAP. En effet, si on augmente la taille du vocabulaire, la taille du modèle de langage et l'espace de recherche du système de reconnaissance croient proportionnellement. De plus, on aura besoin de plus de données textuelles pour entraîner les modèles de langage. Cependant, si la taille du vocabulaire est petite, le taux de mots non reconnus (ou dites mots hors-vocabulaire) du système augmente.

Mots hors vocabulaire

Un problème souvent rencontré, même avec de grand corpora textuel, est que certains n-gramme qui n'apparaissent pas dans l'ensemble d'apprentissage n'auront pas la chance d'apparaître dans le modèle de langage, et de ce fait de ne jamais contribuer dans la transcription. Sachant que tout les mots inconnus (dit hors-vocabulaire ou OOV (*Out Of Vocabulary*) ou UNK (*UNKnown word*)) ne peuvent pas être reconnus par le système. Rappelons que, si l'utilisateur prononce un mot qui n'existe pas au sein du système, comme des noms propres ou des noms communs non inclus dans la tâche, il est alors substitué à une séquence de mots proches acoustiquement (selon les modèles acoustiques) et fortement probables linguistiquement (selon le modèle de langage).

Diverses approches sont proposées ces dernières années, pour prendre en considération les noms propres dans le vocabulaire. Dans [Tangigaki et al., 2000], des modèles hiérarchiques génériques sont utilisés pour la gestion des noms propres en japonais. Cette technique ne peut être réalisable en français car les noms propres ne présentent pas une structure figée dans cette langue. Des travaux sont menés dans ce contexte au niveau de LORIA pour le français [Shaik et al., 2012] [Sheikh et al., 2015] pour améliorer les vocabulaire et les systèmes de RAP.

5. http ://www-igm.univmlv.fr/ laporte/serveur/Dictionnaires/Delaf.html
6. https ://fr.wiktionary.org/wiki/dictionnaire

Combinaison des données pour les modèles de langage

Pour développer des modèles de langage pertinents pour un système de RAP dédié à une tâche cible, les données textuelles proches ou adéquates à ce type de tâches ne sont pas toujours faciles à acquérir. Dans notre cas, afin de faciliter la *transcription d'émissions radiophoniques*, il est impossible de se contenter de développer les modèles de langage avec seulement des données spécifiques au domaine. L'utilisation d'autres sources de données disponibles même de diverses sources (pas trop proches à la tâche cible) pour enrichir les modèles de langage est envisageable.

La combinaison des sources d'information est une solution pour la modélisation du langage lorsque l'on disposent de plusieurs corpus textuels d'apprentissage plus au moins adaptés à la tâche spécifique. En combinant les différentes sources de données, les plus pertinentes ont un poids plus élevé dans le modèle de langage final. Il existe deux principales approches pour réaliser cette opération. La première consiste à estimer un modèle de langage indépendant pour chaque source d'information et de les combiner, pour obtenir le modèle de langage final. Généralement, la combinaison la plus simple à utiliser est l'*interpolation linéaire*. Si l'on considere un ensemble de M modèles de langage représentés par leur distribution p_m avec $m = 1...M$, le modèle de langage final interpolé est définit comme :

$$p_{interp}(w_i \mid h_i^n) = \sum_{m=1}^{M} \lambda_m p_m(w_i \mid h_i^n) \tag{4.1}$$

avec $\sum_{m=1}^{M} \lambda_m = 1$ Les coefficients λ_m (dits : poids) servent à pondérer l'importance de chaque modèle individuel dans le modèle final. L'estimation de ces coefficients est faite par l'algorithme expectation-maximisation (EM) [Dempster et al., 1977] [7] afin de minimiser la perplexité du modèle sur un corpus textuel de développement.

La seconde approche consiste à estimer un modèle directement à partir des diverses sources d'information en les représentant sous la forme de contraintes. Le modèle prenant en compte toutes ces informations est alors optimisé pour maximiser un critère qui varie souvent. Ce critère peut être le *maximum d'entropie* ou le minimum d'information discriminante (*Minimum Discrimination Information*- MDI) proposé par Della Pietra et al. (1992) ou encore le maximum de vraisemblance.

La combinaison à base du maximum d'entropie été proposé par Rosenfeld (1994) et selon l'auteur elle assure des résultats plus intéressants que ceux d'une simple interpolation linéaire. Cependant, les expérimentations réalisées dans [Goodman, 2001] aboutissent à des conclusions différentes. Aussi l'utilisation de cette méthode pour un large corpus d'apprentissage est une tâche particulièrement compliquée et lente [Chen and Goodman, 1999]. Une extension de cette idée est présentée dans [Alumäe and Kurimo, 2010].

Adaptation des modèles de langage

Un modèle de langage spécifique à un domaine est plus efficace qu'un modèle généraliste. Par exemple, un système de reconnaissance appliqué au domaine journaliste donne de meilleures performances avec un modèle de langage adéquat. L'élaboration de modèles de langage spécifiques est coûteux pour le développement de nouvelles applications. La principale cause de leur coût vient du besoin en ressources linguistiques nécessaires à

7. L'algorithme EM est une méthode itérative pour évaluer le maximum de vraisemblance des paramètres de modèles. Il consiste en l'alternance de deux étapes : (1) le calcul de l'espérance de la vraisemblance en tenant compte des dernières variables observées et (2) l'estimation du maximum de vraisemblance trouvée dans l'étape (1).

leur apprentissage. L'adaptation d'un modèle de langage déjà existant est une solution intéressante. Le modèle de langage obtenu par adaptation est plus proche d'une application spécifique que le modèle de langage initial, tout en étant plus robuste qu'un modèle de langage appris sur peu de données d'apprentissage disponibles proches à la tâche de transcription.

De même, il est intéressant de modifier un modèle de langage en fonction des variations du contexte d'une application. Cette approche nécessite de détecter ces variations et d'adapter dynamiquement le modèle de langage. L'adaptation non-supervisée des modèles de langage consiste à rapprocher un modèle de langage général de certains sujets ou thèmes, sans que l'on dispose pour cela de données spécifiques du domaine. L'algorithme d'adaptation opère en deux parties : l'extraction des données d'adaptation, puis l'adaptation du modèle de langage proprement dite. Différentes méthodes d'adaptation [Bellegarda, 2004], [Liu et al., 2013] utilisant les données choisies peuvent être explorées.

Plusieurs techniques d'adaptation utilisent l'interpolation des modèles de langage. La fusion d'un modèle généraliste avec un modèle spécifique par interpolation linéaire est la technique la plus utilisée et la plus simple. Une mixture de modèles qui est une généralisation de l'interpolation de modèles de langage est aussi répondue. Au début, un modèle généraliste est estimé sur l'ensemble du corpus disponible, alors que chaque modèle spécialisé est estimé sur une sous-partie de ce corpus, chaque sous-partie correspondant à un thème particulier. Enfin, une mixture de tout ces modèles de langage est considérée.

Pour l'adaptation par Maximum *a posteriori*, au lieu de combiner les informations au niveau des modèles de langage, il est recommandé de les combiner directement au niveau des fréquences de mots au moment de l'estimation de ces modèles. L'utilisation du critère de maximum *a posteriori* est la technique la plus employée. Cependant, les adaptations par spécification de contraintes consistent à utiliser le corpus d'adaptation pour en extraire ses caractéristiques les plus significatives. Celles-ci doivent alors être satisfaites par le modèle de langage adapté : elles sont considérées comme des contraintes à respecter. Historiquement, les modèles de langage estimés à partir de spécifications de contraintes étaient associés à l'utilisation du critère d'*entropie maximale* qui permet de traiter les événements non observés dans le corpus d'apprentissage. Cette méthode est considérée comme un cas particulier de l'estimation par le critère d'information de discrimination minimale (*minimum discrimination information*- MDI), critère approprié pour l'adaptation des modèles de langage [Estève, 2002].

4 Évaluation des Modèles de Langage

Dans le cas de la RAP, la qualité d'un modèle de langage dépend de sa capacité d'augmenter le taux de reconnaissance des mots du système. Pour des considérations pratiques, les modèles de langage peuvent être évalués séparément de tout système par différentes mesures. La métrique la plus intuitive à utiliser pour évaluer un modèle de langage est la *probabilité* (appelée aussi *vraisemblable*) qu'il affecte aux données textuelles de test (ou de développement). Pour un modèle de langage n-gramme, on peut calculer les probabilités des données d'apprentissage ou du corpus de test T (ou de développement) composé des phrases $(t_1, ..., t_{k_T})$, en calculant le produit des probabilités de toutes les phrases du texte [Chen and Goodman, 1999] :

$$p(T) = \prod_{i=1}^{k_T} p(t_i) \tag{4.2}$$

En pratique, les probabilité $p(t_i)$ affectées par les modèles de langage aux phrases de test peuvent être très petites d'où leur manipulation n'est pas significative [Madnani, 2009]. Le fait d'introduire le logarithme pour appréhender des quantités de ce genre, permet d'obtenir des valeurs de *log-probabilités* plus représentatives en valeur absolue.

Une autre mesure inspirée du principe de la théorie d'information de Shannon (1948), est utilisée : l'*entropie croisée*, appelée aussi simplement *entropie*. Cette valeur est interprétée comme le nombre moyen de bits (ou nats) nécessaires au codage de chaque mot du corpus de test T, en utilisant un modèle de langage. Plus l'entropie croisée est basse, plus le modèle de langage est pertinent puisqu'il permet une meilleure compression. L'entropie croisée $H_p(T)$ du modèle considéré sur les données T est défini par l'équation 4.3 :

$$H_p(T) = -\frac{1}{W_T} log_b p(T) \tag{4.3}$$

où W_T est la longueur en mots du corpus de test T.

En effet l'*entropie croisée* d'une phrase n'est rien que sa *log-probabilité* avec un signe inverse normalisé par le nombre de ses mots.

La mesure la plus répondue pour l'évaluation des modèles de langage est la **perplexité**. Formellement, la perplexité d'un modèle de langage M, est l'inverse de la moyenne géométrique des probabilités qu'il affecte pour chacun des mots du corpus de test T. Intuitivement, la perplexité est vue comme le facteur de branchement dans une langue : plus la perplexité d'un modèle de langage est basse, moins il est confus (ou moins il est indécide) pour le choix des prochains mots dans une phrase. Elle est calculée par la formule :

$$PP_M(T) = b^{H_p(T)} \tag{4.4}$$

ou b est la base utilisé pour la représentation de l'information (bit ou nat).

Il est a noté que la perplexité est fortement dépendante du nombre de mots inconnus (UNK). Si ce dernier est important, la perplexité peut prendre des valeurs faibles dûe à sa sur-estimation. Il est préférable de calculer la perplexité sans le considérer [Haton et al., 2006]. Dans la litterature, d'autres mesures sont proposées pour l'évaluation des modèles de langage [Chen et al., 1998] mais la perplexité reste la mesure standard la plus répandue.

5 Modèles de Langage n-gramme

La modèlisation automatique du langage, qui revient à calculer la probabilité des phrases possibles dans ce langage, nécessite un corpus textuel dédié pour l'apprentissage. Généralement, un tel corpus ne peut jamais couvrir tout l'espace d'entrée (tout les mots du langage). Si la taille du corpus est supposée fixe, la sévérité de la dispersion des données (*data sparsity*) augmente relativement à la taille moyenne ou maximale des phrases. Cela découle du fait que la taille de l'espace d'entrée, qui représente l'ensemble de toutes les phrases possibles, croît avec la longueur maximale possible d'une phrase. Afin d'atténuer le problème de rareté des données, il est recommandé de limiter la longueur des phrases dont en aura à calculer leur probabilité. Cette idée est un fondement sur lequel les modèles de langage dits n-gramme sont basés.

La notion de n-gramme et plus particulièrement bigramme et trigramme (avec respectivement n=2 et n=3) est apparue à l'origine chez (Pratt, 1939) selon [Shannon, 1948]. Ce dernier a introduit la notion de n-gramme dans le cadre des systèmes de prédiction de caractères en fonction des autres caractères précédemment entrés. Un n-gramme de X

peut être défini comme une séquence de n X consécutifs. X peut alors être un caractère ou bien un mot, comme illustré dans la figure 4.1.

Les **n-gramme de caractères** sont les premiers à avoir été utilisés pour une tâche utilisant des données textuelles [Shannon, 1948]. Ce type de n-gramme est principalement utilisé dans l'identification de la langue ou encore la recherche documentaire. Il est a noté que les n-gramme de caractères prennent en considération les espaces. Ce type de descripteurs a plusieurs avantages parmi lesquels la non nécessité d'employer des descripteurs de type *radical*. En effet, la description d'un corpus par les n-gramme de caractères prend automatiquement en compte les racines des mots les plus fréquents. L'utilisation de n-gramme de caractères introduit également la notion d'indépendance de la langue. Aussi, les n-grammes de caractères sont tolérants aux fautes d'orthographes et au bruit pouvant être causé par exemple par l'utilisation de numérisation de documents par OCR (*Optical Character Recognition*).

L'utilisation des **n-grammes de mots** est plus récente que l'utilisation des n-grammes de caractères, dont par exemple [Solso, 1979] fut une des premières publications sur le sujet. Ce type de descripteurs est principalement employé dans le cadre de classification automatique de données textuelles, la reconnaissance automatique de la parole, etc. [Béchet, 2009]. Les n-grammes sont extrêmement efficaces et utilisés en pratique dans les systèmes de RAP, malgré il y a déjà 30 ans qu'ils ont été introduits dans ce domaine par Frederick Jelinek [Jelinek, 1997] et ses collègues.

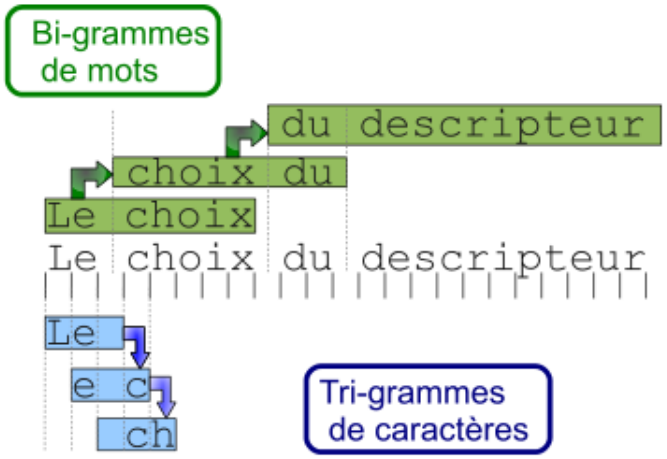

FIGURE 4.1 – Exemple de N-gramme de mots et de caractères.

Cette figure illustre la construction de n-gramme de caractères et de mots par la notion de déplacement de fenêtre. Ce déplacement se fait par étape, une étape étant soit un caractère ou bien un mot. Notons que les n-grammes (de mots et de caractères) sont construits à partir de flexions [Béchet, 2009].

La façon la plus naturelle de prédire l'occurrence d'un mot est de prendre en compte les mots qui lui sont liés avec une relation grammaticale ou sémantique. Dans le cas le plus général, le calcul de la probabilité d'un mot est basé sur l'ensemble des informations contextuelles. Pour une modélisation plus pratique et simplifiée, les relations entre les mots sont limitées aux relations dans la même phrase. Comme le discours peut être considéré comme un processus linéaire dans le temps, le contexte de gauche, semble être la bonne façon pour rechercher des mots liés.

Pour une phrase s composée de k mots $w_1, w_2, ..., w_k$, on peut exprimer sa probabilité $p(s)$, en fonction des probabilités conditionnelles de ses mots, par la formule :

$$p(s) = p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1 w_2) \cdots p(w_k \mid w_1 \cdots w_{k-1}) \tag{4.5}$$

Les **modèles de langages n-gramme** (où n est l'ordre du n-gramme maximal considéré) permettent d'évaluer la probabilité des phrases (suites de mots) apparaissant dans un corpus d'apprentissage de la langue considérée, en se basant sur l'hypothèse markovienne. Au lieu d'estimer la probabilité d'un mot en prenant en compte tout son historique, on considère uniquement les $n-1$ derniers mots. En pratique, les ordres fréquemment utilisés sont pour $n \in \{1,2,3,4,5,6,7\}$, on parle alors de modèle unigramme (n=1), bigramme (n=2), trigramme (n=3) et quadrigramme (n=4), etc. Tout les n-grammes qu'il est possible de construire à partir du lexique sont associés à une probabilité non nulle.

Par exemple pour le modèle bigramme, on aura l'approximation suivante :

$$p(s) \simeq \prod_{i=1}^{k} p(w_i \mid w_1 ... w_{i-2} w_{i-1}) \tag{4.6}$$

Pour mettre un sens à la probabilité $p(w_i \mid w_{i-1})$ dans l'équation (4.6) quand $i = 1$, on peut définir un token $< s >$ (ou $< BOS >$), qui représente le début de la phrase (*Begin Of Sentence*). Par analogie, et pour ajuster la somme des probabilités de toutes les phrases $\sum_s p(s)$ égale à 1, il nécessaire de définir aussi $< /s >$ (ou $< EOS >$), qui représente la fin de la phrase (*End Of Sentence*) [Chen and Goodman, 1999]. L'équation précédente peut être généralisée par l'équation (4.7) :

$$p(s) \simeq p(w_1) \prod_{i=2}^{k+1} p(w_i \mid w_{i-n+1}, ..., w_{i-1}) \tag{4.7}$$

ou encore plus simplement par l'équation (4.8) :

$$p(s) \simeq \prod_{i=1}^{k+1} p(w_i \mid w_{i-n+1}, ..., w_{i-1}) \tag{4.8}$$

en considérant que $\forall j \leqslant 0, w_j =< s >$ et $w_{k+1} =< /s >$.

Sous cette hypothèse, on peut dire que tout symbole (ou mot) dans une phrase peut être prédit à base des $n-1$ symboles précédents. Il s'agit en fait d'un raisonnement possible dans de nombreuses langues. Par exemple, considérant la phrase : "*Je suis de*". Même sans plus d'informations contextuelles entourant cette phrase,nous pouvons prédire que le mot suivant dans cette phrase sera probablement un nom de lieu ou de pays. En d'autres termes, la probabilité que le mot suivant est le nom d'un lieu ou d'un pays étant donné les trois mots précédents : "*Je suis de*" est plus élevée que tout autre mot. En général, ce principe est simple à fonctionner pour des contextes courts ou de longueurs moyennes.

L'estimation des paramètres d'un modèle de langage n-gramme s'effectue en deux étapes : (1) une opération de décompte des n-grammes et (2) une estimation des probabilités de ces n-grammes par le maximum de vraisemblance, qui est fréquemment utilisée. La distribution des probabilités du modèle est celle qui maximise la vraisemblance du corpus d'apprentissage. L'équation (4.8) exprime ce fait.

5.1 Estimation des Modèles de Langage n-gramme

L'estimation des probabilités d'un modèle de langage est effectuée à partir d'un corpus d'apprentissage, tout en considérant un vocabulaire de mots relatif à l'application cible.

Un mot est dit inconnu lorsqu'il apparaît dans les corpora d'apprentissage, de développement ou de test et qu'il ne figure pas dans le vocabulaire. Tout mot de cette

nature est remplacé dans une entité abstraite noté UNK (pour *UNKnown word*). Lors de l'apprentissage, une probabilité sera affecté au mot UNK comme tout autre mot du corpus.

La probabilité d'un mot sachant son historique $p(w_i \mid w_{i-n+1}, ..., w_{i-1})$ est estimée selon la méthode du maximum de vraisemblance en utilisant la fréquence normalisée : on considère le nombre d'occurrence du n-gramme $w_{i-n+1}...w_i$ dans le corpus d'apprentissage et puis on normalise par $\sum c(w_{i-n+1}...w_i)$, qui n'est rien que la fréquence d'occurrences de son historique $c(w_{i-n+1}...w_{i-1})$. L'équation (4.9) exprime ce fait :

$$p(w_i \mid w_{i-n+1}, ..., w_{i-1}) = \frac{c(w_{i-n+1}...wi)}{c(w_{i-n+1}, ..., w_{i-1})} \tag{4.9}$$

La méthode d'estimation, par le maximum de vraisemblance, prend en compte toutes les suites de n mots observés dans le corpus d'apprentissage afin de calculer leurs probabilités d'apparition. Il est fréquent que, les suites de n mots non observés dans ce corpus ont une probabilité nulle alors qu'il se peut qu'elles soient possibles dans la langue mais simplement absentes dans ce corpus.

5.2 Méthodes de Lissage des Modèles de Langage n-gramme

Il est préférable en cas de disponibilité, d'utiliser une quantité importante de données textuelle pour la construction des modèles de langage. Aussi, la formulation de la probabilité d'un mot par le modèle n-gramme admet une coupure de l'historique de dépendance. Même en considérant des contraintes telles que l'utilisation d'un grand corpus d'apprentissage et d'un historique réduit à quelques mots, l'estimation d'un modèle de langage n-gramme est toujours critique. On remarque que beaucoup de n-grammes possibles sont absents de ce corpus et le nombre d'occurrences de beaucoup de n-grammes présents n'est pas assez élevé pour être significatif. Pour pallier ce problème d'estimation on utilise généralement des techniques de lissage.

La plupart des techniques de lissage fonctionnent en deux étapes. Une technique simple de décompte est appliquée aux occurrences de n-gramme observés afin de prélever une masse de probabilité à ces événements. Cette masse est ensuite redistribuée aux événements non ou peu observés en se basant sur le principe que résume Chen [Chen and Goodman, 1999] en : *adjusting low probabilities such as zero probabilities upward, and high probabilities downward*. Plusieurs méthodes de lissage ont été proposées dans la littérature : la méthode additive, la méthode de *Good-Turing*, la méthode *Jelinkel-Mercer* [Jelinek, 1980], la méthode de *Katz*, le méthode de *Witten-Bell*, la réduction absolue (*Absolute discount*), le décompte de *Ristad's natural*, la méthode de *Kenser-Ney*, la méthode *Kneser-Ney* modifiée, etc. En général, ces méthodes peuvent être classées en deux catégories, selon la façon d'estimer les probabilité des mots : le lissage par repli ou par interpolation.

Lissage par repli :

Pour le décompte des occurences des n-grammes, on utilise la distribution du n-gramme si le nombre de ses occurences dans l'ensemble d'apprentissage est non nulle, sinon on se replie vers le niveau inférieur (*backoff*) et on utilise plutôt le décompte du $n-1$-gramme : d'où vient la nomination de la méthode *lissage par repli* ou (*backoff smoothing*). En général, on utilise le décompte des trigrammes, sinon on passe aux bigrammes ou aux unigrammes. La méthode de *Katz* est un exemple standard pour ce type de lissage.

Lissage par interpolation :

Pour ce type de lissage, on utilise une *interpolation* entre l'information de différents niveaux des nombres d'occurrences des n-grammes. En général, on se retrouve avec une mixture des décomptes des trigrammes, bigrammes et unigrammes. La méthode de *Jelinkek-Mercer* et du *décompte absolu* utilisent une interpolation linéaire entre les nombres d'occurrences des différents ordre des n-grammes.

Des adaptations sont faites pour présenter la plupart des méthodes de lissage avec les deux versions d'implémentations[Chen and Goodman, 1999], [Stolcke, 2002]. Aussi, un état de l'art sur les différentes méthodes de lissage pour la modélisation du langage est présenté dans [Chen, 1996], [Chen and Goodman, 1999], [Zhai and Lafferty, 2001].

En général, les facteurs les plus important qui influencent la qualité du modèle n-gramme résultant de l'apprentissage est le choix de l'ordre n et de la méthode de lissage considérée.

Lissage Additif

Le lissage additif *Additive smoothing* [Lindley, 1962] est une des plus simples méthodes de lissage. Pour éviter les probabilités nulles des n-grammes, on prétend que chaque n-gramme se répete δ fois plus qu'ils apparaissent dans l'ensemble d'apprentissage, tel que $0 < \delta \leq 1$, en utilisant l'équation 4.10 :

$$p_{add}(w_i|w_{i-n+1}^{i-1}) = \frac{\delta + c(w_{i-n+1}^{i})}{\delta|V| + \sum_{w_i} c(w_{i-n+1}^{i})} \tag{4.10}$$

où V est le vocabulaire.

Estimation de Good-Turing

Les travaux de Good [Good, 1953] montrent que pour obtenir une estimation précise de la fréquence réelle d'un n-gramme, il faudrait décompter de sa fréquence une certaine quantité. Il propose de modifier la calcul de la fréquence d'un n-gramme apparaissant r fois ainsi : à $r_{gt} = (r+1)\frac{n_{r+1}}{n_r}$ où n_r est le nombre des n-grammes qui apparaissent r fois dans le corpus d'apprentissage. La probabilité du n-gramme d'occurrence r est donnée par : $p_{gt}(w_i|h) = \frac{r_{gt}}{N}$ avec $N = \sum_r n_r r_{gt}$. En pratique, l'estimation de Good-turing n'est pas utilisée seule mais elle est combinée à d'autres méthodes.

Interpolation de Jelinek-Mercer

Pour un modèle bigramme, le principe de la méthode de Jelinek-Mercer [Jelinek, 1980] est de supposer si un événement ne s'est pas produit, sa probabilité sera estimée en prenant en compte l'événement de rang inférieur. Un moyen simple est de faire un repli sur les modèles inférieurs et combiner les probabilités des deux événements de rang i et $i-1$, par l'équation suivante :

$$p(w_i|h) = \lambda_h p(w_i|h) + (1 - \lambda_h) p(w_i|h^{-1}) \tag{4.11}$$

où $h^{-1} = w_{i-k+2} \cdots w_{i-1}$. Ce processus est appliqué récursivement jusqu'au niveau du zéro-gramme où l'on utilise une distribution uniforme.

Repli de katz

Une autre approche proposée par Katz [Katz, 1987] consiste à ne se servir d'un modèle d'ordre inférieur que lorsque le n-gramme dont on cherche la probabilité n'a pas été observé dans le corpus d'apprentissage. Cette technique est couramment appelée *backoff* de katz et peut être formulée ainsi :

$$p_{lissee}(w_i|h_i^n) = \begin{cases} p_{decompte(w_i|h_i^n)} & si\ r(h_i^n, w_i) > 0 \\ \alpha_{h_i^n} p_{lissee}(w_i|h_i^{n-1}) & sinon \end{cases}$$

où $\alpha_{h_i^n}$ est un coefficient de normalisation et de redistribution de la masse de probabilité décomptée à la distribution initiale. L'avantage d'une telle approche est que la distribution sur laquelle le repli est effectué peut être optimisée pour les événements non observés étant donné qu'elle n'est pas interpolée avec les événements observés.

Méthode de Witten-Bell

La méthode de lissage de Witten-Bell [Witten and Bell, 1991] est développée initialement pour la tâche de la compression des textes. En particulier, le modèle de langage lissé du n^{ieme} ordre est définie récursivement par une interpolation lineaire entre le n^{ieme}-maximum vraisemblant modèle (calculé en utilisant l'équation (4.7)) et le $(n-1)^{ieme}$ modèle lissée comme suit :

$$p_{wb}(w_i|w_{i-n+1}^{i-1}) = \lambda_{i-n+1}^{i-1} p_{ML}(w_i|w_{i-n+1}^{i-1}) + (1 - \lambda_{i-n+1}^{i-1}) p_{wb}(w_i|w_{i-n+2}^{i-1}) \quad (4.12)$$

Pour calculer les paramètres λ_{i-n+1}^{i-1}, on utilise le nombre des mots qui suivent l'histoire de w_{i-n+1}^{i-1}. Avec la probabilité λ_{i-n+1}^{i-1}, on doit utiliser le modèle de l'ordre le plus élevé et avec $1 - \lambda_{i-n+1}^{i-1}$ on utilise le modèle à faible ordre. Donc, l'idée est d'utiliser le modèle d'ordre élevé si le n-gramme correspondant apparaît dans l'ensemble d'apprentissage sinon on fait un repli au modèle d'ordre inférieur.

Décompte absolu

La technique du décompte absolu (*absolute discounting*) a été introduite par Ney et Essen [Ney and Essen, 1991]. Il s'agit d'une technique de *backoff* plus simple que celle de Good-Turing. Au lieu de calculer la quantité à décompter en fonction de la fréquence du n-gramme, cette quantité est fixe pour toutes les fréquences. La fréquence r d'un n-gramme devient alors :$r_{ad} = r - D$ où D est estimé par :

$$D = \frac{n_1}{n_1 + 2n_2} \quad (4.13)$$

et n_1, n_2 est le nombre total des n-grammes avec exactement un et deux comptes respectivement dans l'ensemble d'apprentissage. Une autre version de décompte absolu par interpolation existe [Ney et al., 1994]. Le résultat de l'interpolation est encodé par l'équation 4.14 :

$$p_{abs}(w_i|w_{i-n+1}^{i-1}) = \frac{max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{abs}(w_i|w_{i-n+2}^{i-1}) \quad (4.14)$$

Lissage de Kneser-Ney

La méthode originale de Kneser-Ney [Kneser and Ney, 1995] est une extension de la méthode de la réduction absolue où la distribution des ordres supérieur et inférieur sont combinées d'une nouvelle manière. La version originale de Kneser-Ney est donnée par l'équation suivante 4.15 : $p_{ukn}(w_i|w_{i-n+1}^{i-1}) =$

$$\begin{cases} \frac{max\{c(w_{i-n+1}^{i})-D,0\}}{\sum_{w_i} c(w_{i-n+1}^{i})} & c(w_{i-n+1}^{i}) > 0 \\ \gamma(w_{i-n+1}^{i-1}) p_{KN}(w_i|w_{i-n+2}^{i-1})) & c(w_{i-n+1}^{i}) = 0 \end{cases} \tag{4.15}$$

où $\gamma(w_{i-n+1}^{i-1})$ est choisi tel que la somme de la distribution est égale à 1. Aussi, une deuxième version à base d'interpolation existe, en utilisant l'équation suivante :

$$p_{ukn}(w_i|w_{i-n+1}^{i-1}) = \frac{max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{ukn}(w_i|w_{i-n+2}^{i-1}) \tag{4.16}$$

Méthode de Kneser-Ney modifiée

Kneser et Ney [Kneser and Ney, 1995] ont proposé une méthode de lissage qui repose sur une technique de décompte absolu et une redistribution par repli. Chen et Goodman ont introduit une modification sur cette technique pour traiter le cas particulier des n-grammes peu observés. La nouvelle techniques de lissage dite de *Kneser-Ney modifiée* est fréquemment utilisées vu ses performances testées dans [Chen and Goodman, 1999]. La *Méthode de Kneser-Ney modifiée* utilise trois paramètres, D_1, D_2, D_{3+}, qui sont appliqués aux n-grammes avec respectivement un, deux et trois décomptes. L'équation utilisée est la suivante :

$$p_{kn}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) - D(c(w_{i-n+1}^{i}))}{\sum_{w_i} c(w_{i-n+1}^{i})} + \gamma(w_{i-n+1}^{i-1}) p_{kn}(w_i|w_{i-n+2}^{i-1}) \tag{4.17}$$

où

$$D(c) = \begin{cases} 0 & c = 0 \\ D_1 & c = 1 \\ D_2 & c = 2 \\ D_{3+} & c \geqslant 3 \end{cases}$$

et $\gamma(w_{i-n+1}^{i-1})$ est défini par une combinaison de D_1, D_2 et D_{3+}. Dans plusieurs cas la version de Kneser-Ney modifiée à base d'interpolation est considérée la plus performante en la comparant aux autres méthodes précédemment citées. Une extension de cette méthode est proposée dans [Andrés-Ferrer et al., 2012].

Repli stupid

Le repli stupide est introduit en [Brants et al., 2007] est motivé par l'apprentissage des modèles de langage sur des corpora extrêmement larges (plus de 2 billions (10^{12}) de mots). Au lieu d'utiliser n'importe quelle méthode de décompte, on utilise directement les fréquences relatives selon l'equation 5.2 :

$$r_{sp}(w_n|w_1^{n-1}) = | \begin{cases} p_{MV}(w_n|w_1^{n-1}) & if\ c(w_1^n) > 0 \\ \gamma r_{sp}(w_n|w_2^{n-1}) & sinon \end{cases}$$

où γ est un facteur de repli fixé pratiquement à 0.4. Comme aucune normalisation n'est faite, on manipule des scores au lieux des probabilités. Avec des quantités de données très importantes pour l'apprentissage ($\gtrsim$ des billions de mots), la non-normalisation ne cause aucune dégradation sur la performance du système. Selon les résultats empiriques, la performance du repli stupide est proche de celle de la méthode de Kneser-Ney. Il est possible de conclure que, l'utilisation de grande quantité de données pour l'apprentissage des modèles de langage n-gramme est plus important que le choix de telle ou telle méthode de lissage. Or la disponibilité d'énormes quantités de données et de puissantes machines n'est pas évident.

6 Modèles de Langage avancés

Pendant plus de vingt ans, la modélisation du langage se base sur les techniques de lissage des modèles n-gramme. Ces derniers sont simples, efficaces et souples. Ils se sont bien imposés dans les systèmes de l'état de l'art et ils continuent d'être quasi systématiquement intégrés dans nombreuses applications réussies de la RAP, la traduction automatique statistique, etc. Malgré le fait que les modèles de langage n-gramme utilisent seulement un court historique des mots précédents pour prédire le mot suivant, ils restent des éléments clés pour une modélisation de bonne qualité et à perplexité relativement faible.

Quelques faiblesses des modèles de langage n-gramme ont émergé dans la communauté scientifique. Un des majeurs inconvénients des modèles de langage n-gramme est leur performance de modélisation limitée ou détériorée, dans le cas d'une sévère dispersion des données (*data sparseness*) pour les tâches difficiles de reconnaissance. Même lorsque des corpora d'apprentissage importants et/ou de bonnes méthodes de lissage sont utilisées, de faibles probabilités peuvent être affectées à de nombreux n-grammes valides.

Cependant, de nouvelles alternatives considérées plus efficaces sont proposées dans la littérature. L'objectif de cette section est de présenter les modèles de langage spatiaux discrets et les architectures des modèles à base de réseaux de neurones artificiels les plus utilisés.

6.1 Modèles de Langage spatiaux discrets

Plusieurs modèles de langage de type spatiaux discrets sont introduits pour la modèlisation du langage comme : les modèles de langage basés sur les classes de mots, les modèles de langage structurés, les modèles de langage à base de forêts aléatoires qui ont contribué partiellement à l'amélioration de la modélisation langagière. Cependant, les modèles de langage à maximum d'entropie, ou les modèles exponentiels ont introduit des améliorations intéressantes pour le cas de dispersion des données.

Modèles de Langage n-classes et leurs variantes

Les modèles de langage n-classes (*Class-based LMs*) [Brown et al., 1992] consistent à regrouper les mots de classes selon plusieurs critères. Trois types de classes sont les plus utilisées : les classes syntaxiques qui regroupent les mots selon leurs catégories grammaticale, les classes morphologiques qui regroupe les mots ayant la même racine morphologique (lemme) dans une seule classe et enfin les classes obtenues par d'autres méthodes de classification automatique. Dans la pratique, la combinaison des modèles n-classes avec des modèles de langage n-gramme est souhaitable. Plusieurs variantes des modèles n-classes [Emami and Jelinek, 2005] sont proposées dans la littérature. Par exemple, le modèle

Part Of Speech (POS) se base sur une partition de la parole et une connaissance morpho-syntaxique. Dans ce genre de classes, un mot peut appartenir à plusieurs classes à des instants différents, par exemple le mot *parti* peut correspondre à un nom *parti politique* ou à un verbe (*verbe partir).* Les modèles de langage à base de classe nécessitent des corpora d'apprentissage pré-étiquetés [Smaïli, 1991]. Cependant, ce processus d'étiquetage est coûteux s'il est fait manuellement et les résultats obtenus sont moins exacts s'il est fait d'une façon automatique. Aussi, ces modèles se limitent à un historique restreint de quelques classes (identiquement à ce qui se passe avec les mots pour les n-grammes), ce qui limite la qualité de la modélisation du langage.

Modèles de Langage à cache et déclencheurs

Le modèle à cache est un cas particulier du modèle déclencheur (*Trigger LM*), qui est considéré comme un modèle à historique long. Le principe dans ce type de modèle est qu'il existe une quantité d'information significative dans l'historique des mots rentables à exploiter. Le modèle cache [Kuhn and De Mori, 1990] permet de renforcer la probabilité de la re-apparition d'un mot s'il a déjà été rencontré à un endroit dans le texte. Ce modèle utilise une fenêtre glissante (dite : le cache) de taille fixe (10-1000) pour examiner la fréquence relative des mots. Tandis que le modèle déclencheur suppose que certains mots apparus dans l'historique ont une influence sur l'apparition d'un autre. Il s'intéresse donc aux mots qui apparaissent souvent ensemble. Par exemple, le mot *avion* déclenchera probablement le mot *voyage*, *aéroport*, etc.

En pratique, les modèles de langage à cache et déclencheur ne sont pas utilisés seuls. Le modèle cache est le plus souvent combiné à un modèle langage n-gramme (un bigramme ou un trigramme) et le modèle déclencheur est généralement combiné à un modèle de langage n-gramme standard et à un modèle à cache.

Modèles de Langage structurés

Les modèles de langage structurés (*Structured LMs*) [Chelba and Jelinek, 2000] [Filimonov and Harper, 2009] tentent de combler les différences entre la théorie linguistique à base de grammaires et les modèles statistiques des langues naturelles. Dans le cadre de cette modélisation, la phrase est considérée comme une structure arborescente générée par une grammaire libre de contexte, où les feuilles sont les mots et les nœuds individuels sont des symboles non-terminaux. L'approche statistique est utilisée lors de la construction de l'arbre : les dérivations attribuent des probabilités estimées à partir des données d'apprentissage, ainsi la probabilité pour chaque nouvelle phrase peut être générée par la grammaire donnée.

L'avantage de ces modèles de langage réside dans leur capacité théorique de représenter des modèles de phrases à travers de nombreux mots. En outre, ces modèles rendent la modélisation du langage beaucoup plus attrayante pour la communauté linguistique. Cependant, il y a de nombreux inconvénients pratiques des modèles linguistiques structurés. On cite deux principaux points : (1) la complexité du calcul et (2) la nécessité d'une grande intervention manuelle par des linguistes experts lorsque cette technique doit être appliquée à des modèles de langage dédiés à de nouveaux domaines ou de nouvelles langues, qui peut être très coûteux.

Modèles de Langage à base d'arbres de décision et de forêts aléatoires

Un arbre de décision est un outil d'aide à la décision utilisé en apprentissage automatique, représentant un ensemble de choix sous une forme graphique d'un arbre. L'algorithme des forêts d'arbres décisionnels effectue à son tour un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

De ce fait, un arbre de décision peut être utilisé pour la modélisation du langage puisqu'il permet de partitionner les données en posant des questions sur leurs historiques à chaque noeud. Ce type de modèles est dit modèle de langage à base d'arbres de décision (*Decision Trees LM*). Par exemple, il est possible de poser des questions au sujet de la présence d'un mot spécifique dans l'histoire de dix mots. Cependant, dans la pratique, il a été constaté que la recherche d'un bon arbre de décision peut être assez difficile. D'où l'utilisation des modèles à base de forêts aléatoires (*Random Forest LMs*) [Xu and Jelinek, 2004] [Oparin et al., 2008] est considérée plus intéressante. Une combinaison de plusieurs arbres de décision aléatoires est habituellement faite par une interpolation linéaire afin d'obtenir les modèles de langage à base de forêts aléatoires. Ces modèles ont une possibilité de bien fonctionner dans les langues flexionnelles, en posant des questions au sujet de la morphologie des mots dans leurs historiques.

L'inconvénient de ce type de modèle est la complexité élevée des calculs. Aussi, ses performances semblent diminuer lorsque la quantité de données d'apprentissage est importante. Ainsi, ces techniques semblent fonctionner similaire à les modèles de classes sur la base, à certains égards.

Modèles de Langage à maximum d'entropie

Les modèles de langage à maximum d'entropie ou dits exponentiels (*maximum entropy LM* or exponential LM)[Rosenfeld, 1996] se basent sur le classificateur de régression logistique qui permet de prédire les mots suivants. Ces classificateurs peuvent utiliser les fonctions standard (les n-grammes), mais aussi beaucoup d'autres prédicateurs utiles, tout comme d'autres types de modèles de langage discriminants [Roark et al., 2007]. Si les modèles de langage factorisés ont mis en place une solution partielle au problème de dispersion des données en utilisant des notions de similarité. Ces modèles permettent d'incorporer manuellement différents paramètres (parties de parole (*Part of speech tags*), structure syntactique, etc.) dans les modèles de langage, au lieu de se contenter seulement des séquences de mots.

Selon Mikolov [Mikolov, 2012], les modèles à maximum d'entropie peuvent être facilement entraînés par l'algorithme de gradient de descente stochastique (Stochastic Gradient Descent - SGD). En fait, ces modèles, qui apportent une valeur ajoutée à la modélisation du langage, peuvent être considérés comme un simple réseau de neurones sans couches cachées.

Modèle de Langage à base de sous-mots

En outre, les modèles de langage à base de sous-mots (*Sub-word based LMs*) ont introduit une contribution supplémentaire à la solution de ce problème en décomposant les mots en plus petits et plus fréquents unités de sous-mots. Les mots de la langue considérée sont décomposés en un certain type de sous-mots appelés unités sous-lexicales (comme des morphèmes ou des syllabes), puis les modèles de langage n-gramme sont estimés sur des séquences de ces sous-mots. Normalement, le nombre des sous-mots possibles dans un corpus de texte donné est inférieur au nombre de mots complets qui conduit à une

couverture lexicale plus élevée. Ce type de modèles sont bien adaptés pour la modélisation des langues à riche morphologie [Yvon].

6.2 Modèles de Langage neuronaux continus

Cependant, aucune solution vigoureuse au problème de la rareté des données (*data sparsity*) n'était encore atteinte malgré tout les modèles de langage proposés. Une meilleure solution à ce problème pourrait être trouvée en traitant les sources de l'origine du problème à savoir : le mode de représentation du texte, ainsi que la construction fondamentale de l'espace des paramètres lors de l'estimation du modèle de langage. En fait, l'échec à traiter les domaines de données clairsemées est un inconvénient commun à tout les modèles de langage qui sont estimés dans un espace discret. Cela rend difficile de parvenir à des niveaux élevés de généralisation, même après l'application des techniques de lissage les plus efficaces, comme la méthode de Kneser-Ney modifiée (MKN) pour les modèles de langage *backoff* n-gramme ou l'utilisation des autres modèles de langage discrets.

Pour les approches usuelles qui reposent sur des modèles de langage n-gramme discrets, estimés avec des méthodes de lissage, l'occurrence d'un mot dans son contexte est considérée comme la réalisation d'une variable aléatoire discrète. L'espace de réalisation de cette variable aléatoire est le vocabulaire tout entier et au sein duquel il n'existe aucune relation entre les mots. Par exemple, aucune information statistique n'est partagée entre formes morphologiquement ou sémantiquement apparentées. Le caractère très inégal des distributions d'occurrences dans les textes implique que les modèles de langage résultants sont souvent estimés à partir du nombre d'occurrences dans l'ensemble d'apprentissage ce qui n'assure pas une bonne capacité de généralisation. Par ailleurs, le nombre de paramètres d'un modèle de langage n-gramme augmentant de manière exponentielle avec son ordre n [Mikolov, 2012], d'où il devient théoriquement difficile de résoudre ce problème en augmentant la quantité des ressources textuelles disponibles.

Le problème principal pour la modélisation du langage est le manque d'une notion de similitude entre les mots. Les mots sont représentés dans un espace discret (l'espace vocabulaire), ce qui rend impossible d'effectuer une véritable interpolation à rapprocher les probabilités des n-grammes non vus. Pour ces raisons, le passage à un espace continu de représentation est considéré comme une promotion naturelle de l'estimation des probabilités des modèles de langage.

Le principe de la représentation continue des mots avec les réseaux de neurones est de convertir les indices numériques du vocabulaire des mots dans un espace continu de représentation et d'utiliser un estimateur de probabilité de fonctionnement dans cet espace. Étant donné que les distributions qui en résultent sont des fonctions lisses de la représentation de mot, de ce fait une meilleure généralisation peut être assurée.

Rappelons que, les approches neuronales furent les premières à être utilisées afin de réaliser un apprentissage automatique. Ces approches s'inspirent du fonctionnement du système nerveux humain. Ainsi, elles se fondent sur l'utilisation de neurones artificiels qui vont effectuer la tâche d'apprentissage. Les automates à seuil visant à modéliser l'activité neuronale [McCulloch and Pitts, 1943] ainsi que les règles d'apprentissage locales introduites par Hebb et Windrow [Hebb, 1961] et [Widrow and Stearns, 1985] se sont fortement inspirés des méthodes d'apprentissage neuronal. Les premières approches de ce type furent les réseaux de neurones monocouches tels que le *Perceptron* [Rosenblatt, 1958] ou bien encore *Adaline* [Widrow et al., 1960] (ré-édité dans [Widrow and Hoff, 1988]). Plus tardivement furent introduits les réseaux de neurones multicouches dont notamment le Perceptron multicouches [Cybenko, 1989] et l'algorithme de rétro-propagation

du gradient.

Historiquement, les modèles de langage neuronaux sont une des premières réalisations intéressantes avec des architectures profondes, pour des applications en RAP depuis les travaux pionniers de Elman [Elman, 1990] puis de Dahl [Dahl et al., 2012] et Hinton [Hinton et al., 2012]. Aussi, Collobert et Weston(2008) ont montré qu'un seul modèle de réseau neuronal peut obtenir les résultats de l'etat de l'art sur les tâches telles que le marquage de partie du discours (*part-of-speech tagging*) et de la reconnaissance des entités nommées(*named entity recognition*), etc. Enfin, les modèles linguistiques à base de réseau de neurones [Bengio et al., 2003], [Schwenk, 2007], [Mnih and Hinton, 2007], [Mikolov, 2012] ont surpassé les modèles de langage basé sur le comptage traditionnel des n-grammes. Holger Schwenk a repris les modèles de langage proposés par Bengio et il a montré qu'ils fonctionnent bien dans les systèmes de RAP, et qu'ils sont même complémentaires aux modèles de langage n-gramme.

Les modèles de langage neuronaux se caractérisent par une méthode d'estimation alternative des probabilités qui se fonde sur une représentation continue. Chaque mot du vocabulaire est représenté comme un point dans un espace métrique. La probabilité d'un mot est alors une fonction des représentations continues des mots qui composent son contexte. Ces représentations, ainsi que les paramètres de la fonction d'estimation, sont apprises conjointement par un réseau de neurones direct qui peut être soit simple, large ou profond (*Feed-forward Neural Network* - FNN) ou convolutionnel [8] [LeCun and Bengio, 1995] [Collobert et al., 2011], récurent (*Reccurent Neural Network* - RNN) (voir la figure 4.2), reccursive [9] [Socher et al., 2011], etc. Cette stratégie d'estimation permet aux mots similaires d'avoir des représentations proches. Ainsi, ce type de modèle introduit la notion de similarité entre les mots et son utilisation est sencée permettre une meilleure exploitation des données textuelles.

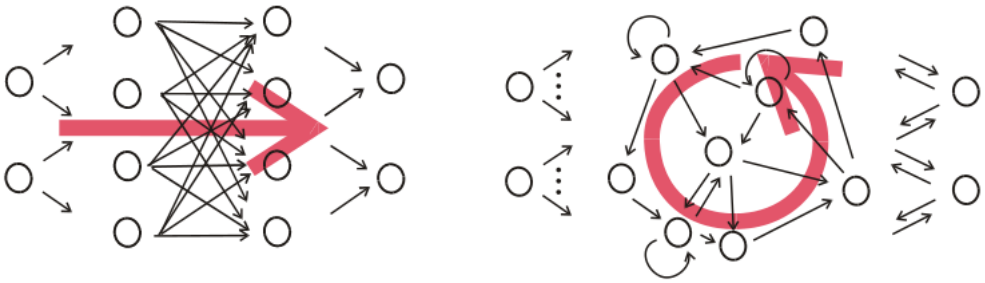

FIGURE 4.2 – Architectures du perceptron multicouches directe (gauche) et le réseau de neurones récurrent (à droite) [Jaeger, 2002].

Pour le cas des réseaux de neurones directs (*Feedforward neural network* - FNN) tels que le perceptron multicouches, l'activation est canalisée à travers le réseau : sortie de gauche à droite. Un réseau neuronal récurrent (RNN) possède (au moins) un chemin cyclique de connexions synaptiques.

8. Un réseau de neurones constitutionnel CNN est formé par un empilement de couches de traitement indépendantes : la couche de convolution (CONV) qui traite les données d'un champ récepteur, la couche de *pooling* (POOL), qui permet de compresser l'information intermédiaire (souvent par sous-échantillonnage), la couche de correction (ReLU) en référence à la fonction d'activation (Unité de rectification linéaire), la couche entièrement connectée (FC) qui est une couche de type perceptron et la couche de perte (LOSS).

9. Un réseau neuronal récursif (à ne pas confondre avec *récurent*) est un type de réseau neuronal profond crée en appliquant le même ensemble de poids récursivement sur une structure, pour produire une prédiction structurée sur une entrée à longueur variable, ou une prédiction scalaire sur celle-ci, en parcourant une structure donnée en ordre topologique.

Représentation des mots

L'idée de la représentation distribuée pour les concepts était introduite par Hinton [Hinton, 1986]. Tandis que le principe de représentation des mots par les réseaux de neurones, qui est le point fort des modèles de langage neuronaux et le secret du succès de nombreux applications etait exploré à l'origine par Bengio [Bengio and Bengio, 1999] [Bengio et al., 2003], à une époque où les réseaux de neurones étaient démodés et délaissés. Bengio et al. ont démontré dans leurs travaux comment les représentations distribuées des symboles pouvaient être combinées avec des prédictions de probabilité de réseau neuronal afin de surpasser les modèles de langage standard n-gramme sur les tâches de modélisation statistique du langage.

Quelques années après, un sucés de l'apprentissage neuronal en profondeur- où plusieurs couches cachées sont considérées- a émergé. Depuis 2006, le domaine a fait de grandes avancées avec l'apprentissage de réseaux de neurones dits profonds (*deep learning*). Les réseaux profonds sont composés de plusieurs couches de neurones. Chaque couche est une étape qui représente les données de façon un peu plus abstraite en se basant sur ce qui a été appris dans la couche précédente. Les neurones humains reçoivent en entrée les signaux provenant des autres neurones par des synapses. Appliquée aux réseaux de neurones artificiels, la connexion neuronale s'effectue par le biais de liaisons pondérées (les synapses) unidirectionnelles. Les couches suivantes du réseau sont classiques, le but va être d'attribuer des poids synaptiques à chaque neurone afin d'obtenir le résultat voulu en sortie. C'est ainsi que l'on construit une base d'apprentissage en pondérant les différents poids synaptiques d'un réseau afin d'améliorer le résultat ou de se rapprocher au mieux du résultat souhaité.

Rappelons que, les réseaux de neurones sigmoïdes (et avec d'autres fonctions d'activation) sont universels : il est utile d'utiliser suffisamment de neurones cachés [10] pour apprendre (au lieu d'extraire) les caractéristiques des entrées, puis passer à l'approximation de n'importe quelle fonction continue [Hornik et al., 1989]. La valeur ajoutée avec cette approche dite *profonde* est ou d'apprendre une représentation riche à partir des données d'entrée. comme il est illustré dans la figure 4.3.

Cette approche est à l'image de l'apprentissage humain qui commence par apprendre des concepts simples, comme l'addition et la soustraction en mathématiques. On se base ensuite sur ces concepts afin d'en apprendre des plus complexes, comme la multiplication et le principe de fonction.

En ce qui concerne le langage naturel, l'entrée x code des caractéristiques telles que des mots w ou d'autres informations linguistiques. Le passage des modèles de langage n-gramme au réseaux de neurones permet de représenter chaque entité comme une dimension non unique mais plutôt par un codage dit chaud (*one-hot*) et/ou par des vecteurs denses de valeur réelle de taille fixe. C'est-à-dire que chaque mot est incorporé dans un espace dimensionnel et représentée comme un vecteur dans cet espace. Les représentations sont normalement stockées dans une table (notée C).

En effet, la représentation chaude (*one-hot*) consiste en un vecteur où tout les éléments ont une valeur de 0 à l'exception d'un seul qui est égal à 1. Dans le cas de la modélisation langagière, chaque élément du vecteur correspond à un mot du vocabulaire et l'identifiant (qui est à la fois l'index pour le vecteur et pour le vocabulaire) signale quel élément

10. Une idée exploitée et rendue commune est que les représentations des mots sont faites par un *apprentissage profond* dans un certain sens puisque ces représentations peuvent être réalisées par des réseaux profonds ou peu profondes, et même sans doute, superficiels (avec une seule couche). Ce dernier choix est le choix dominant.

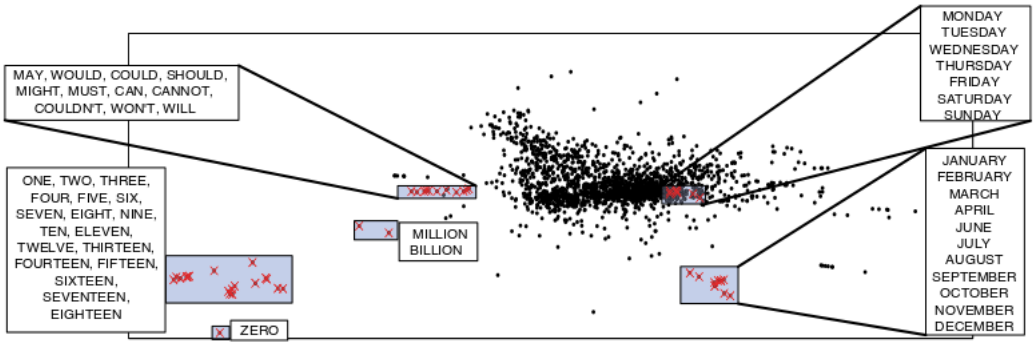

FIGURE 4.3 – Exemple d'une représentation distribuée de 2-dimensions des mots, obtenue dans [Blitzer et al., 2005].

Le cerveau humain apprend en utilisant des représentations distribuées, qui permettent de comprendre mieux les choses. Chaque dimension de l'espace de représentation distribuée doit correspond à une dimension sémantique ou grammaticale, etc. des mots. D'où les mots semblables se rapprochent les uns des autres dans cet espace, du moins certaines directions. Une représentation distribuée est opposée à une représentation locale, dans laquelle un seul neurone (ou très peu) est actif à chaque instant. On peut voir les modèles de langage n-gramme comme une représentation principalement locale : seules les unités associées aux sous-séquences spécifiques de la séquence d'entrée sont activées.

(mot) a la valeur 1. Chaque identifiant possède donc une représentation vectorielle *one-hot* qui peut être utilisée comme représentation de base qui servira d'entrée au réseau de neurones. Notons que le nombre d'éléments (dimension) du vecteur grandit linéairement avec la taille du vocabulaire. Il est facile de voir que ce schéma d'encodage correspond parfaitement à notre objectif d'avoir un minimum de similitude entre les mots similaires, par exemple w_i et w_j sont deux mots : si $w_i = [0, 0, \cdots, 1, \cdots, 0, 0]^T \in \{0, 1\}^{|V|}$

$$|w_i - w_j| = \begin{cases} 1 & i \neq j \\ 0 & sinon \end{cases}$$

Dans le domaine du traitement automatique du langage et les systèmes LVCSR (*Large Vocabulary Continuous Speech Recognition*), les vocabulaires ont tendance à être volumineux, ce qui peut donner des vecteurs de très haute dimension. Le problème avec cette approche est que le vecteur *one-hot* ne donne pas beaucoup d'informations sur le mot qu'il représente. Par exemple, il ne permet pas de faire des comparaisons pour savoir si deux mots sont sémantiquement similaires.

Maintenant, l'entrée de la couche du réseau de neurone est une séquence de $n-1$ vecteurs de ce genre, ce qu'on peut désigner par $(w_1, w_2, ..., w_{n-1})$, ces vecteurs seront multipliés par une matrice de poids E. En bref, la multiplication de la transposition d'une matrice avec un vecteur uni-colonne équivaut à prendre une seule rangée de la matrice. Après cela, on obtient une suite de vecteurs continus $(p_1, p_2, \cdots, p_{n-1})$. Ces vecteurs p sont une représentation de $n-1$ mots d'entrée dans un espace vectoriel continu et souvent appelé *vecteur de contexte* (ou *word embedding*) [Cho, 2015].

Concernant la représentation dense (*word emdeddings*), elle permet de représenter un mot par un vecteur plus petit et plus riche en information que l'encodage chaud. En effet, le vocabulaire est représenté dans un espace continu où les mots sémantiquement similaires sont des points placés à courte distance les uns des autres. Il suffit d'apprendre la valeur de ces paramètres en utilisant la technique de descente de gradient minimisant la fonction objectif du modèle. On apprend ainsi des représentations de mots contenants de l'information utile au modèle de langage pour accomplir la tâche de reconnaissance.

C'est-à-dire que chaque mot est incorporé dans un espace dimensionnel qui permet de le représenter comme un vecteur dans cet espace. La représentation vectorielle de

chaque caractéristique centrale peut alors être réalisée par une couche cachée du réseau de neurones. Aussi, d'autres techniques avancées pour la représentation vectorielle des mots [11] peuvent être consultées dans [Mikolov et al., 2013b], [Mikolov et al., 2013a], etc.

A titre d'exemple, on peut représenter les caractéristiques du mot par un vecteur de 100 dimensions (*One-hot*) et en utilisant les parties de la parole (Part-Of-Speech *POS*) [12] par un vecteur de 20 dimensions. La figure 4.4 montre ces deux approches de représentation possibles.

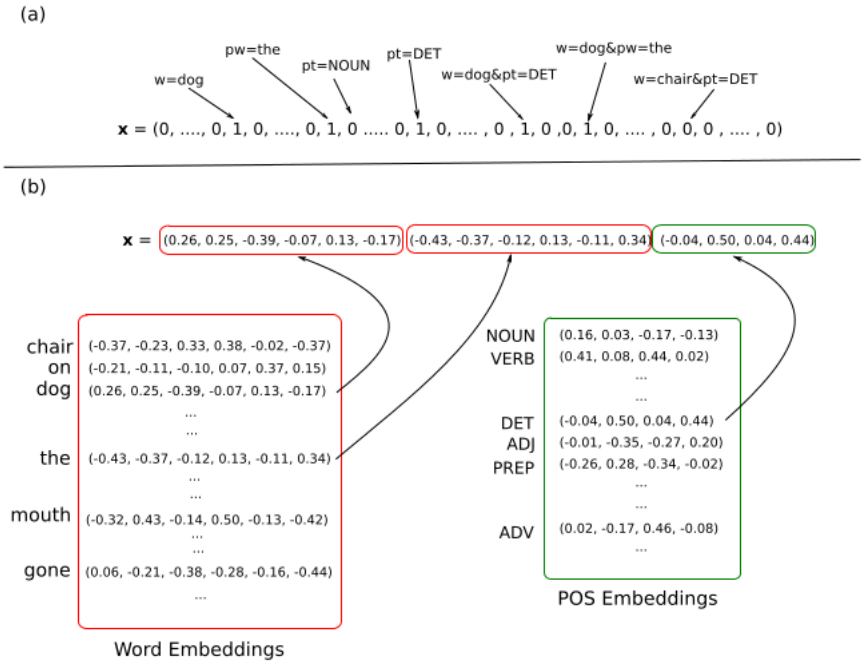

FIGURE 4.4 – Deux types de représentation des caractéristiques espacées (*One-hot*) ou denses [Goldberg, 2015].

Les représentations distribuées des mots peuvent être dérivées de plusieurs façons. Deux exemples de codages des informations existent : (a) le vecteur de caractéristique est espacé : chaque caractéristique représente une dimension et les valeurs des caractéristiques sont binaires, (b) le vecteur de caractéristiques est dense : chaque caractéristique principale est représenté comme un vecteur de valeurs réelles. Le mot courant est *dog* ; le mot précédent est *the* ; l'étiquette postérieure précédente est *DET*.

Modèles de Langage neuronaux directs

La catégorie des réseaux de neurones directs (*Feedforward neural network*) comporte des réseaux avec des couches totalement connectées, tels que le perceptron multicouches (*Multi-Layer Perceptron* - MLP). Ce dernier est un classifieur linéaire de type réseau neuronal formel organisé en plusieurs couches au sein desquelles une information circule (dans un seul sens) de la couche d'entrée, via les (la) couches cachées, vers la couche de sortie qui correspond à la sortie du système.

Les modèles de langage à base de réseaux de neurones directs standards de type perceptron multicouches avec une seule couche cachée (*shallow feed-forward neural network*) sont des modèles de langage neuronaux dont les probabilités sont estimées dans un espace légerement continu sont illustrés par la figure 4.5. Introduits par Bengio [Bengio et al.,

11. https ://code.google.com/archive/p/word2vec/
12. Les partie de parole (tel que VER, ADJ, NOM, PRP, etc.) sont obtenus par l'étiquetage morpho-syntaxique (aussi appelé étiquetage grammatical).

2001], [Bengio et al., 2003] puis utilisés par Schwenk [Schwenk, 2007], ces modèles se distinguent des modèles discrets (n-gramme) par leur aptitude à prendre en compte un large contexte. Aussi, le nombre de paramètres de ces modèles ne croît que linéairement avec l'ordre du modèle. En revanche, une limitation de ces modèles est que la taille du vocabulaire de prédiction est limitée afin de rendre le temps de calcul acceptable. Cette limitation rend leurs performances proche au modèles de langage de type n-gramme.

Le problème initial est de calculer la probabilité d'un mot en fonction des $n-1$ mots précédents. Puisqu'on fournit à un réseau des entrées sous forme de vecteurs de caractéristiques (*features*). La première idée sera de donner à chaque mot du vocabulaire une dimension et de le représenter par un vecteur *one-hot*.

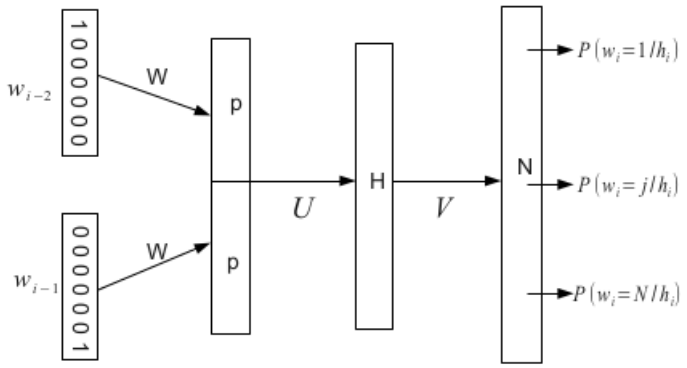

FIGURE 4.5 – Architecture directe d'un modèle de langage neuronal noté NNLM ou CSLM [Bengio et al., 2003].

Le réseau comporte trois couches. D'abord, les mots du contexte (w_1^{n-1}) sont projetés dans un espace continu afin d'obtenir leurs représentations vectorielles. Ces vecteurs sont des paramètres appris en même temps que le reste des données dans le réseau de neurones. Puis, ces vecteurs du contexte sont concaténés pour créer l'entrée de la couche cachée. La fonction d'activation de la couche cachée est non linéaire (*tangent hyperbolique* ou *sigmoid*). Au niveau de la couche de sortie, chaque neurone correspond à la probabilité d'un mot (calculée par la fonction d'activation *softmax* qui permet de transformer les scores de sortie de chaque mot en probabilités, en les normalisant afin de garantir que la somme de toutes les probabilités soit égale à 1). La taille de cette dernière couche est donc égale à la taille du vocabulaire.

Ainsi, les entrées du réseau neuronal sont les indices des $n-1$ mots précédents dans le vocabulaire. Les mots d'entrée sont encodés par 1 - de - N codes. Le mot i^{th} du vocabulaire est codé en plaçant l'élément i^{th} du vecteur à 1 et tout les autres éléments à 0, désignant ce c_l par $l = 1, ..., (n-1).P.$ où P est la taille de la projection :

$$h_j = w_{j-n+1}, ..., w_{j-2}, w_{j-1} \tag{4.18}$$

et les sorties sont les probabilités postérieures de tout les mots du vocabulaire :

$$P(w_j = i|h_j) \quad \forall i \in [1, N] \tag{4.19}$$

où N est la taille du vocabulaire. Les paramètres H et N correspond respectivement à la taille de la couche cachée et à la couche de sortie. Le réseau neuronal effectue certaines opérations, où l'identification de quelques notations est nécessaire pour suivre le séquencement : d_j les activités de la couche cachée :

$$d_j = tanh(\sum_l c_l \; u_{jl} + b_j) \quad \forall j = 1, ..., H \tag{4.20}$$

y_i les sorties du réseau de neurones :

$$y_i = \sum d_j \; v_{ij} + k_i \tag{4.21}$$

$g(y)$ leur normalisation softmax :

$$g(y_i) = \frac{e^{y_i}}{\sum^N e^o} \tag{4.22}$$

où u_{jl} et b_j sont respectivement les poids de la couche cachée et ses biais, v_{ij} et k_i sont les poids de la couche de sortie et ses biais.

Habituellement, l'apprentissage est effectuée avec l'algorithme standard de rétro-propagation (Back-Propagation-BP) [Bottou, 2012]

Algorithme BP ***début***

- *Régler le compteur d'itération $i = 0$,*
- *Les matrices de poids U, V et W sont initialisées avec de petits nombres aléatoires (on utilise la distribution normale avec la moyenne 0 et la variance 0.1).*
  pour *chaque itération* ***faire***
- *Augmentez le compteur de 1*
- *Présenter à la couche d'entrée le mot courant w.*
- *Effectuer une passe avant, comme décrit dans la section précédente pour obtenir couches cachée et de sortie.*
- *Calculer le gradient d'erreur E dans la couche de sortie.*
- *Propager l'erreur à travers le réseau de neurones et changer les poids en conséquence.* ***fin pour***

fin

La fonction d'erreur à minimiser est la suivante :

$$E = \sum_{i=1}^{N} y_i \ log \ P(y_i) + \beta(\sum_{jl} u_{jl}^2 + \sum_{ij} v_{ij}^2) \tag{4.23}$$

où $P(y_i)$ indique la probabilité de sortie souhaitée. La première partie de cette équation (4.23) est l'entropie croisée entre les distributions de probabilité de sortie et de cible et la deuxième partie est un terme de régularisation qui vise à empêcher le réseau de neurones de sur-apprendre les données d'apprentissage.

Modèles de langage neuronaux récurrents

Les réseaux neuronaux récurrents (RNN) sont approximativement ressemblent aux réseaux neuronaux directs (FNN), mais des poids supplémentaires de la couche cachée dans l'étape précédente sont ajoutés. Les neurones cachés du RNN reçoivent des valeurs d'entrée à la fois des neurones d'entrée et des neurones cachés. Cependant les réseaux neuronaux directs ne reçoivent les valeurs d'entrée que depuis les neurones d'entrée.

La formulation RNN la plus simple, connue sous le nom de réseau de Elman ou RNN simple (S-RNN), était Proposé par Elman (1990) et exploré pour être utilisée dans la modélisation du langage par Mikolov [Mikolov et al., 2010]. Le simple RNN respecte l'architecture illustrée dans la figure 4.6.

Le contexte est étendue à une taille indéfinie (ou théoriquement infinie) en utilisant une version récurrente de réseaux neuronaux qui peut gérer des longueurs de contexte arbitraires. Les représentations de mots considérées par un modèle de langage de réseau neuronal récurrent (RNNLM) sont proposées par Mikolov [Mikolov et al., 2010]. L'architecture se compose d'une couche d'entrée, une couche cachée avec des connexions récurrentes et une couche de sortie. Le vecteur d'entrée $w(t)$ représente le mot d'entrée à

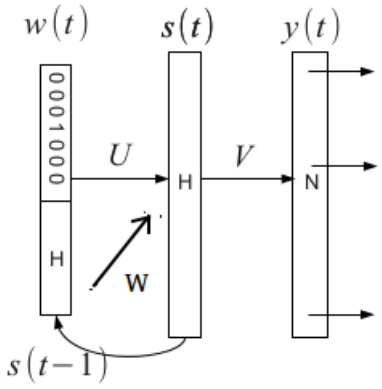

FIGURE 4.6 – Modèle de langage à base de réseau de neurones récurent introduit par [Mikolov et al., 2010].

l'instant t codé en utilisant 1 -de- N codes, et la couche de sortie $y(t)$ produit une distribution de probabilité sur les mots. La couche cachée $s(t)$ maintient une représentation de l'histoire de la phrase. Le vecteur d'entrée $w(t)$ et le vecteur de sortie $y(t)$ ont la dimension du vocabulaire. Les entrées et les sorties des couches cachées et de sortie sont calculées par les formules suivantes [Mikolov et al., 2013b] :

$$s_j(t) = f(\sum_i w_i(t)u_{ji} + \sum_l s_l(t-1)w_{jl}) \tag{4.24}$$

$$y_k(t) = g(\sum_j s_j(t)v_{kj}) \tag{4.25}$$

où $f(z)$ est une fonction d'activation sigmoide :

$$f(z) = \frac{1}{1+e^{-z}} \tag{4.26}$$

et $g(z)$ est une fonction softmax, qui assure les sorties en probabilités :

$$g(z_m) = \frac{e^{z_m}}{\sum_k^N e^{z_k}} \tag{4.27}$$

Les baises ne sont pas utilisés puisqu'ils n'ajoutent empériquement aucune amélioration à l'apprentissage. Les réseaux neuronaux récurrents (RNN) sont habituellement entrainés par l'algorithme de réto-propagation à travers le temps (*Back-Propagation Through Time* - BPTT) [Werbos, 1990].

L'apprentissage du RNN pour une époque est effectuée comme suit :

Algorithme BPTT ***début***

- *Régler le compteur de temps t = 0, initialiser l'état des neurones dans la couche cachée $s(t)$ à 1 ;*

- *Les matrices de poids U, V et W sont initialisées avec de petits nombres aléatoires (on utilise la distribution normale avec la moyenne 0 et la variance 0.1)* ;
  pour *chaque époque* ***faire***
- *Augmentez le compteur de temps t de 1* ;
- *Présenter à la couche d'entrée $w(t)$ le mot courant w_t* ;

— *Copiez l'état de couche cachée* $s(t-1)$ *dans le couche d'entrée* ;
— *Effectuer une passe avant, comme décrit dans la section précédente pour obtenir* $s(t)$ *et* $y(t)$;
— *Calculer le gradient d'erreur* $e(t)$ *dans la couche de sortie* ;
— *Propager l'erreur à travers le réseau de neurones et changer les poids en conséquence* ; ***fin pour***

fin

La fonction objective que nous visons à maximiser est la vraisemblance des données d'apprentissage :

$$f(\lambda) = \sum_{t=1}^{T} logy_{lt}(t) \tag{4.28}$$

où les exemples d'apprentissage sont étiquetés pas $t = 1$ et l_t est l'indice du $^{t-me}$ mot correct. Le Gradient du vecteur d'erreur de la couche de sortie est $e_y(t)$ calculé par le critère de l'entropie croisée (au lieu du l'erreur moyenne quadratique) et permet de maximiser la vraisemblence de la classe correcte, selon l'équation :

$$e_y(t) = d(t) - y(t) \tag{4.29}$$

où $d(t)$ est le vecteur cible qui représente le mot $w(t+1)$ qui doit être prédit (encodé en *one-hot* présentation).

Les réseaux récurrents reprennent l'état caché précédent en compte pour calculer l'état caché présent. Ils sont bien adaptés à la structure de séquence naturelle dans le langage. Sachant qu'ils sont difficiles à entraîner, ils ont fait l'objet de nombreuses améliorations notamment pour étendre leur capacité à mémoriser des informations. Plusieurs variantes sont proposées et utilisées depuis quelques années telles que : LSTM (*Long Short Term Memory networks*), GRUs (*Gated Recurrent Units*), etc.

Malheureusement, malgré la force et la qualité de ces modèles neuronaux directs, récurrents, ou autres, le coût computationnel relatif tant à l'apprentissage qu'au test limitent leur diffusion. Le calcul se concentre à la couche de sortie et le temps de calcul est donc linéaire par rapport à la taille du vocabulaire. Pour réduire ce coût, plusieurs solutions sont proposées dans la littérature, comme l'utilisation de liste courte pour le vocabulaire (*short-list*) impliquant les modèles neuronaux uniquement pour les mots les plus fréquents [Schwenk, 2007], ou encore l'usage d'une couche de sortie structurée [Mnih and Hinton, 2009] comme le modèle SOUL (*Structure OUtput Layer*) [Le et al., 2011], la modification du critère d'optimisation [Mnih and Teh, 2012] dans l'algorithme de rétopropagation en introduisant quelques modifications. Aussi, une façon courante de réduire le bruit est d'estimer l'erreur et les gradients sur la base des échantillons (ou mini-lots) de m exemples au lieu d'un seul exemple par itération. Cela donne naissance à l'algorithme de rétopropagation à mini-lots (*SGD minibatch*) [Li et al., 2014], etc.

7 Expérimentations et évaluation

Pour évaluer les différentes approches de modélisation du langage nous avons utilisé deux corpus textuels français et anglais, en utilisant des outils standard et récents de modélisation du langage.

7.1 Corpora et outils utilisés

Les données textuelles utilisées pour la modélisation langagière sont des textes français et anglais utilisés dans des compagnes d'évaluation.

Corpus ETAPE

Le premier corpus utilisé, dans nos expérimentations, est un corpus textuel français (noté corpus ETAPE pour simplification) de la compagne d'évaluation ETAPE [Gravier and Adda]. ETAPE est un projet visant à organiser des campagnes d'évaluation dans le domaine du traitement automatique de la parole pour la langue française. Financé partiellement par l'Agence Nationale de la Recherche française (ANR), le projet réunit des experts français pour l'organisation de campagnes de ce genre sous la direction scientifique de l'Association Francophone de la Communication Parlée (AFCP).

Nous divisons ces données manuellement en trois sous-ensembles pour les tâches de l'apprentissage (ou l'entraînement), le développement et le test, en réspectant les recommendations de [Haton et al., 2006] : 90% des données pour l'apprentissage et 5% pour le test et le développement. Il est intéressant de noter que, étant donné que nos données d'apprentissage sont très proches en matière et en style aux données de test, les résultats devraient donner la vraie performance des modèles, même si la taille des données d'apprentissage est faible.

Un processus de prétraitement [13] est nécessairement appliqué aux données textuelles brutes :

Algorithme Pré-traitement

> **pour** chaque fichier du corpus textuel **faire**éliminer les lignes vides du texte entier ; segmenter (*tokenising*) le texte de telle sorte que chaque phrase est dans une seule ligne avec les symboles début et fin de phrase ($< s >$ et $< /s >$; convertir les caractères majuscules au début de la ligne en minuscules ; traiter les chiffres, les nombres, etc. éliminer tout les éléments non-mot, non-phrase et les signes de ponctuation. **fin** de **pour**

Avec cette tokenisation, la taille de notre premier corpus en termes de phrases et de mots est présentée dans le tableau 4.1. Le nombre de mots ne comprend pas les jetons ajoutés au début et à la fin des phrases.

Corpus	# phrase	# mots
Etape-train	18 083	249 569
Etape-dev.	1 004	10 782
Etape-test	1 004	16 419

Table 4.1 – Taille du corpus ETAPE.

Corpus NAACL

Deuxièmement, nous exploitons les données *nc7* et *eparl7* utilisés lors du NAACL'2012, le septième atelier sur la traduction automatique statistique, pour entrainer les modèles langage en anglais. Le corpus *newstest2010* est utilisé en tant que données

13. Les script *tokenize.perl* et *clean-corpus-n.perl* de l'outil de traduction automatique moses (http ://www.statmt.org/moses/) peut être modifiés et utilisés dans ce cadre.

de développement et le corpus *newstest2011* comme données de test. Ces données (que nous avons noté pour simplification : le corpus NAACL) sont pré-traités avec les outils standard de Moses [Koehn et al., 2007]. La taille totale du corpus en termes de phrases et de mots est présentée dans le tableau 4.2.

L'utilisation de deux (ou plusieurs) corpus d'apprentissage pour les modèles de langage dépend du fait que, si un modèle de langage est entraîné sur les données correspondant au domaine d'intérêt (appelée *in-domain*) qui souffre généralement de pénurie, la formation d'un autre modèle de langage peut bénéficier de données provenant d'autres sources ou d'autres domaines (appelées *non-domain*) de taille plus importante. L'approche conventionnelle consiste à former un modèle de langage distinct sur chaque corpus d'apprentissage individuel (*in-domain* et *non-domain*). Une faible quantité de données du domaine cible (dans ce cas : le corpus *nc7*) est utilisée pour former le premier modèle de langage individuel et une grande quantité de données hors domaine (*eparl7*) est utilises pour former un deuxième modèle de langage. Une combinaison linéaire pondérée (ou appelée en particulier ″interpolation linéaire" [Jelinek, 1980]) est utilisée pour construire le modèle final. Les poids interpolés des deux modèles de langage séparés sont ajustés en optimisant la perplexité des données de développement.

Corpus	*# phrases*	*# mots*
nc7-train	212 517	5 085 447
eparl7-train	2 218 201	59 940 634
newstest2010-dev.	2 489	61 924
newstest2011-test	3 003	74 833

TABLE 4.2 – Taille du corpus NAACL.

Boites à outils utilisées

Tout les modèles de langage n-gramme sont construits avec la boîte à outils *SRILM* [Stolcke, 2002],[Stolcke et al., 2011]. Ce paquetage est développé, maintenu et distribué sous une libre licence par SRI, le laboratoire international de technologie et de recherche de la parole en Californie. Cette boîte à outils n'est pas disponible en précompilation et doit être compilée et installée manuellement. La plupart des recherches et des développements de la modélisation des langages au SRI et dans différentes universités dans le monde sont basés sur le SRILM [Madnani, 2009].

Tout nos modèles de langage spatial continus NNLM sont construits avec la boite à outils libre *CSLM*. Ce paquetage programmé en $C++$ est modulaire et repose sur des bibliothèques mathématiques hautement optimisées pour les parties intensives de calcul telles que BLAS (*Basic Linear Algebra Subprograms*), y compris le support possible des cartes GPU. Plusieurs réseaux de neurones avec différentes architectures peuvent être utilisés. Les blocs de construction de base sont les machines linéaires, sigmoïdes, tanh et softmax bien connues. Chaque machine (ou réseau de neurones) fournit une fonction pour propager l'entrée à la sortie et pour rétro-propager à nouveau l'erreur dans la direction opposée [Schwenk, 2013].

Cependant, tout les modèles de langage neuronaux récurrents RNNLM sont entraînés avec la boîte à outils libre de Mikolv *RNNLM* [Mikolov et al., 2011]. Il s'agit d'un outil ouvert et librement disponible pour l'apprentissage des modèles de langage statistiques basés sur des réseaux neuronaux récurrents et des modèles d'entropie maximale. Il est écrit en $C/C++$, simple à installer (et à utiliser). Il ne dépend pas des bibliothèques externes

(telle que BLAS) et comprend des techniques pour réduire la complexité informatique (classes dans la couche de sortie et connexions directes entre la couche d'entrée et la couche de sortie).

7.2 Estimation et lissage des ML n-gramme

L'ordre des modèles n-gramme est un paramètre important qui affecte leur performance. De ce fait, nous avons commencé par chercher le meilleur ordre n pour nos contextes relatifs aux deux corpora ETAPE et NAACL. Plusieurs expérimentations sont faites avec différentes valeurs pour l'ordre n des n-grammes pour $\{1, 2, 3, 4, 5\}$, en utilisant les paramètres par défaut de SRILM, ce qui correspond au lissage de *Good-Turing*. Nous évaluons les performances des différents modèles de langage en mesurant les perplexités sur les données de développement qui sont reportées dans les tableaux 4.3 et 4.4.

Selon les résultats obtenus, nous constatons que l'utilisation de 3 ou 4-grammes est adéquate et donne de bonnes perplexités. Nous choisissons de fixer n à 4 dans les expérimentations présentées dans les prochaines sections.

n-gram	*ppl_ dev*	*ppl_ test*
1	950.5	1049.9
2	402.9	413.0
3	**379.2**	387.3
4	**380.3**	387.7
5	380.7	387.8

TABLE 4.3 – Perplexité pour les differents ordres n pour le corpus ETAPE.

n-gram	*ML*	*ppl_ dev*	*ppl_ test*	*poids*
1	nc7	2 246,3	2 285,4	0,63
	eparl7	2 415,5	2 457,8	0,37
		1 977,5	1 9996,1	1
2	nc7	808,4	859,8	0,53
	eparl7	795,9	887,6	0,47
		578.2	627.0	1
3	nc7	754,9	799,7	0,51
	eparl7	719,3	813,7	0,49
		496,4	542,0	1
4	nc7	**755,6**	801,9	0,51
	eparl7	714,0	808,0	0,49
		487,4	533,6	1
5	nc7	757,2	803,1	0,51
	eparl7	715,1	809,4	0,49
		487,4	533,6	1

TABLE 4.4 – Perplexité pour les différents ordres n pour le corpus NAACL.

Vocabulaire et mots hors-vocabulaire

Les auteurs dans [Mezzoudj et al., 2015b], n'ont pas utilisé explicitement le fichier de vocabulaire pour l'apprentissage des modèles de langage, un modèle de langage à

vocabulaire ouvert est utilisé (*open-vocabulary*) en utilisant la configuration de SRILM par défaut. Cependant, un système de reconnaissance vocale automatique et les modèles de langage relatifs ont besoin du fichier de vocabulaire comme paramètre important qui sera utilisé (après l' ajout de phonétisation) pour la modélisation acoustique. Dans cette section, nous considérons cette situation. Deux fichiers de vocabulaire sont générés de façon indépendante et explicite (en utilisant le modèle unigramme tel qu'il est proposé dans le tutoriel de la boîte à outils CSLM [Schwenk, 2013]) à partir des données des deux corpora respectivement ETAPE et NAACL.

Pour le corpus ETAPE, nous générons un fichier de vocabulaire de 15 662 mots, en utilisant les données d'apprentissage (Etape-train), de développement (Etape-dev) et de test (Etape-test). Naturellement, la taille du vocabulaire croît avec la taille du corpus dont il est extrait. De même pour le corpus NAACL, nous générons un fichier de vocabulaire de 166 952 mots, en utilisant les données d'apprentissage disponibles (nc7-train et eparl7-train), de développement (newtest2010-dev) et de test (newstest2011-test).

A cette étape, nous montrons l'influence de l'utilisation du vocabulaire (ouvert ou fermé) sur les modèles de langage lissés par les méthodes les plus couramment utilisées : l'algorithme de *Modifié interpolé de Kneser-Ney* et l'algorithme *modifié backoff de Kneser-Ney*. Les résultats sont rapportés dans les tableaux 4.5 et 4.6.

D'autres expérimentations sont effectuées avec d'autres méthodes de lissage, telles que *Absolute-discount* et *Witten-Bell*, mais les résultats ne sont pas représentés explicitement, et seules quelques inférences et remarques relatives sont présentées par la suite.

Méthode	*Interp-Modif-KN*		*Backoff-Modif-KN*	
	ppl_ dev	*ppl_ test*	*ppl_ dev*	*ppl_ test*
OOV,Vocab.	242,5	244,6	221,2	**221,3**
OOV,~~Vocab.~~	241,7	244,3	159,6	**157,6**
~~OOV~~,Vocab.	242,2	**164,7**	221,0	184,6
~~OOV~~,~~Vocab.~~	163,2	**161,3**	175,7	173,8

TABLE 4.5 – Utilisation du vocabulaire et du *UNK* pour les OOV avec le corpus ETAPE.

Méthode	*Interp-Modif-KN*			*Backoff-Modif-KN*		
	ML	*ppl_ dev*	*ppl_ test*	*poids*	*ppl_ dev*	*ppl_ test*	*poids*
OOV,Vocab.	nc7	477,7	501,5	0,51	478,1	496,6	0,45
	eparl7	451,4	510,9	0.49	410,4	455,4	0,55
	All	343,8	376,4	1	338,2	**365,8**	1
OOV,~~Vocab.~~	nc7	307,3	314,8	0,52	310,2	314,9	0,49
	eparl7	326,8	370,4	0,48	308,3	341,8	0,51
	All	261,4	286,7	1	264,0	**285,7**	1
~~OOV~~,Vocab.	nc7	477,7	391,6	0,49	478,1	399,4	0,44
	eparl7	451,4	385,2	0,51	410,4	367,2	0,56
	All	344,3	**288,3**	1	338,1	293,7	1
~~OOV~~,~~Vocab.~~	nc7	315,5	325,3	0,47	327,1	334,2	0,44
	eparl7	319,9	365,1	0,53	311,6	346,5	0,56
	All	260,7	**287,3**	1	286,2	291,0	1

TABLE 4.6 – Utilisation du Vocabulaire et du *UNK* pour les OOV pour le corpus NAACL.

Selon la littérature, les méthodes de *backoff* ne fonctionnent pas bien mais c'est plutôt les modèles interpolés qui combinent toujours la distribution d'ordre supérieur et inférieur

qui sont plus intéressants [Chen and Goodman, 1996]. De plus, dans le didacticiel fourni avec la boîte à outils CSLM [Schwenk, 2013], les modèles de langage obtenus avec la méthode de *Kneser-Ney modifiée et interpolée* sont considérés comme les modèles de base (donc les meilleurs pour les modèles n-gramme lissés).

Cependant, nous avons rencontré une situation étrange après plusieurs expérimentations effectuées sur les deux corpora, tel qu'il est présenté dans les tableaux 4.5 et 4.6. Lorsque nous n'utilisons pas explicitement le fichier du vocabulaire et nous ne considérons pas les mots hors vocabulaire *OOV* pour l'entraînement de nos modèles de langage, nous trouvons que les meilleurs résultats sont obtenus par l'algorithme *modifié interpolé de Kneser-Ney*. Cependant, si nous utilisons le fichier de vocabulaire et nous considérons les *OOV* (qui correspond dans les commandes SRILM au options : *-vocab -unk*), les meilleurs résultats sont obtenus avec l'algorithme *modifié backoff de Kneser-Ney*.

De plus lorsque nous utilisons le vocabulaire et les *OOV* lors de l'apprentissage, l'algorithme *non-modifié backoff de Kneser-Ney* (ou appelé *original*) donne de meilleurs résultats que l'algorithme *non-modifié interpolé de Kneser-Ney* sur les deux corpora contrairement à ce que nous obtenons si nous ôtons le vocabulaire.

Dans [Mezzoudj et al., 2015b], les auteurs ont remarqué que les méthodes de lissage interpolées fonctionnent bien comme il est connu dans la littérature [Chen and Goodman, 1996]. Cependant, pour plus de précision, il est essentiel de noter que cette situation est juste sans l'utilisation explicite du vocabulaire et de l'étiquette *UNK* pour les OOV (via la boîte à outil SRILM). Sachant que ces deux paramètres sont essentiels pour les modèles de langage intégrés dans de réels systèmes de reconnaissance, alors nous utilisons ces deux options dans toutes les expériences présentées dans les sections suivantes.

Sélection des Hyper-paramètres du lissage

Pour le corpus ETAPE, nous entraînons des modèles de langage 4-gramme en utilisant différentes méthodes de lissage telles que le *lissage additif* et le *décompte absolu* qui sont liées à des hyper-paramètres, nous effectuons des validations croisées en utilisant les données de développement pour la sélection des modèles. Les résultats de cette phase sont rapportés dans le tableau 4.7.

Pour le *lissage additif*, Lidstone et Jeffreys [Chen and Goodman, 1999] prennent $\delta = 1$ comme valeur idéale. Selon nos expérimentations, nous constatons que $\delta = 0,007$ donne de meilleur résultat.

δ	*ppl_ dev*
0,001	497,85
0,003	449,79
0,005	433,98
0,006	431,29
0,007	**430,32**
0,008	430,36
0,009	431,01
1	1504,27

TABLE 4.7 – Optimisation des paramètres de le lissage additif avec les données de développement ETAPE.

Le modèle d'interpolation par *décompte absolue de Ney* utilise le paramètre λ comme étant la constante de soustraction dans la formule 4.14. La valeur de cet hyper-paramètre

affecte la performance du modèle de langage. pour une valeur $\lambda = 0,9$, la performance du modèle de langage est la meilleure avec une perplexité de $294,58$. Quelques résultats de la validation croisée pour cet hyper-paramètre sont présentés dans le tableau 4.8.

λ	*ppl_ dev*
0,1	549,45
0,2	436,85
0,3	383,47
0,4	350,93
0,5	328,93
0,6	313,44
0,7	302,67
0,8	296,04
0,9	**294,59**
1	320,90

TABLE 4.8 – Influence du hyper-paramètre du décompte absolu interpolé avec les données de développement ETAPE.

De même pour le corpus NAACL, les méthodes telles que le *lissage additif* et *décompte absolu* qui sont liées à un hyper-paramètre, nécessitent aussi une validation croisée en utilisant des données de développement pour sélectionner les meilleurs hyper-paramètres. Les résultats de ces validations sont présentés dans les tableaux 4.9 et 4.10.

Pour le *lissage additif*, la valeur de $\delta = 0,0005$ donne le meilleur modèle de langage avec une perplexité de $627,57$.

δ	*ppl_ dev*
0,0001	683,38
0,0003	633,26
0,0005	**627,57**
0,0007	632,59
0,001	642,40
0,003	715,43

TABLE 4.9 – Optimisation du hyper-paramètre du lissage additif sur les données de développement NAACL.

L'hyper-paramètre pour le *décompte absolu interpolé* est également ajusté en utilisant la validation croisée. La performance du modèle de langage avec $\lambda = 0,7$ est la meilleure et donne une perplexité de $451,15$ au modèle de langage correspondant. Les résultats de cette phase sont présentés dans le tableau 4.10.

La méthode de *Good-Turing* est influencée par les paramètres de coupure appellés aussi *cutoff* (en utilisant les commandes SRILM *-gtNmin* ou *-gtN max*) qui précise la façon dont les n-grammes sont réduits. Ce type de paramétrage est disponible pour les versions interpolées des méthodes de *Witten-Bell*, *Kneser-Ney non-modifié* et *Kneser-Ney modifié*. D'autres méthodes, telles que les versions backoff de *Witten-Bell*, *Kneser-Ney* et *Kneser-Ney modifié* sont libres de tout hyper-paramètres.

λ	*ppl_ dev*
0,1	490,31
0,2	487,57
0,3	481,54
0,4	477,42
0,5	464,84
0,6	455,96
0,7	**451,15**
0,8	453,19
0,9	456,54

TABLE 4.10 – Optimisation du hyper-paramètre du décompte absolu sur les données de développement NAACL.

7.3 Modèles de Langage n-gramme

Corpus ETAPE

Nous générons différents modèles de langage 4-gramme, avec différentes méthodes de lissage, en considérant le fichier de vocabulaire, les mots hors vocabulaire *OOV* et les valeurs optimisées des hyper-paramètres correspondants. On note que la méthode standard utilisée par défaut avec l'outil SRILM (si aucune méthode de lissage n'est spécifiée) est la méthode de *Good-Turing* combinée avec la méthode *Katz*. Nous avons utilisé cette méthode standard sans changer les paramètres de coupure définis par défaut, la perplexité du modèle de langage généré correspond à $387,7$.

Cependant, si nous l'utilisons avec les paramètres de coupure tels que $-gt1min\ 1\ -gt1max\ 3\ -gt2min\ 1\ -gt2max\ 3\ -gt3min\ 1\ gt3max\ 3\ -gt4min\ 2\ -gt4max\ 3$, la perplexité du modèle de langage diminue à $258,34$. Ce résultat correspond à la deuxième ligne du tableau 4.11. Avec des paramètres de coupure optimisés qui correspond à $-gt1min\ 1\ -gt1max\ 7\ -gt2min\ 1\ -gt2max\ 7\ -gt3min\ 1\ -gt3max\ 7\ -gt4min\ 2\ -gt4max\ 7$, la perplexité du modèle diminue encore jusqu'à $250,10$. Ce dernier résultat correspond à la troisième ligne du tableau 4.11.

Ces résultats montrent que le fait d'ajuster les paramètres de coupure peut améliorer la perplexité des modèles correspondant, ce qui concorde avec l'hypothèse et les expériences de [Sundermeyer et al., 2011].

Selon les expérimentations réalisées, nous remarquons que le *décompte absolu backoff* donne des résultats significativement meilleurs que le *décompte absolu interpolé*. De même la méthode du *Witten-Bell backoff* est significativement meilleure en terme de lissage que la méthode *Witten-Bell interpolée* avec l'utilisation explicite du vocabulaire.

Le résultat le plus important, qui se manifeste dans le tableau 4.11, est qu'avec les versions de *backoff de Kneser-Ney* nous obtenons les meilleurs modèles de langage avec des perplexités de $220,1$ et $221,3$ et non pas avec les versions interpolées de l'algorithme de *Kneser-Ney modifié* qui donne une perplexité de $244,6$.

Corpus NAACL

Nous reprenons le même plan d'expérimentation sur les données du corpus anglais NAACL. Nous formons des modèles de langage 4-gramme sur les données d'apprentissage (du domaine et du non-domaine). En utilisant l'interpolation linéaire, nous obtenons un modèle de langage interpolé unique (tel qu'il est montré dans le tableau 4.6). Pour un

Méthode	λ, δ	*ppl_ dev*	*ppl_ test*
Good-Turing_ nonParamètre		380,3	387,7
Good-Turing_ standard		253,3	252,9
Good-Turing_ optimisé		247,7	247,6
Lissage addtive	,007	430,3	448,5
Backoff-décompte absolu	0,8	242,8	242,5
Interp-décompte absolu	0,9	294,6	298,8
Backoff-Witten-Bell		237,6	238,4
Interp-Witten-Bell		334,1	337,7
Ristad décompte naturel		294,2	293,9
Backoff-nonModifié-KN		219,5	**220,1**
Interp-nonModifié-KN		255,6	258,4
Backoff-Modifié-KN		221,2	**221,3**
Interp-Modifié-KN		242,5	244,6

TABLE 4.11 – Perplexité des ML avec différents lissages sur les données de développement et de test (ETAPE).

but de simplification, seules les perplexités des modèles de langage interpolés finaux sont exposés dans le tableau 4.12. Chaque fois que nous utilisons une méthode de lissage différente, nous en considérons le fichier de vocabulaire adéquat, les OOV et les valeurs des hyper-paramètres optimisés du lissage.

Méthode	λ, δ	*ppl_ dev*	*ppl_ test*
GoodTuring_ nonParamtère		487,4	533,6
GoodTuring_ standard		413,5	453,9
Goodturing_ optimisé		409,8	450,1
Lissage additif	.0005	626,1	682,1
Backoff-décompte absolu	0,7	406,3	443,6
Interp-décompte absolu	0,7	454,9	502,7
Backoff- Witten-Bell		387,5	421,4
Interp-Witten-Bell		458,4	510,5
Ristad NaturelDiscount		492,6	542,6
Backoff-nonmodified-KN		338,2	367,3
Interpo.Unmodified-KN		356,4	392,3
Backoff Modified-KN		338,2	**365,8**
Interpo.Modified-KN		343,8	376,4

TABLE 4.12 – Perplexités des MLs avec différents lissages sur les données de développement et de test NAACL.

Nous confirmons que certaines méthodes telles que le *lissage additif*, *décompte naturel de Ristad* ou l'algorithme de *Witten-Bell interpolé* ont des performances faibles, elles donnent des modèles de langage avec respectivement les perplexités de 682,1, 542,6 et 510,5.

Encore une fois, nous remarquons selon les résultats reportés dans le tableau 4.12 que la version modifiée *Kneser-Ney backoff* génere le meilleur modèle de langage avec une perplexité de 365,8 tandis que la perplexité obtenue par l'algorithme *interpolé modifié de Kneser-Ney* est de 376,4.

En général, les versions de lissage à base de repli (*Backoff*) fonctionnent bien lorsque

nous incluons explicitement le fichier de vocabulaire lors de l'apprentissage des modèles de langage n-gramme, ce qui peut être relatif aux implémentations proposées dans la boîte à outils SRILM [Stolcke, 2002]. Le fait de limiter le vocabulaire à l'aide de *-ngram-count* peut directement troubler les fréquences de comptage car nous affectons les n-gramme à faible fréquence au mots hors vocabulaire *OOV*. Probablement, cette situation affecte l'interpolation de l'algorithme de *Kneser-Ney* d'une mauvaise manière.

Enfin, nous constatons que les modèles de langage 4-gramme appris avec le lissage *backoff* de Kneser-Ney modifié sur les données d'ETAPE et de NAACL représentent les meilleurs modèles de langage. Nous les considérons comme des modèles de base pour le reste de nos expérimentations.

7.4 Modèles de Langage neuronaux

Dans cette section, nous évaluons et nous comparons les modèles de langage à base de réseaux de neurones directs (*feed-forward neural network*) à espace de représentation continu (*Continous Space Language Model*) notés CSLM et les modèles de langage à base de réseaux de neurones reccurents (*Reccurent Neural Network Language Model*) notés RNNLM, entrainés sur les deux corpora textuels ETAPE et NAACL.

Tout les modèles de langage neuronaux directes (CSLM) sont construits via la boîte à outils libre CSLM [Schwenk, 2013]. Le modèle de langage CSLM est implémenté via un réseau de neurones multicouches contenant : la couche d'entrée (première couche) qui projette tout les mots dans le contexte h_i vers la couche de projection (deuxième couche). La couche cachée est la troisième couche d'apprentissage. Enfin, la couche de sortie (quatrième couche) finalise l'estimation non linéaire des probabilités.

Le CSLM est capable de calculer les probabilités de tout les mots du vocabulaire. Cependant, en raison de la complexité de calcul trop élevée, le CSLM est principalement programmé (court-circuité) pour calculer les probabilités d'un sous-ensemble du vocabulaire. Le sous-ensemble considéré est appelé liste courte (*short-list*), qui se compose des mots les plus fréquents dans le vocabulaire. Le CSLM redistribue la masse des probabilités calculées de tout les mots de la liste courte, à l'aide du modèle de langage n-gramme standard sur le reste des mots.

Le choix des hyper-paramètres du réseau de neurones paramétrable sont configurés manuellement : type de fonction d'activation, choix de l'architecture (nombre de couches cachées, nombre de neurones par couche), taux d'apprentissage, nombre d'époques (ou itérations), les caractéristiques présentées sur la couche d'entrée (exp. représentation du mot), etc.

Cette phase semble compliquée au début, la meilleure façon de procéder est de réutiliser la configuration existante proposée avec le tutoriel [14] de la boîte à outil CSLM et puis d'essayer par la suite d'introduire des modifications appropriées. De nombreuses architectures des réseaux neuronaux (simple, profonde et large) existent et peuvent être utilisées. Cette flexibilité est une propriété très puissante pour les modèles de langage CSLM et qui peut conduire à de grandes améliorations dans la précision par rapport aux modèle de langage neuronal de base.

Cependant, tout les modèles de langage à base de réseaux neuronaux récurrents RNNLM sont construits en utilisant la boîte à outils libre de Mikolv [Mikolov et al., 2011]. Vu la difficulté d'entraînement de ces modèles de langage, nous allons se contenté d'entraîner les différente architectures de réseaux de neurones en s'inspirant des conseils

14. http ://www-lium.univ-lemans.fr/cslm/, re-consulté en 22 avril 2018.

présentés dans les boîtes à outils utilisés [15], sans passer par la validation croisée pour le choix de tout les paramètres des réseaux de neurones peut être très coûteuse. Cette stratégie que nous adaptons est la plus utilisée dans la littérature.

Corpus ETAPE

Les modèles de langage CSLM sont entraînés sur les données d'apprentissage du corpus ETAPE avec l'outil CSLM en se basant sur les modèles de langage standards 4-gramme. Nous avons utilisé les trois types d'architectures recommandés dans la boîte à outils : le CSLM simple (avec une couche cachée), le CSLM large (nombre important de neurones par couche) et le CSLM profond (avec trois couches cachées).

Les modèles CSLM simples sont entraînés pendant 10 époques (ou itérations). Les paramètres du réseau de neurones correspondant sont comme suit : la dimension de la couche d'entrée est de 15 662 neurones qui correspond à la taille de vocabulaire, la couche de projection a une dimension de 256 pour chaque mot, suivie d'une couche cachée *tanh* de dimension 768 x 192 et d'une couche de sortie *softmax* de 1024 neurones qui correspond à la taille de la liste courte (*shortlist*).

Les modèles de langage CSLM larges sont entraînés aussi pendant 10 époques (itérations). La couche de projection a une dimension de 256 neurones, suivie d'une couche cachée *tanh* de dimension 768 x 192 et d'une couche de sortie *softmax* de dimension 8192 qui correspond à taille de la liste courte considérée.

Les modèles CSLM profonds sont entraînés pendant 10 époques. La couche de projection a une dimension de 256, suivie de trois couches cachées *tanh* de dimensions (768 x 512), (512 x 256) et (256 x 192) respectivement et d'une couche de sortie *softmax* de dimension 8192, qui correspond à la taille de la liste courte considérée.

Nous avons espéré pouvoir utiliser des architectures plus profondes et plus larges avec des quantités de données plus importantes mais malheureusement, nous sommes limités par la capacité du matériel disponible (Intel Core i5-2430M CPU 2.40GHz 4 avec 5.9 Go de RAM , 62.5 Go DD et aucune GPU considérée).

L'apprentissage (et le test) des modèles de langage CSLM avec les données ETAPE ont duré respectivement 30 min, 75 min et 49 min pour les architectures des réseaux neuronaux simple, large et profond.

Pour les modèles de langage à base de réseaux de neurones récurrents, nous avons créé des modèles de langage RNNLM avec 100 neurones dans la couche cachée. Selon le tutoriel correspondant, la taille optimale pour la couche cachée dépend de la taille des données d'apprentissage - pour moins de 1M mots, 50-200 neurones est généralement suffisant, pour 1M-10M mots utiliser 200-300 neurones. Le fait d'utiliser une couche cachée plus grande n'améliore pas la performance, mais rend la progression de l'apprentissage plus lente. Nous utilisons 100 classes pour accélérer l'apprentissage. L'algorithme BPTT est utilisé en mode mini-lots avec une taille de 10 bloc pour au moins 4 étapes. L'utilisation des mini-lots [Ruder, 2016] avec un lot de petite taille et avec échantillons stochastiquement sélectionnés des données afin d'améliorer la performances de l'algorithme de rétropropagation est recommendée.

Pour l'apprentissage des modèles de langage RNNLM formés avec les données ETAPE, l'expérimentation a duré 101 min pour les étapes d'apprentissage et de test, comme présenté dans le tableau 4.13.

Selon les résultats obtenus des évaluations, les perplexités des CSLM pour les trois architectures simple, large et profonde sont respectivement de 295.6, 455.3 et 272.7 comme

15. http ://www.fit.vutbr.cz/ imikolov/rnnlm/, consulté en 22 avril 2018.

ML	*ppl_ dev*	*ppl_ test*	*Temps[min]*
Backoff Modified-KN ML	221.2	221.3	15
Simple CSLM	278.2	295.6	30
Large CSLM	432.6	455.3	75
Deep CSLM	253.3	272.7	49
RNLM	159.4	**153.6**	101

TABLE 4.13 – Perplexité du ML sur les données de développement et de test (ETAPE).

présenté dans le tableau 4.13. Ces résultats ne sont pas compétitives, en les comparant avec les modèles de langage 4-gramme obtenus avec les méthodes de lissage de repli (*backoff*) de kneser-Ney. Ce fait est peut-être dû à la faible quantité de données, vue insuffisante pour assurer un bon apprentissage au réseau neuronal direct gourmand que nous avons utilisé. Cependant, le RNNLM a la plus faible perplexité de 153.6 en le comparant à tout les autres modèles de langage CSLM et 4-gramme.

Ces premières expériences ont prouvé que les RNNLM peuvent être concurrentiels au modèles de langage CSLM et les modèles n-gramme backoff, formés sur les mêmes données d'apprentissage et le même vocabulaire.

Corpus NAACL

Nous reprenons le même type d'expérimentations sur les données NAACL. L'apprentissage du CSLM avec l'architecture simple a duré plus de deux jours, la perplexité du modèle de langage résultant est de $356,4$. Les CSLM avec les architectures large et profonde sont plus coûteux en terme de temps d'apprentissage.

En utilisant les deux sous-ensembles d'apprentissage de NAACL (données du domaine et du non-domaine), nous avons aussi formé deux RNNLM. Le modèle de langage interpolé RNNLM présenté dans le tableau 4.14 est construit à partir de ces deux modèles de langage individuels. Cet apprentissage a duré presque cinq jours et demi avec une architecture récurrente simple. La perplexité du RNNLM correspondant est de 264.9 qui représente le meilleur résultat.

Malgré que les RNNLM donnent de bons résultats, ils souffrent malheureusement d'inconvénients qui limitent leur utilisation comme la lenteur en temps d'apprentissage.

ML	*ppl_ dev*	*ppl_ test*	*Temps[h]*
Backoff Modified-KN ML	338,2	365,8	$\simeq$ 1
Simple CSLM	327,9	356,4	49
Large CSLM	331,8	359,5	90
Deep CSLM	334,4	362,5	72
RNNLM	248,8	**264,9**	133

TABLE 4.14 – Perplexité du ML sur les données de développement et de test NAACL.

Le fait de ne pas traiter les domaines de données clairsemés est un inconvénient inhérent à tout les modèles de langage qui sont estimés dans un espace discret. La nature discrète de tels modèles de langage rend difficile l'obtention de niveaux élevés de généralisation même après l'application des techniques de lissage les plus efficaces, comme le lissage modifié de Kneser-Ney pour les modèles de langage n-gramme *backoff*.

Les résultats obtenus dans les tableaux (4.13 et 4.14) peuvent être justifiés par le fait que, avec les modèles de langage à base des réseaux de neurones directs (CSLM),

l'historique du mot est représenté par le contexte de $n-1$ mots. Il est limité d'une manière proche à ce que nous reprochons aux modèles n-grammes backoff, surtout quand on utilise une faible quatité de données pour l'apprentissage. En général, avec un modèle de langage CSLM, un n-gramme invisible est généralement basé sur la distance et la similarité entre les mots dans un espace continu. Si les mots dans un n-gramme non-vu durant l'apprentissage sont assez proches des mots vus, une probabilité similaire est attribuée.

Cependant, l'histoire dans les réseaux récurrents est représenté par les neurones avec connexions récurrentes, ce qui rend la longueur de l'histoire du mot théoriquement illimitée. Ce phénomene permet d'exploiter au maximum la quatité disponible des données. Donc, les RNN peuvent apprendre des contextes de longueur potentiellement infinie en utilisant une couche cachée connectée de façon récurrente.

8 Conclusion

La modélisation statistique du langage est l'une des tâches les plus importantes du traitement automatique de langage et de la reconnaissance de la parole. Les modèles linguistiques (ou de langage) sont au cœur des systèmes de la reconnaissance automatique de parole à grand vocabulaire et de nombreuses autres applications.

Avec le modèle de langage n-gramme qui est considéré comme standard, la probabilité d'une séquence de mots est approximée au produit des probabilités conditionnelles de chaque mot dans la séquence conditionnée aux $n-1$ mots précédents. Un n-grammes invisible est généralisé par des techniques de lissage par repli (*Backoff*) ou interpolation avec le modèle n-gramme d'ordre inférieur.

Le lissage est une technique fondamentale qui contribue au problème de la réduction de la rareté des données (*data sparseness*). Malheureusement, le modèle de langage n-gramme est défini dans un espace binaire discret d'indices de mots sans aucun moyen d'apprendre les similarités entre les mots. Cependant, les modèles de langage à base de réseaux neuronaux qui sont introduits, apprennent des représentations de mots par des valeurs continues distribuées pour calculer la probabilité d'un mot donné. Ce type de modèles de langage qualifiés de *continus* contribuent à résoudre efficacement le problème de la rarté des données.

Dans ce chapitre, nous avons présenté un état de l'art sur les modèles de langage n-gramme, leurs méthodes de lissage, les modèles de langage avancées et ceux à base de réseaux de neurones, etc. Une analyse du comportement des différents modèles de langage basés sur les n-grammes et les réseaux de neurones (directs et récurrents) en utilisant deux corpus différents (ETAPE et NAACL) de différentes langues (Francais et Anglais) est réalisée et présentée. Les expériences préliminaires sur les données textuelles ETAPE et NAACL montrent que les modèles de langage n-gramme standard demeurent remarquables pour leur simplicité, leur rapidité de calcul et leur puissance. En outre, les modèles CSLM à base de réseaux de neurones direct peuvent être utiles pour la modélisation du langage mais nécessitent une quantité de données très importantes pour assurer un bon apprentissage (qui était difficile pour notre cas). Bien que, les meilleurs résultats sont obtenus avec les modèle RNNLM à base de réseaux neuronaux récurrents.

En utilisant les réseaux neuronales directs qui sont gourmands en terme de données pour l'apprentissage de nos modèles de langage CSLM, les résultats obtenus sont sûrement influencés (négativement) par la quantité des données. Cependant, il semble que la force des modèles de réseaux neuronaux récurrents (RNNLM) est basée sur la réutilisation

de l'information résidante dans les couches précédentes, ce qui améliore la qualité de la représentation continue des mots.

Malheureusement, les modèles de langage neuronaux sont évidemment lents à former et à tester ; quelques jours sont nécessaires pour ces étapes. Malgré leur force et leur qualité, ce coût computationnel relativement élevé tant à l'apprentissage qu'au test peut limiter leur diffusion sauf au grandes compagnies et laboratoires. Nous avons publié une étude empirique dans ce cadre (Mezzoudj et Benyettou. 2018, en presse). L'utilisation de ressources linguistiques et matérielles plus riches que ce que nous possédons actuellement et le fait d'exploiter certaines techniques prometteuses d'optimisation et/ou d'adaptation sont nécessaires pour accélérer et améliorer l'apprentissage des modèles de langage.

Dans le chapitre suivant, nous présentons et explorons d'autres techniques statistiques prometteuses pour l'amélioration des systèmes actuels de la reconnaissance automatique de la parole.

Chapitre 5

Sélection des données textuelles pour les modèles de langage

1 Introduction

L'approche probabiliste est suffisamment universelle puisque les techniques de l'apprentissage automatique statistique permettent de modéliser beaucoup de phénomènes. Les méthodes statistiques pour le traitement et la reconnaissance automatique de la parole [Rabiner and Juang, 2004] représentent une méthodologie générale dans laquelle les connaissances à la fois du signal audio et du langage sont exprimées par un système à base de formalismes mathématiques et statistiques bien fondés. Ces concepts sont traités dans les chapitres précédents.

Dans le contexte actuel de la reconnaissance de la parole spontanée et/ou conversationnelle à large vocabulaire, l'environnement sonore n'est pas toujours maîtrisé. Des phénomènes extra-linguistiques interfèrent dans la communication tels que la musique, des bruits extérieurs, etc. L'élocution de l'utilisateur est souvent marquée par différentes caractéristiques du discours spontané, ce qui génèrent une grande variabilité des phrases prononcées. La même phrase (ou proche) peut être prononcée correctement, ou bien avec quelques hésitations sur certains mots, ou sur d'autres mots, ou encore elle peut être prononcée avec des erreurs que l'utilisateur s'empresse de corriger, avec des raclements de gorge, des variations émotionnelles de l'utilisateur (énervement, rire, etc.), des hésitations, ou encore des reprises de quelques mots, etc. A cette variabilité, s'ajoute la faible quantité de corpus étiquété généralement disponible pour développer une nouvelle application de transcription. Ceci rend l'estimation des modèles acoustiques et des modèles de langage robustes une tâche difficile.

Le potentiel des modèles statistiques standard commence à atteindre ses limites et des nouvelles techniques toujours statistiques sont introduites pour pousser encore l'amélioration des systèmes automatiques de reconnaissance à large vocabulaire (*Large Vocabulary Continuous Speech Recognition* - LVCSR). Parmi ces techniques introduites, nous pouvons citer : l'utilisation des mesure de confiance ou/et des analyses des erreurs des systèmes de la RAP, l'utilisation au lieu des modèles de Markov (GMM) d'autres modèles pour la modélisation acoustique tel que : les champs aléatoires conditionnels(CRF) [Lafferty et al., 2001] ou les réseaux de neurones profonds (*Deep learning*). Dans le cadre de la modélisation du langage, il est possible d'utiliser au lieu des n-grammes standards d'autres modèles tel que : les réseaux de neurones, ce qui correspond au sujet du chapitre précédent. Aussi, la sélection des données audio ou textuelles pour l'amélioration des modèles acoustiques ou linguistiques est considérée, etc. Dans les sections suivantes, nous allons présenter les

détails de ces techniques statistiques dédiées pour l'amélioration des systèmes de LVCSR.

2 Mesures de confiance et analyse des erreurs

En modélisant le système de la RAP comme une boîte noire probabiliste avec des paramètres qui sont appris d'une façon complètement automatiquement, il est important de vérifier si les paramètres appris sont vraiment représentatifs du signal acoustique modélisé. Dans ce contexte, les *mesures de confiance* servent à estimer la probabilité d'une hypothèse de reconnaissance déduite depuis le système de la RAP. Elles sont utilisées dans divers champs d'applications tel que l'extraction des annotations pertinentes de transcriptions automatiques ou l'amélioration de l'efficacité des systèmes de dialogue grâce à la détection des erreurs et des mots hors-vocabulaire. Aussi, l'*analyse des erreurs* générées par le système de la transcription de la parole peut être exploité pour améliorer ses performances.

L'objectif d'une mesure de confiance d'un mot est d'estimer la probabilité qu'il a été correctement reconnu ou non par le système de RAP [Rahim et al., 1997] [Jiang, 2005]. Cette mesure doit être comprise dans l'intervalle $[0, 1]$: idéalement, une valeur de 0 est affectée pour une hypothèse incorrecte, et une valeur de 1 pour une hypothèse correcte.

Il existe plusieurs méthodes et algorithmes pour le calcul des mesures de confiance. Sachant que le système de RAP détient divers informations sur la pertinence d'une hypothèse (un mot) qui peuvent être extraites des paramètres acoustiques, des modèles acoustiques (MA), des modèles de langage (ML) ou du processus de décodage. Les modèles acoustiques fournissent un score de vraisemblance pour un mot contenu dans le dictionnaire de phonétisation et retenu lors du processus de décodage. Le modèle de langage donne une indication sur la pertinence de la séquence de mots (c à.d ses phrases) que le système de RAP vient de décoder. Ces informations peuvent être utilisées directement comme des mesures de confiance, après modification ou combinaison entre eux. Aussi, des connaissances sémantiques ou syntaxiques peuvent être utilisées comme mesures de confiance.

L'analyse des erreurs générées lors de la transcription, par le système de RAP contribue à détecter les points de faiblesse de ce système afin de l'améliorer. Plusieurs idées sont exploitées dans la littérature. Dans [Acero, 1990], la performance du système de RAP dégrade lorsque le microphone était changé du CLSTK (*Sennheiser HMD224 close-talking microphone*) au CRPZM (*Crown PZM 6 fs microphone*). Dans ce cas, le CLSTK présentait un SNR (*Signal-to-Noise Ratio*) de $38,4$ dB alors que le CRPZM avait un SNR de $19,7$ dB. L'auteur a analysé les erreurs qui se sont produites quand le mirophone CRPZM était utilisé pour l'enregistrement des données phonétiques au lieu du microphone CLSTK. Il a étudié les spectrogrammes et il a écouté attentivement tout les énoncés qui contienent des erreurs de test qui ont apparu avec le CRPZM et qui ne figurait pas avec l'utilisation de CLSTK. Les causes estimées des nouvelles erreurs en utilisant le CRPZM sont recensées. La principal conséquence de l'utilisation de CRPZM est que le SNR effectif a été abaissé. Par conséquent, il y avait beaucoup de confusions entre les segments de silence ou de bruit avec des événements phonétiques faibles. Ces confusions représentaient environ 55% des erreurs supplémentaires.

Durant la campagne ESTER [Galliano et al., 2005], il a été remarqué lors de l'analyse des résultats de la transcription que de nombreuses erreurs portent sur les mots outils, qui sont des mots fréquents dans le langage et souvent confondues lors de la transcription automatique comme : et, est (verbe être), a (verbe avoir), à (préposition), un, que, qui,

il, y, etc. [Nemoto et al., 2008].

Des méthodes pour générer des listes d'erreurs ordonnées selon leur impact sur la tâche de transcription, produites par un système de RAP, sont proposées dans [Galibert et al., 2016]. De telles listes aident à corriger le système de RAP en cas de possibilité ou de l'améliorer pour le rendre plus robuste face aux erreurs identifiées.

3 Modèles neuronaux profonds de parole

Il est évident que la modélisation acoustique et/ou linguistique a un impact important sur la qualité du système de RAP. Nous avons discuté dans le chapitre précédent, l'utilité et l'avantage de l'apprentissage profond pour la modélisation du langage. Il y' a eu un intérêt remarquable et progressif pour l'utilisation de l'apprentissage profond même pour la modélisation acoustique.

La reconnaissance automatique de la parole est depuis longtemps, plus de trente ans, dominée par les modèles acoustiques à base de GMM-HMM. Le réseau de neurones et particulièrement le réseau neuronal traditionnel MLP est une approche populaire. Ce dernier a été utilisé auparavant pour la reconnaissance vocale pendant de nombreuses années mais il n'était pas concurrentiel et son rendement était généralement inférieur à celui des modèles HMM-GMM.

La plupart des techniques d'apprentissage et de traitement des signaux avaient exploité des architectures peu profondes ou plats. Ces architectures contiennent typiquement au plus une ou deux couches de transformations non linéaires des caractéristiques. Parmi les exemples d'architectures peu profondes on peut citer : les mélanges des gaussiennes (GMM) [1]. , les systèmes dynamiques linéaires ou non linéaires, les champs conditionnels aléatoires [2] (*Conditional Random Fields* - CRFs) [Lafferty et al., 2001], les modèles d'entropie maximale (MaxEnt), les machines à vecteurs de support (SVM) [Boser et al., 1992], la régression logistique, la régression à base de noyau, le perceptron multi-couches (MLP) avec une seule couche cachée et les machines d'apprentissage à couches extrêmes (ELM) [3] [Deng and Yu, 2014].

Cependant, les mécanismes de traitement humain de l'information (par exemple, vision et audition) suggèrent la nécessité d'architectures profondes pour extraire la structure complexe et construire une représentation interne des entrées sensorielles riches. Par exemple, les systèmes de la production et de la perception de la parole humaine sont équipés de structures hiérarchiques clairement profondes qui permettent de transformer l'information du niveau acoustique au niveau linguistique. Dans le même ordre d'idées, le système visuel humain est également de nature hiérarchique, surtout dans le domaine de la perception.

Les modèles acoustiques standard à base de mélanges de gaussiennes (GMM) permettent de relier les différents états d'HMM aux entrées acoustiques. D'habitude, l'entrée

1. Les modèles GMM-HMM sont considérée comme des modèles peu profond ou même plat

2. Les champs conditionnels aléatoires sont des modèles graphiques non dirigés ayant pour objectif de définir une distribution de probabilités d'annotations (ou étiquettes ou labels) y conditionnellement aux observations x.

3. le terme *extreme learning machine* fait référence à un type de réseau de neurones qui a une seule couche cachée de nœuds, où les poids des entrées de connexion des nœuds cachés sont répartis au hasard et jamais mis à jour. Ces poids entre les nœuds cachés, d'entrée et les sorties sont appris en une seule étape, ce qui revient essentiellement à l'apprentissage d'un modèle linéaire. Ces modèles peuvent produire une bonne performance de généralisation et avoir un processus d'apprentissage beaucoup plus rapide que les réseaux formés en utilisant la rétropropagation du gradient

acoustique est typiquement représentée par la concaténation de Coefficient Cepstral de Fréquence Mel (MFCC) ou de Coefficients Prédictifs Linéaires Perceptifs (PLP) calculés à partir de la forme des ondes audio, et leurs dérivées de premier et second ordre. Cette extraction des caractéristiques prend du temps et le résultat obtenu est souvent incomplet. Selon [Hinton et al., 2012], ce prétraitement de la forme d'onde (par MFCC ou autres) n'est pas vraiment adapté pour la reconnaissance automatique malgré qu'il est basé sur de hautes techniques. Il est conçu sur le principe d'éliminer une grande quantité d'informations considérées comme non pertinentes des ondes d'entrées et ensuite d'exprimer l'information restante sous une forme qui facilite la discrimination avec les modèles GMM-HMM. Si l'apprentissage automatique était capable de prendre en charge la phase d'extraction des caractéristiques (ou encore mieux : d'apprendre automatiquement ces caractéristiques), les autres tâches d'analyse ou de classification pourraient être résolus d'une façon plus fiable et robuste.

En effet, l'apprentissage en profondeur fournit un moyen d'automatiser la fonction d'extraction des caractéristiques.

A ce stade, il est nécessaire, afin d'éviter la confusion et faciliter la compréhension, d'introduire une terminologie standard de l'apprentissage profond [Deng, 2014] :

1. **Apprentissage profond** (*Deep Learning*) est une classe de techniques d'apprentissage automatique, où plusieurs couches de traitement de l'information sont exploitées pour un apprentissage non supervisé des caractéristiques et pour l'analyse / la classification des formes. L'essence de l'apprentissage profond est de calculer et d'extraire les caractéristiques représentants les données. La famille des méthodes d'apprentissage profond englobent les réseaux de neurones, les modèles probabilistes hiérarchiques et une variété d'algorithmes supervisés et non-supervisés pour l'apprentissage des caractéristiques des données. Les réseaux d'apprentissage profond les plus populaires sont :
2. Le réseau de croyance profonde (*Deep Belief Network* - DBN) : des modèles probabilistes génératifs composés de plusieurs couches de variables stochastiques et cachées. En haut deux couches ont des connexions non dirigées et symétriques entre elles, tandis que les couches inférieures reçoivent des connexions dirigées de haut en bas depuis la couche de dessus.
3. La machine de Boltzmann (*Boltzmann Machine* - BM) : un réseau de connexions symétriques, des unités neuronales qui prennent des décisions stochastiques pour s'allumer ou s'éteindre.
4. Le réseau neuronal profond (*Deep Neural Network* - DNN) [4] : un perceptron multicouches avec de nombreux couches cachées, dont les poids sont entièrement reliés et sont souvent (bien que pas toujours) initialisés en utilisant soit une technique de prétraitement supervisée ou non supervisée.
5. L'auto-codeur profond (Deep Autoencoder) : un DNN discriminatif dont les cibles de sortie sont elles même les données d'entrée, plutôt que les étiquettes de classe, d'où il est considéré comme un modèle d'apprentissage non supervisé. Aussi, si l'auto-codeur profond est formé avec un critère de dé-bruitage, il est considéré comme un modèle génératif.
6. Représentation répartie (*Distributed representation*) : une représentation interne des données observées de telle sorte qu'elles soient modélisées par les interactions

4. Dans la littérature antérieure à 2012, un DBN a souvent été utilisé incorrectement pour désigner un DNN.

de nombreux facteurs cachés. Un facteur particulier appris depuis la configuration d'autres facteurs peut souvent être utilisé pour généraliser correctement de nouvelles configurations. Les représentations distribuées s'apprend par un réseau neuronal connexionniste, où un concept est représenté par une activité à travers un certain nombre d'unités et où en même temps une unité contribue généralement à de nombreux concepts. Cette représentation fournit la robustesse dans la représentation de la structure interne des données, ce qui facilite la généralisation des concepts et des relations entre ces concepts, permettant ainsi des capacités de raisonnement intéressentes.

Les GMMs sont des modèles génératifs appris par maximisation de la vraisemblance sur les données d'apprentissage alors que les Réseaux neuronaux profonds (DNNs) sont des modèles discriminants, dont les paramètres sont estimés pour minimiser les erreurs de classification. Cependant avant les années 2000, ni le matériel ni les algorithmes d'apprentissage n'étaient suffisamment capable de former des réseaux neuronaux avec de nombreuses couches cachées sur de grandes quantités de données. Aussi, les avantages de l'utilisation du perceptron multi-couches MLP avec une seule couche cachée n'étaient pas suffisamment importants pour compromettre sérieusement les modèles de mélanges de gaussiennes (GMM).

Au cours des dernières années, les progrès des algorithmes d'apprentissage et du matériel informatique ont conduit à des méthodes plus efficaces pour la formation des réseaux de neurones profonds qui contiennent de nombreuses couches d'unités cachées non linéaires et une très grande couche de sortie. La principale contribution pratique des réseaux de neurones à cette époque est de fournir des fonctionnalités supplémentaires dans les systèmes pour extraire en amont les caractéristiques des données avant de passer à toutes autres actions de classification, prédiction ou autre. Les modèles d'apprentissage profonds apprennent les caractéristiques à partir des entrées brutes, d'où ils nécessitent plus de données que les modèles traditionnels. De nombreuses institutions (Google, Facebook, etc.) peuvent recueillir d'énormes ensembles de données qui peuvent être utilisés pour former des modèles profonds avec de nombreux paramètres. Aussi, les modèles neuronaux ont besoin d'un grand nombre de multiplications matricielles qui peuvent être parallélisées sur les architectures actuelles puissantes de calcul multi-core des processeurs et des cartes graphiques (CPU et GPU).

Selon Hinton [5] [Hinton et al., 2012], la parole est produite en modulant un nombre relativement petit de paramètres d'un système dynamique ce qui implique que sa vraie structure sous-adjacente a une taille de dimension beaucoup plus petite que ce qui est immédiatement apparent dans une fenêtre qui contient des centaines de coefficients. D'où vient la proposition que d'autres types de modèles peuvent fonctionner mieux que les GMM pour la modélisation acoustique s'ils peuvent exploiter plus efficacement des informations incorporées dans une grande fenêtre de trames, explique Hinton . Les réseaux de neurones artificiels formés par des dérivées d'erreurs de rétropropagation ont le potentiel d'apprendre de bien meilleurs modèles de données qui se trouvent sur ou à proximité d'un collecteur non linéaire. En fait, il y'a deux décennies, les chercheurs ont obtenu un certain succès en utilisant des réseaux de neurones artificiels avec une seule couche d'uni-

5. Geoffrey Hinton (né le 6 décembre 1947) est un chercheur canadien spécialiste de l'intelligence artificielle et plus particulièrement des réseaux de neurones artificiels. Il fait partie de l'équipe *Google Brain* et est professeur au département d'informatique de l'Université de Toronto. Il a été l'un des premiers à mettre en application l'algorithme de rétropropagation du gradient pour l'entraînement d'un réseau de neurones multi-couches. Il fait partie des figures de proue de la communauté de l'apprentissage profond, avec Yann LeCun (né le 8 juillet 1960 à Paris) et Yoshua Bengio (né en France en 1964), etc.

tés cachées non linéaires pour prédire les états HMM à partir de fenêtres de coefficients acoustiques. Plus précisément, l'apprentissage profond a commencé à faire son impact sur la modélisation acoustique pour la reconnaissance de la parole en 2010, après des collaborations étroites entre des chercheurs industriels (Google, Youtube, Microsoft, IBM, etc.) [Hinton et al., 2012] [Deng et al., 2013]. Au début, la technique d'apprentissage profond est appliquée avec succès à la reconnaissance des phonèmes dans [Dahl et al., 2010], [Mohamed et al., 2012], puis aux tâches de reconnaissance de parole à grands vocabulaires pour les émissions radiophoniques en Anglais dans [Sainath et al., 2013] et même à des tâches de reconnaissance de la parole conversationnelle téléphonique [Seide et al., 2011]).

Aussi, Saon et al. [Saon et al., 2013] ont exploré un nouveau système pour l'adaptation des DNN à la reconnaissance vocale. Le procédé combine les caractéristiques des vecteurs d'identité (*I-vectors* ou *speaker Identity vectors*) avec les fMLLR [6] comme entrée dans un DNN. Les vecteurs d'identité sont couramment utilisés pour la vérification des locuteurs pour des applications de reconnaissance des locuteurs, car elles offrent des informations sur l'identité d'un locuteur via un vecteur des caractéristiques de faible dimension. dans [Cardinal et al., 2014].

De nombreux algorithmes d'adaptation ont été développés pour les modèles GMM-HMM [Gales, 1998], [Gauvain and Lee, 1994] mais ne peuvent pas être facilement appliqués aux DNNs en raison des différentes natures de ces systèmes. Plusieurs méthodes d'adaptation sont proposées pour les DNNs, et quelques unes tirent avantage de l'adaptabilité des GMMs. Cependant il n'existe pas de méthode universelle pour transférer de manière efficace les algorithmes d'adaptation du paradigme gaussien vers celui des DNNs. Une possibilité est d'utiliser les paramètres acoustiques dérivés de GMM pour estimer les modèles DNN [Tomashenko et al., 2016].

4 Sélection des données de parole

Pour la transcription automatique de la parole à grand vocabulaire, de grands corpora audio (incluant généralement des centaines d'heures de parole) servent à estimer des modèles acoustiques précis de phonèmes contextuels. Ces modèles de sons élémentaires sont ensuite concaténés pour aboutir à des modèles de mots en s'appuyant sur la connaissance de leur prononciation avec des modèles GMM-HMM ou ils sont appris par un réseau neuronal DNN.

La sélection des données audio pour améliorer la modélisation acoustique est un domaine de recherche actif, où un certain nombre de méthodes intéressantes sont proposées et exploitées dans la littérature. Pour mesurer la performance des méthodes de sélection spécifiques, la plupart des auteurs ont effectué des expériences avec de grandes quantités de données audio de haute qualité, par exemple [Wu et al., 2007] [Wei et al., 2014a]. D'autres chercheurs ont étudié des techniques de sélection sur des langues avec de faibles ressources [Xiao et al., 2015] [Syed, 2015] et avec des paramètres différents du point de vue la qualité de la parole et des canaux de transmissions utilisés, etc. [Siohan and Bacchiani, 2013].

En général, trois principales approches pour la sélection des données audio sont utilisées : une sélection non supervisée, une sélection supervisée et une combinaison (et/ou

6. L'apprentissage fMLLR (*feature Maximum Likelihood Linear Regression*) [Gales, 1998] s'applique sur les paramètres directement issus de l'observation. Ainsi les paramètres acoustiques sont rapprochés de ceux de l'apprentissage, en modifiant leur espace de représentation et en débruitant les caractéristiques particulières du signal (locuteur, bruit, etc.).

une optimisation) de ces deux méthodes. La sélection des données audio non-supervisée nécessite une décision de sélection sans consultation de la parole transcrite [Wessel and Ney, 2005] [Wei et al., 2014b]. Ce type de méthodes traitent des caractéristiques acoustiques telles que les PLP/MFCC en considérant la variation du locuteur ou du canal de transmission [Dehak et al., 2011], lors du processus de sélection.

Cependant, la sélection supervisée nécessite des données audio transcrites (par exemple quelques heures de parole). Ces données seront utilisées pour l'apprentissage du système de la RAP afin de décoder des données sans transcription. La sélection se fera grâce à des informations dérivées du système de reconnaissance, telles que les mesures de confiance ou les graphes de confusion, etc. Une combinaison de ces informations avec des méthodes supervisées permet d'améliorer la sélection des données.

Dans [Lin and Bilmes, 2009] deux scénarios de la sélection des données sont utilisés sur les données phonétiques TIMIT. Le premier scénario consiste à ce qu'il n'y a pas de modèle acoustique initial disponible. Avec un échantillonnage aléatoire, une petite quantité de données non étiquetées est sélectionnée, puis utilisée pour construire un modèle acoustique initial. Par la suite, ce modèle apprit permet de prédire les données non-étiquetées. Une sélection des m échantillons les plus incertains pour l'étiquetage est réalisée. Enfin le modèle est recyclé en utilisant toutes les données étiquetées. Le processus s'arrête, si le nombre des échantillons étiquetés atteint la quantité ciblée. Le deuxième scénario se base sur le fait qu'un modèle initial est disponible pour faciliter la sélection des données. Un tel modèle acoustique doit être de bonne qualité entraîné (avec 16 composantes GMM-HMM) avec suffisamment de données TIMIT étiquetées, contrairement au premier sénario. Ce modèle initial est ensuite utilisé pour l'échantillonnage, ainsi que pour la sélection sous-modulaire des données disponibles. Selon les résultats obtenus des sélections des données audio, le deuxième sénario est le plus efficace pour la reconnaissance.

5 Sélection des données textuelles

Dans le contexte de l'apprentissage automatique, il est communément admis que plus les corpora textuels utilisés à entraîner les modèles de langage sont larges plus ces derniers sont représentatives, comme atteste Mercer (1985) : *There in no data like more data.* Cette affirmation, bien que correcte, est-elle toujours vraie ? La sélection des données textuelles pour la modélisation de langage est devenue un domaine de recherche actif. Dans la littérature, diverse idées sont proposées dans ce cadre.

Dans [Klakow, 2000], un critère de vraisemblance est utilisé pour sélectionner des articles (de journaux) : les données du corpus d'apprentissage sont évaluées sur un modèle de langage estimé sur un corpus de développement proche de la tâche. La première méthode *one-pass* consiste à considérer un petit corpus de base et de l'enrichir en ajoutant de plus en plus d'articles sélectionnés parmi les données disponibles. La deuxième méthode itérative consiste plutôt à enlever les articles non pertinents du corpus global. Les meilleurs résultats sont obtenus avec un modèle de langage tri-gramme par la stratégie itérative.

Dans [Shen and Xu, 2001], l'utilisation du critère de la perplexité pour une sélection *one-pass* des paragraphes d'un corpus général donne de meilleurs résultats qu'une sélection aléatoire. L'évaluation des modèles de langage est présentée en terme de perplexité et en taux d'erreur WER pour le système de RAP correspondant. Dans [Gao et al., 2000] et[Gao et al., 2002], les auteurs proposent une méthode de sélection basée sur le critère de perplexité pour la segmentation des mots du corpus d'apprentissage et du vocabulaire, appliquée à la reconnaissance du Chinois.

Par ailleurs dans [Moore and Lewis, 2010], les auteurs sont intéressés par la sélection des phrases monolingues du corpus textuel d'apprentissage dans le cadre de la traduction automatique. Ils comparent quatre stratégies de sélection des phrases pour la modélisation du langage : (1) une sélection aléatoire (2) la méthode de Klakow [Klakow, 2000], (3) une sélection à base de l'entropie croisée évaluée sur le domaine spécifique, inspirée de [Gao et al., 2002] et enfin (4) une sélection basée sur la différence de l'entropie croisée des phrases du domaine non-spécifique évaluée entre le modèle de langage appris sur le domaine spécifique de la tâche (*in-domain*) et un modèle de langage réduit généré par une quantité de données de taille similaire aux données spécifiques, tirée du corpus non-spécifique (*non-domain*). Selon les résultats présentés, cette dernière stratégie proposée conduit à une réduction plus importante en terme de perplexité pour le modèle de langage obtenu.

Dans [Axelrod et al., 2011], la sélection des données textuel par le critère de perplexité n'a introduit aucune amélioration pour le système de traduction. Des résultats plus intéressants sont obtenu par le critère de l'entropie croisée et le critère de la différence de l'entropie croisée [Moore and Lewis, 2010] modifiée pour le contexte bilingue. Aussi les auteurs trouvent que : (1) les modèles de langage considérés individuellement sont plus intéressants que le modèle final interpolé. Selon les auteurs, la combinaison par interpolation n'est pas appréciée chez la communauté de la traduction automatique. Ce constat coinside avec ce que annonce Foster [Foster et al., 2010], (2) l'utilisation des modèles de langage, séparément dans le systéme de traduction automatique par la technique *multiple paths decoding* introduite par Birch(2007) dépasse les performances d'un seul modèle de langage développé par les données d'apprentissage concaténées et/ou un modèle de langage linéairement interpolé. Enfin, les auteurs utilisent les meilleures techniques pour développer un système de traduction automatique pertinent.

Dans [Schwenk et al., 2012], l'utilisation de la méthode de [Moore and Lewis, 2010] permet, avec 20% des données du corpus Gigaword GW Anglais, de diminuer la perplexité du modèle de langage 4-gramme : de $87,0$ à $86,6$. Les auteurs utilisent toutes les données disponibles pour construire le modèle de langage *Huge back-off LM*. Ils constatent que l'interpolation des modèles de langage diminue leurs pertinences par rapport aux modèles de langage individuels (ou construit avec concaténation des données). Aussi, l'utilisation d'un modèle de langage neuronal à espace continu (CSLM) entraîné à base de 7-gramme donne de bons résultats.

Dans [Koehn and Haddow, 2012], les auteurs reprennent les deux types d'échantillonnage des données utilisées dans [Schwenk et al., 2012]. Le modèle de langage de référence, développé sur deux sources de données 50 M mots du corpus *Europarl* et 3 M mots de corpus *NewsCommentary*, est amélioré après l'ajout de 550 M mots du corpus Gigaword (*GigaFrEn*) mais faiblement avec 300 M mots du corpus *UN*. Aussi, les auteurs arrivent à de meilleurs résultats en utilisant le score de moore-modifié [Axelrod et al., 2011] qu' avec le score proposé par IBM *length-normalised sum*. Ils constatent que l'interpolation des modèles de langage développés sur des corpora segmentés par source et année est nocive. Des données de même source sur une période plus longue du corpus (*newstest* de 2008 à 2010) permet de développer des modèles de langage plus intéressants.

Dans Toral [2013], l'auteur a utilisé les informations linguistiques : lemmes et entités nommées d'une façon séparée et/ou combinée, pour renforcer la simple sélection par le critère de perplexité. Les résultats obtenus sont encourageants.

6 Expérimentations sur la sélection des données textuelles

Dans le but d'améliorer les modèles de langage pour le système de reconnaissance dédié à la transcription automatique des émissions radiodiffusés et télévisées en Français, nous avons eu la chance durant un stage de perfectionnement à l'étranger financé, par mon Université Hassiba Benbouali de Chlef, d'exploiter les données disponibles dans le laboratoire LORIA de Nancy. Une partie importante de ces données est distribuée dans le cadre des campagnes d'évaluation ESTER, ETAPE ainsi que le projet EPAC. Le reste des données sont collectées au niveau de l'équipe Multispeech de LORIA.

Rappelons que les premières campagnes ESTER1 de 2003 et de 2005 [Galliano et al., 2005] ont ciblé le traitement des émissions de radio, l'édition 2009 ESTER2 [Galliano et al., 2009] a présenté des discours accentués et des émissions de nouvelles avec des discours spontanés. L'évaluation ETAPE 2011 [Gravier et al., 2012] s'est concentrée sur des contenus télévisés avec différents niveaux de parole spontanée et de discours de plusieurs locuteurs. Le projet EPAC de l'Agence Nationale de Recherche (ANR) a contribué à la construction manuelle et automatique du corpus d'un discours conversationnel [Esteve et al., 2010].

Les données textuelles qui correspondent le mieux à la tâche de transcription ciblée sont les données textuelles de la transcription manuelle des émissions radiophoniques et télévisées. Ces données sont bien adaptées pour le développement des modèles de langage d'un tel système de transcription. Cependant ce type de données textuelles coûte cher à produire d'où leur quantité disponible est limitée. Si on se contente d'utiliser seulement ces données pour l'apprentissage lors de la modélisation du langage, cela induira à des modèles de langage de faible qualité.

Selon Bazillon, la transcription manuelle de la parole préparée nécessite environ 2,67 fois plus de temps qu'une transcription assistée. Il est surtout important de retenir que cette assignation demande presque deux fois plus de temps quand la parole est spontanée. Cela s'explique du fait que la parole spontanée, avec ses nombreux tours de parole, contraint le transcripteur humain à leur assigner un locuteur, quand bien même il peut n'y en avoir que deux différents dans un fichier. Cependant, un segment de parole préparée contient souvent de nombreux locuteurs (journalistes, reporters, interviewés, locuteurs (*speakers*), etc.), mais beaucoup moins de tours de parole, dans la mesure où ceux-ci sont beaucoup plus longs. De plus, dans un segment de parole spontanée se trouve parfois de la parole superposée. Quand trois locuteurs ou plus sont susceptibles de prendre la parole, il est parfois long et difficile de déterminer qui parle réellement dans un segment audio[Bazillon, 2011].

En général, la modélisation du langage pour une tâche de transcription ciblée correspond aux données d'apprentissage reflétant le domaine d'intérêt. Cependant, en cas de pénurie de ces données spécifiques (comme dans notre cas), cet apprentissage peut utiliser de données provenant d'autres sources ou domaines relativement proches à la tâche de transcription considérée. Dans notre cas, nous préferons construire des modèles de langage riches en se basant lors de l'apprentissage, sur des corpora textuels représentatifs. Ces données profitent de la diversité qu'on peut trouver dans les émissions radiophoniques et télévisées en termes de source, de période temporelle, de vocabulaire, de syntaxe, etc. Nous utilisons des données textuelles transcrites manuellement à partir des émissions radio-télévisées, des données textuelles extraites des pages web, des journaux et le corpus

Gigaword Français [7].

Il est à noter que, ces données d'apprentissage peuvent être utiles pour notre tâche, comme ils peuvent aussi être bruyants en raison de la variabilité des sources. Les données bruitées causent une baisse de performances des modèles de langage. Pour éviter ce phénomène, il est utile de sélectionner des sous-ensembles pertinents des données de chaque source. Dans ce but, nous avons menés différentes expérimentations de sélection de données textuelles afin d'ajuster au mieux les paramètres des modèles de langage pour la transcription de la parole.

6.1 Données textuelles utilisées

Les données textuelles disponibles au laboratoire LORIA, de l'équipe Multispeech, pour le développement des modèles de langage dédiés à la transcription des émissions radiophoniques et télévisées représentent un large corpus textuel français d'environ deux milliards de mots provenant de quatre sources hétérogènes telles que des journaux, des rapports d'agences de presse, des données Web et une petite quantité de transcriptions manuelles de programmes de radiodiffusion apparus dans des périodes différentes :

— une faible quantité de transcriptions manuelles d'émissions radiophoniques et de dépêches d'agences de presse (Ester, Presse-plus et TNS de la période 1998-2005).
— des journaux (Le Monde et l'Humanité de la période 1987-2011) ;
— des données extraites des sites web (des journaux, TV et Radio web de la période 2006-2011) ;
— le corpus Gigaword (AFP, APW, Gigahead et wiki-news de la période 1994-2008) ;

Après un prétraitement effectué sur les données brutes, nous avons obtenu les statistiques illustrées dans le tableau 5.1. Les tailles des fichiers sont exprimées par le nombre de mots sans les balises de début et de fin des phrases $< s >$ et $< /s >$.

Source	#fichier	#phrase	#mot (avec $< s >$)	#mot (sans $< s >$)
Gigaword	637	28 699 758	840 779 979	783 380 463
Journaux	350	23 111 598	571 673 424	525 450 228
Webdata	293	16 590 162	367 237 324	334 057 000
Transcription	74	5 437 203	124 861 133	113 986 727
All	1 300	73 838 721	1 904 551 860	1 756 874 418

TABLE 5.1 – Statistiques du corpus textuel disponible pour l'apprentissage des modèles de langage, de Mutlispeech LORIA 2015.

Les même fichiers de développement, de test et le vocabulaire (de 97 349 mots) sont utilisés lors de toutes les expériences réalisées. Leurs statistiques sont présentes dans le tableau 5.2. Ces fichiers sont relativement proche de notre tâche.

Fichier	#fichier	#phrase	#mot (avec $< s >$)	#mot (sans $< s >$)
Développement	1	20 091	316 952	276 770
Test	1	7 551	100 293	85 191

TABLE 5.2 – Statistiques des fichiers de *développement* et de *test*.

7. ttps ://catalog.ldc.upenn.edu/LDC2011T10

Dans le cas de données hétérogènes, l'approche conventionnelle consiste tout d'abord à former un modèle de langage individuel sur chaque corpus (ou source de données), puis de les combiner de manière à maximiser l'ajustement du modèle de langage interpolé résultant avec des données de développement représentant la tâche ciblée. Cette approche est utilisée pour la création de notre modèle de langage référentiel (*Baseline LM*), qui est formé par l'interpolation linéaire à partir du large corpus textuel, des quatre sources de données, présenté dans le tableau 5.2.

Lors du développement d'un modèle de langage, les n-grammes varient selon l'époque de production du corpus d'apprentissage utilisé, son style et les sujets traités. Nous avons considéré les corpora des différentes sources sur des période relativement longue pour garder une homogénéité dans nos données tout en s'inspirant des expériences de [Koehn and Haddow, 2012]. Nos modèles de langage référentiels (*baseline*) peuvent changer selon les sources de données utilisées (deux sources ou quatre sources, etc.) d'une expérimentation à une autre.

Quelques notations sont utilisées pour désigner les sources de données disponibles : Gw pour le corpus *Gigaword*, Np pour les journaux (*NewsPapers*), Tr pour les transcriptions, Web pour les données du Web (*Web data*) et *All* (tout) pour les quatre sources de données considérées ensemble. Aussi rappelons que, tout nos modèles tri-grammes sont développés par la boîte d'outils *SRILM* [Stolcke, 2002],[Stolcke et al., 2011], lissés par la méthode de Kneser-Ney modifiée [Chen and Goodman, 1999] avec repli. Le modèles de langage finaux sont interpolés linéairement et ajustés par l'algorithme EM (*Expectation-Maximisation*) en cas de différentes sources de données. Les modèles de langage développés sont évalués par la perplexité noté *ppl*, calculée pour chaque mots et sans considérer les symboles de début et de fin des phrases, dans le but d'assurer le plus de précision possible. Ce terme est connu dans le lexique de l'outil SRILM sous la notation *ppl1*.

Un modèle de langage construit avec les données des transcriptions Tr (113 M de mots) atteint une perplexité ppl de 253.11. Cependant ce modèle enrichie avec les données des deux sources de journaux Np et du Web (859 M de mots en total) permet de diminuer la perplexité de 218.93, ce qui correspond à un gain de -34.19 en perplexité. Ce phénomène nous encourage à ajouter encore plus de données pour développer un modèle de langage plus robuste. L'incorporation de 783 M de mots du corpus Gigaword Gw, diminue la perplexité du modèle à seulement $218.87 \simeq 218.9$ (un gain de (-0.064)).

LM	**ppl** $DevLM$	**ppl** $TestLM$
$LM - Tr$	215.6	253.0
$LM - (Tr + Web + Np)$	185.8	218.93
$LM - (Tr + Web + Np + Gw)$	185.7	218.87

TABLE 5.3 – Perplexités des ML entrainés avec les differentes sources de données.

Les détails du modèle interpolé, qui est considéré comme modèle référentiel (*Baseline LM*), sont présentés dans le tableau 5.4. Nous remarquons que l'ajout des quantités de données pour la construction des modèles de langage n'est pas toujours encourageant, mais la qualité des données qu'on ajoute est plus importante. De ce fait, la nécessité d'effectuer des sélections sur nos données textuelles pour en extraire les plus pertinentes pour la modélisation du langage, en respectant la tâche spécifique de transcription d'émissions radiophoniques peut être nécessaire.

Sources	MLs individuels	ML interpolé		
	ppl $DevLM$	**poids**	**ppl** $DevLM$	**ppl** $TestLM$
Tr	215.6	0.685	185.7	218.9
Web	264.7	0.246		
Np	364.2	0.062		
Gw	531.4	0.007		

TABLE 5.4 – ML referentiel, interpolé à partir des MLs individuels.

6.2 Pré-sélection sur le corpus complet

Nous commençons par réaliser des sélections sur le corpus complet *All*, ce qui correspond à toutes les sources de données disponibles : GW, Np, Tr et Web. Les méthodes de sélection sont implémentées (en perl) et un processus de parallélisme est utilisé sur les différents nœuds de la machine -du LORIA- pour accélérer la sélection des données et l'apprentissage des modèles de langage correspondants.

- Nous sélectionnons des portions aléatoires {5%, 10%, ... 100%} du corpus complet avec lesquelles nous générons des modèles de langage 3-grammes et nous évaluons leurs perplexités.
- Nous sélectionnons les phrases ayant une perplexité $ppl1$ (noté pour simplicité ppl) inférieure ou égale à un certain seuil [8] qui varie entre {50, 100,..., 1700}. Pour chaque seuil, les données sélectionnées servent pour la génération d'un nouveau modèle de langage.
- Nous évaluons toutes les phrases du corpus complet sur deux modèles de langage : (1) le modèle généré par les phrases du corpus complet $LM-All$, déjà utilisé ; (2) le modèle généré par toutes les phrases du GW, noté $LM-GW$; Pour chaque phrase s du corpus complet, nous calculons une différence de log-probabilité évaluée entre ces deux modèles, avec la formule : $(logprob(s)_{(LM-All)} - logprob(s)_{(LM-GW)}$. Les phrases ayant une différence supérieure ou égale à un certain seuil de la différence de log-probabilité pris parmi {20, 10,..,-200 } seront sélectionnées et utilisées pour la génération de nouveaux modèles de langage.

En considérant les résultats de ces premières expérimentations de sélection réalisées sur le corpus complet *All* (voir figure 5.1), nous constatons que la sélection aléatoire est plus intéressante qu'une sélection basée sur le critère de la perplexité évaluée respectivement sur le $LM-All$ et $ML-GW$. En général, l'utilisation du critère de la perplexité évaluée sur un domaine spécifique permet de sélectionner parmi les phrases du corpus non-spécifique seulement les phrases qui sont proches à celles du domaine spécifique, c'est à dire celle ayant des perplexités proches. Dans notre cas, cette faiblesse du critère de perplexité peut être justifié par deux propositions :

- le fait que le corpus GW qui représente presque la moitié des données utilisé pour l'apprentissage du modèle de langage est di-similaire (ce qui est bizarre mais peut être vrai) du sous-corpus Tr spécifique, puisqu'il s'est présenté lors de la combinaison linéaire avec des poids (lamdba) de l'ordre de $0,0072$ voir le tableau 5.5, d'où

8. Nous générons des fichiers avec l'option *-debug 1*, de la commande *ngram* de SRILM, pour toutes les phrases du corpus complet évaluées avec le modèle de langage noté $LM-ALL$ et construit à partir de ces phrases (toutes les phrases du corpus complet). Dans ces fichiers (qu'on note avec une extension .ppl) nous avons les détails suivants pour chaque phrase : nombre de mots, MSDP, ppl et ppl1.

apparaît la difficulté de notre situation proche à celle traiter dans [Foster et al., 2010].

— et/ou bien le fait de choisir l'ordre 3 pour les modèles de langage appris sur le corpus *GW*, peut être insuffisant pour y extraire l'information utile, sachant que le contexte des phrases du corpus *GW* est long.

Pourcentage	Données	ppl1-Dev	ppl1-Test	#Mots	Poids
Corpus complet					
100%	*GW*	531.368	218.865	783 380 463	0.0072
	Np	364.187		525 450 228	0.0617
	Tr	215.617		113 986 727	0.6847
	Web	264.682		334 057 000	0.2464
	All	185.702		1 756 874 418	0.99 = 1

TABLE 5.5 – Poids et perplexités des MLs par sources de données et le ML interpolé *Baseline LM*.

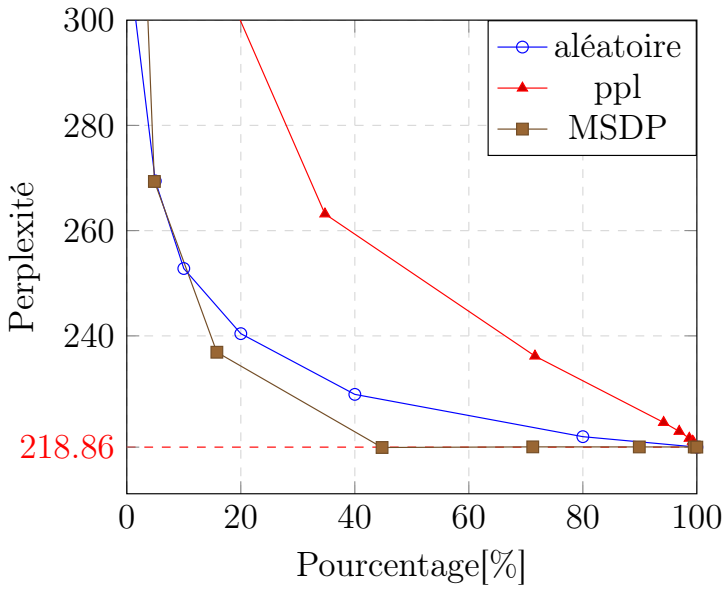

FIGURE 5.1 – Perplexité des MLs générés par trois types de sélection à base de scores différents.

Les trois courbes représentent les types de sélection appliquées : (1) *aléatoire* sur les phrases du corpus complet, (2) *ppl*, basée sur le critère de la perplexité (ppl) évaluée sur le $ML - All$ et (3) $dLogP$, basée sur le critère de différence de log-probabilité entre les modèles de langage $ML - All$ et $ML - GW$ des phrases du corpus complet.

D'autre part les résultats obtenus par la sélection basée sur la différence de log-probabilité ne sont pas optimaux mais encourageants. Pour la suite du travail, nous focalisons nos expérimentations sur deux points :

— considérer seulement les deux corpus critiques : Tr (comme *in-domain*) et Gw (comme *non-domain*).

— utiliser un critère de log-probabilité inspiré du travail de [Moore and Lewis, 2010].

6.3 Stratégie de sélection des données textuelles

Nous utilisons des méthodes de sélection de données sur le corpus textuel en vue d'améliorer la modélisation du langage pour la transcription automatique des émissions de la radio et de la télévision. Les méthodes de sélection utilisées reposent sur le calcul d'un score pour chaque phrase, qui représente une évaluation de la proximité de la phrase dans les données du domaine par rapport aux données non spécifiques au domaine.

Les expérimentations rapportées dans cette section sont conduites pour valider la mise en œuvre de la procédure de sélection basée sur la différence d'entropie croisée, comme décrit dans [Moore and Lewis, 2010]. Pour ce faire, un contexte similaire à celui utilisé dans ce document est simulé : seules deux sources de données sont considérées. La première source : les données du domaine de transcription Tr, puisque les transcriptions manuelles des émissions diffusées sont les plus semblables aux données de développement et de test *ETAPE* selon les valeurs de perplexité rapportées dans le tableau 5.5. Pour la deuxième source qui représente les données du domaine général ou les données non-spécifiques au domaine (*non-domain*) en respectant la terminologie définie dans [Moore and Lewis, 2010] (voir figure 5.2), elle contient les données du Gigaword Gw.

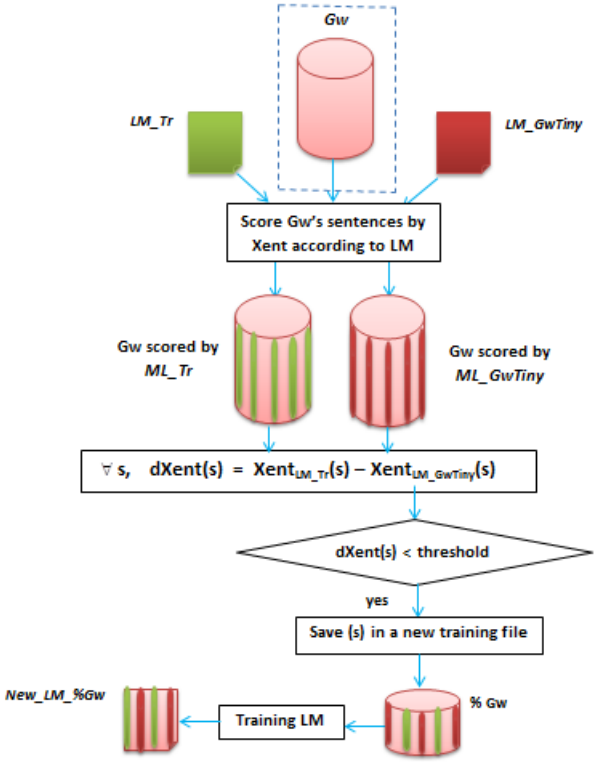

FIGURE 5.2 – Pricipe de la sélection de donnée par le critère $dXent$.

Deux processus de sélection sont donc évalués. Le premier est basé sur une sélection aléatoire des données. Un ensemble de modèle de langage est formé avec des sous-ensembles correspondant à la sélection aléatoire des pourcentages {5%, 10%,..100%} des données Gw. La seconde est la méthode de sélection de données décrite dans [Moore and Lewis, 2010], basée sur la différence entre l'entropie croisée des phrases évaluée sur le modèle de langage correspondant au domaine et le modèle de langage non-spécifique au domaine. Pour chaque phrase s du corpus Gw, la différence d'entropie $dXent(s)$ est calculée en utilisant deux modèles de langage de taille similaire : un modèle e langage $LM-Tr$ est entraîné sur les données de transcription et l'autre modèle de langage $LM-GwTiny$

est entraîné sur un sous-ensemble sélectionné au hasard du corpus Gw d'environ la même taille que le corpus Tr (soit environ 114 M mots). Par l'équation suivante 5.1 :

$$dXent(s) = H_{(LM-Tr)}(s) - H_{(LM-GwTiny)}(s) \tag{5.1}$$

Nous fixons une valeur pour le critère $dXent$, nous construisons un modèle de langage avec les phrases sélectionnées avec les valeurs inférieures au seuil choisi du score. La figure 5.3 représente la perplexité du corpus $TestLM$ par rapport au pourcentage des mots sélectionnés pour l'apprentissage des modèles de langage. Les résultats sont obtenus avec une sélection aléatoire et avec la stratégie basée sur la $dXent$ présentée dans cette section. La perplexité $(671, 4)$ obtenue en utilisant le corpus entier du Gw est illustré dans la figure suivante 5.3 :

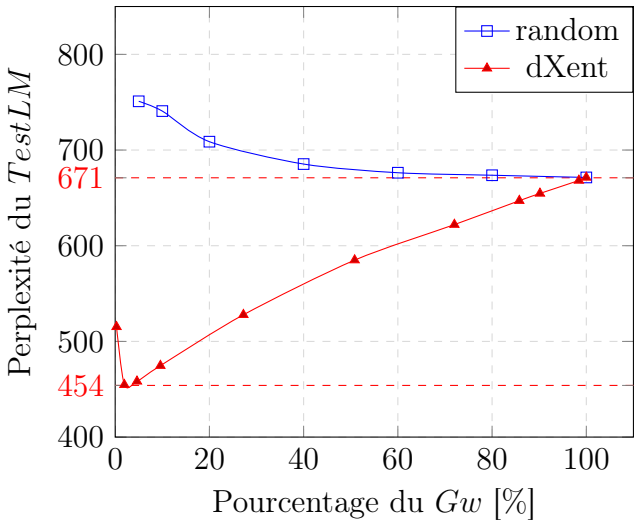

FIGURE 5.3 – Perplexité sur le corpus LM-Test par rapport au pourcentage des données du corpus Gw sélectionnées pour le processus d'apprentissage du ML.

La sélection est appliquée sur les données du corpus Gw : (1) une sélection aléatoire (*random*) appliquée uniquement sur les données du Gw et (2) une sélection basée sur le critère de la différence de l'entropie croisée $dXent$ calculée entre le modèle (*in-domain*) $LM - Tr$ et le modèle (*non-domain*) $LM - GwTiny$.

Évidemment, l'utilisation des sous-ensembles d'apprentissage résultant d'une sélection aléatoire sur les données Gw dégrade la perplexité. En revanche, en utilisant la différence du critère de $dXent$ pour la sélection des données dans le corpus Gw, nous obtenons une amélioration de la perplexité (avec la meilleure valeur égale à $454, 7$). Ceci est dû au fait que ce critère de sélection est capable de sélectionner des phrases proches au corpus du domaine Tr et loin des données non-spécifiques au domaine représentée par le modèle de langage $LM - GwTiny$.

Le comportement de ces résultats est très similaire au comportement rapporté dans [Moore and Lewis, 2010]. Les meilleurs résultats sont obtenus avec un petit ensemble de données sélectionnées du Gw qui correspondent le mieux aux données de la cible. Ceci valide la mise en œuvre de la procédure de sélection.

6.4 Expérimentation sur le corpus complet

Dans cette section, nous étudions la sélection de données textuelles dans le contexte de modèles de langage en considérant une multi-source. En utilisant le critère de sélection basé sur la différence d'entropie croisée $dXent$, on analyse plusieurs choix de modèles

pour représenter des données dans le domaine spécifique (*in-domain*, ici ETAPE) et le domaine non-spécifique (*non-domain*).

Approche 1

Comme dans le cas de la section précédente, un processus de sélection aléatoire est également évalué. Le principe de cette approche de sélection est illustré dans la figure 5.4. Des sous-ensembles de respectivement de 5%, 10%, etc. de données sélectionnées au hasard sont extraits de chaque source de données. La perplexité obtenue pour les modèles interpolés construits à partir de ces sous-ensembles est rapportée dans la figure 5.6. Il est clair que, la perplexité se dégrade à fur et à mesure que la quantité de données sélectionnées diminue.

Cette section est consacrée à l'exploration de différents choix de modèles de langage pour la représentation du domaine et des données non-spécifiques au domaine. Aussi, une autre question qui se pose : est ce que le processus de sélection doit être appliqué sur toutes les sources de données ou uniquement sur les données du Gigaword Gw ?

FIGURE 5.4 – Principe de la sélection des données par l'approche 1.

Comme dans la section précédente, les données du domaine sont présentées par le modèle de langage $LM - Tr$ formé sur les transcriptions manuelles Tr et les données non-spécifiques au domaine sont représentées par le modèle de langage $LM - GwTiny$ formé sur un sous-ensemble aléatoire du Gw de taille similaire au corpus Tr (comme déjà mentionné dans la section précédente) et illustré dans la figure 5.4. Le critère $dXent(s)$ est calculé pour chaque phrase s des corpus Tr, Web, Np et Gw (c'est-à-dire sur chaque source) en utilisant ces deux modèles de langage par l'équation suivante :

$$dXent(s) = H_{(LM-Tr)}(s) - H_{(LM-GwTiny)}(s) \tag{5.2}$$

Pour une valeur fixée de $dXent$, un modèle de langage interpolé est formé à l'aide des sous-ensembles de phrases sélectionnées des 4 sources. La perplexité obtenue sur le corpus $TestLM$ est rapportée dans la figure 5.6. Bien que les résultats soient meilleurs que ceux obtenus avec la sélection aléatoire, il n'y a pas d'amélioration de la perplexité, par

rapport au modèle de référence. La sélection est la même que celle décrite dans la section précédente. Cependant, lors de la création d'un modèle interpolé à partir de plusieurs sources, les pondérations combinées sont optimisées pour correspondre aux données du modèle de développement $DevLM$. Cette étape d'optimisation peut masquer l'avantage de la sélection des données sur certains des sous-ensembles, ou bien les modèles utilisés pour représenter les données du domaine et les données non-spécifiques au domaine peuvent ne pas être assez bons pour l'apprentissage.

Approche 2

Dans cette section, les données du domaine ciblé sont maintenant représentées par le modèle de langage de base, interpolé $LM-(Tr+Web+Np+Gw)$ qui représente mieux les données ETAPE que le modèle $LM-Tr$ formé uniquement sur les transcriptions (voir le tableau 5.2, où la perplexité sur les données de développement $DevLM$ pour le modèle individuel $LM-Tr$ est supérieure à celle du modèle de langage interpolé $(215,6 > 185,7)$. Pour représenter les données non-spécifiques au domaine, nous utilisons le modèle $LM-Gw$ formé sur l'ensemble du corpus du Gw. Le critère $dXent(s)$ est calculé pour chaque phrase s des corpora Tr, Web, Np et Gw (c'est-à-dire, chaque source de données) avec ces deux modèles de langage.

$$dXent(s) = H_{(LM-(TrWebNpGw))}(s) - H_{(LM-Gw)}(s) \tag{5.3}$$

Comme illustré dans la figure 5.5, pour chaque seuil fixe du critère $dXent$, un modèle de langage interpolé est formé en utilisant les sous-ensembles de phrases sélectionnées en respect à ce seuil. La perplexité obtenue sur le corpus $TestLM$ est rapportée dans la figure 5.6.

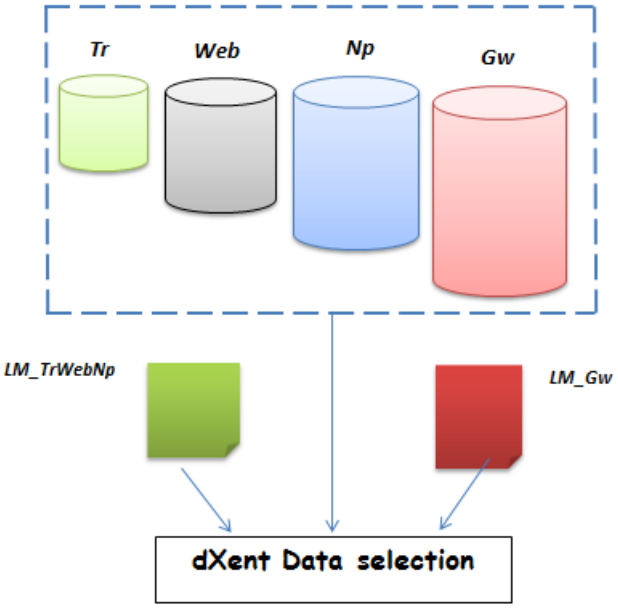

FIGURE 5.5 – Principe de la sélection des données par l'approche 2.

Nous obtenons des résultats encourageants en utilisant seulement 25% du corpus $(Tr+Web+Np+Gw)$ (plus précisément 88% de Tr, 62% de Web, 26% de Np et 0,2% de Gw) pour l'apprentissage de notre modèle de langage. Ce modèle obtenu conduit à une

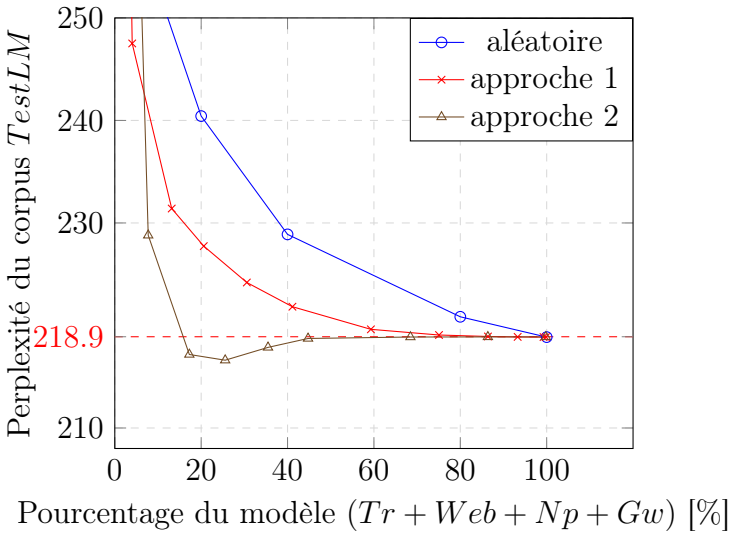

FIGURE 5.6 – Perplexités sur le corpus Test-LM par rapport au pourcentage de mots sélectionnés du $(Tr + Web + Np + Gw)$ pour le processus d'apprentissage. La sélection est appliquée sur chaque source de données basée sur : a) sur un processus aléatoire (rand) ; b) sur le critère $dXent$ calculé avec $LM - Tr$ et $LM - GwTiny$ (approche 1) ; et c) sur le critère $dXent$ calculé avec $LM - (TrWebNpGw)$ et $LM - Gw$ (approche 2).

perplexité de 216.6 sur le corpus $TestLM$. Ses détails sont présentées dans le tableau 5.13.

Nous constatons que le poids accordé aux données du corpus Gw (ou plus précisément au 0.2% de données sélectionnées à partir de Gw) est plus important dans ce modèle interpolé que dans le modèle référentiel de base (0.096 au lieu de 0.007). En comparant les résultats obtenus avec les approches 1 et 2, nous remarquons que le choix des modèles de représentation des données du domaine (*in-domain*) et des données non-spécifiques au domaine (*non-domain*) joue un rôle important sur la sélection des données.

Sources	**Individual LMs**	**Interpolated LM**		
	ppl $DevLM$	**poids**	**ppl** $DevLM$	**ppl** $TestLM$
Tr (88%)	217.6	0.608	185.1	216.6
Web (62%)	262.2	0.234		
Np (26%)	333.0	0.062		
Gw (0.2%)	435.6	0.096		

TABLE 5.6 – Le meilleur ML, interpolé à partir des ML individuels, après la sélection des données en utilisant l'approche 2.

Approche 3

Avec cette approche 3, les données dans le domaine sont de nouveau représentées par un modèle interpolé formé à partir de plusieurs sources de données (Tr, Web et Np) et les données non spécifiques au domaine de la tâche sont représentées par le modèle de langage entraîné sur l'ensemble du corpus Gw.

Cependant, la différence entre les entropies est évaluée pour chaque phrase s du corpus Gw en utilisant ces deux modèles de langage :

$$dXent(s) = H_{(LM-(TrWebNp))}(s) - H_{(LM-Gw)}(s) \tag{5.4}$$

Pour chaque seuil appliqué au critère $dXent$, un modèle de langage interpolé est entrainé en utilisant les données sélectionnées depuis le corpus Gw, et le corpus entier correspondant aux autres sources (Tr, Web et Np). La perplexité obtenue sur le corpus $TestLM$ est rapportée dans la figure 5.8, où une échelle logarithmique est utilisée pour visualiser l'axe horizontal (pourcentage des données Gw sélectionnées). Cette stratégie

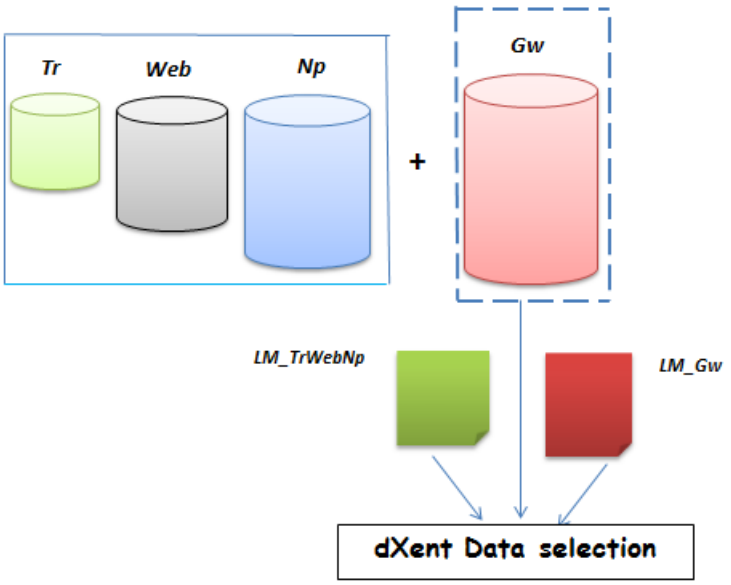

FIGURE 5.7 – Principe de la sélection des données par l'approche 3.

proposée par l'approche 3 donne de bons résultats. Les meilleurs modèles de langage sont obtenus avec une petite quantité de données sélectionnées à partir du corpus Gw (7 K à 70 K mots) ajoutée aux trois autres sources de données. Dans le meilleur des cas, la perplexité diminue à $210,5$. Les détails du modèle correspondant sont présentés dans le tableau 5.7. A ce niveau encore, le poids accordé au modèle entraîné à partir des données sélectionnées du corpus Gw est plus élevé que dans le modèle de référence ($0,054$ au lieu de $0,007$).

Sources	**ML individuel**	**ML interpolé**		
	ppl $MLDev$	**poids**	**ppl** $MLDev$	**ppl** $MLTest$
Tr (100%)	215,6	0,660	179,9	210,6
Web (100%)	264,7	0,240		
Np (100%)	364,2	0,065		
Gw (0,05%)	2822,8	0,054		

TABLE 5.7 – Le meilleur ML, interpolé des MLs individuels après la sélection par l'approche 3.

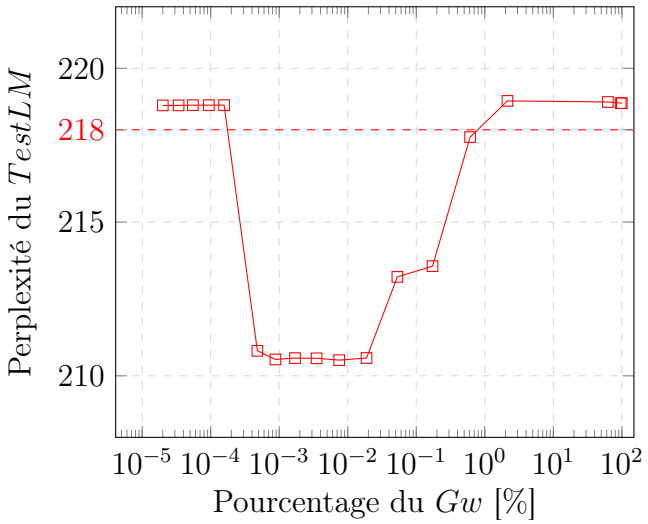

FIGURE 5.8 – Perplexité sur le corpus Test-LM par rapport au pourcentage des données Gw sélectionnées pour le processus d'apprentissage des ML. La sélection est appliquée uniquement sur les données Gw par le critère $dXent$ calculé avec $LM-(TrWebNp)$ et $LM-Gw$ (approche 3).

6.5 Expérimentation de la transcription

Un ensemble sélectionné de modèles de langage résultant des expériences précédentes sont utilisés dans les expériences de transcription de la parole. Comme indiqué dans le tableau 5.8, cela inclut les modèles de base, ainsi que les modèles ayant les perplexités les plus faibles après la sélection des données avec les approches 2 et 3. La performance de la transcription de la parole est évaluée sur les données $ETAPE-Dev$.

Les modèles acoustiques [Jouvet and Fohr, 2014] utilisés pour le système de transcription automatique de la parole sont entraînés avec des données audio de ESTER2, ETAPE et EPAC. Cette quantité de données représente près de 300 heures de signal et près de 4 millions de mots courants (*running word*). Les ensembles de test et de développement des données de la compagne ESTER2 sont utilisés pour améliorer les performances des modèles acoustiques. Les résultats en terme de taux d'erreur sur les mot WER (*Word Error Rate*) sont estimés pour l'ensemble de développement ESTER2 des radio non-africains (environ 72000 mots courants). Les ensembles de test et de développement des données de la compagne ETAPE, dont chaqun d'eux contient presque 80000 mots courants, sont aussi utilisés pour évaluer les performances de la reconnaissance de la parole. Le principe de la modélisation acoustique utilisée dans nos expérimentations de transcription via le SRAP de Loria (Nancy) est illustré dans la figure 5.9.

Les caractéristiques acoustiques MFCC avec leurs dérivées premières et secondes sont extraites avec les outils de la plateforme HTK, le décodage de la parole est basée sur l'outil Cmu-Sphinx[9]. Une étape de segmentation homme/femme est faite (*diarization step)*.Le vocabulaire utilisé est de 95000 mots. Les prononciations du vocabulaire sont prise parmi les variances présentent dans le lexique BELDEX [Pérennou, 1998]. Les prononciations des mots restants sont obtenus en utilisant les deux méthodes *JMM-based* et *CRF-based Grapheme-to-Phoneme converters* [Illina et al., 2011] [Jouvet et al., 2012a]. Les modèles de langage utilisés sont des 3-grammes crées par l'outil SRILM [Stolcke, 2002], [Stolcke et al., 2011].

9. http ://cmusphinx.sourceforge.net/

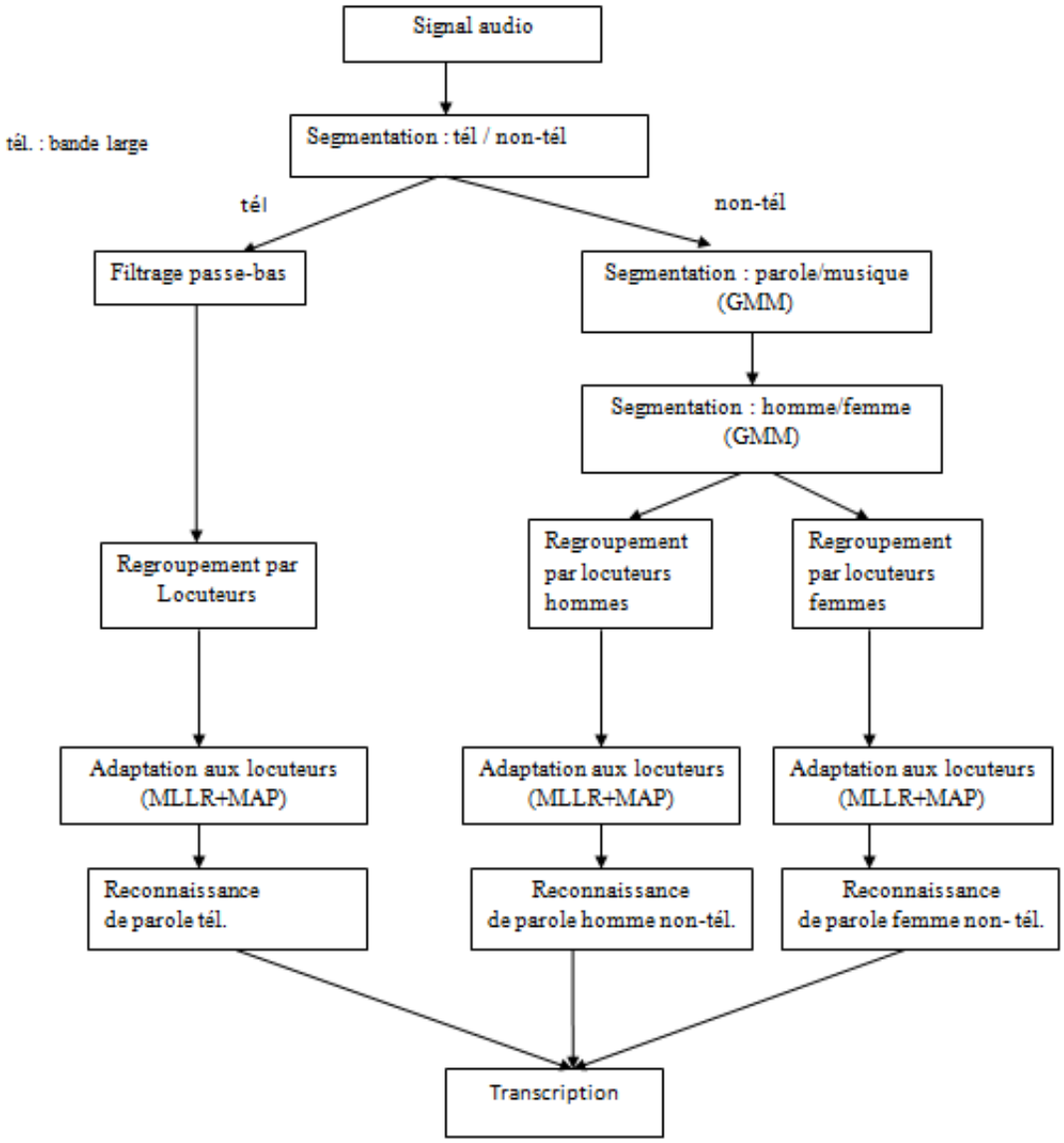

FIGURE 5.9 – Principe de la reconnaissance du système de transcription des émission radio-phonique de Loria.

L'amélioration majeure en termes de perplexité pour nos modèles de langage, après la sélection des données, est d'environ $8,3$. Tandis que l'amélioration correspondante en termes de WER est d'environ $0,2\%$. Le plus important est que le modèle de langage interpolé entraîné avec les données sélectionnées qui est de meilleure qualité, a une taille plus réduite (réduction d'un facteur de $2/3$) que le modèle référentiel (*Baseline LM*).

Les expérimentations menées pendant le séjour au LORIA de Nancy sont achevées à ce niveau. Un article récapitulant ces travaux [Mezzoudj et al., 2015a] été présenté dans la conférence internationale ICNLSP'15 [10].

6.6 Exploration des données textuelles

Dans le même contexte et pour avoir une idée claire sur l'influence des types de données disponibles sur l'apprentissage des modèles de langage, nous avons poursuivé nos

10. http ://www.icnlsp.org/

ML	Taille (gz file)	corpus Etape Dev	
		ppl	WER
$LM-(Tr+Web+Np+Gw)$	1,2 Go	218,9	27,84%
$LM-(Tr+Web+Np)$	809,8 Mo	218,9	27,82%
LM_(approche 2, seuil -0,3)	391,3 Mo	217,2	28,07%
LM_(approche 2, seuil -0,2)	501,6 Mo	216,6	27,89%
LM_(approche 3, seuil -0,8)	809,3 Mo	210,5	27,75%
LM_(approche 3, seuil -0,7)	809,3 Mo	210,6	27,72%
LM_(approche 3, seuil -0,6)	809,3 Mo	**210,6**	**27,68** %
LM_(approche 3, seuil -0,1)	809,9 Mo	217,8	27,73%
LM_(approche 3, seuil 0)	881,1 Mo	218,9	27,85%

TABLE 5.8 – Résultats de la transcription de la parole sur le corpus ETAPE Dev.

expérimentations dans ce sens. Donc, afin d'explorer les données textuelles de LORIA, nous les avons utilisé pour l'entrainement de differents modèles de langage selon le type des sources, la période temporelle, etc. bloc par bloc.

Données textuelles Web

Les données du Web sont constitués de données textuelles extraites des journaux web, divers, radiotv et wikinews, notés respectivement "WEBJOURNAUX.C", "WEBDIVERS.C", "WEBRADIOTV.C" et "WIKINEWS.C". L'extension ".C" indique que ces données appartiennent à une période étalée sur 2006 - 2011.

Nous avons entrainé des modèles de langage en utilisant les données du domaine Tr et différents bloc des données du corpus Web. Les résultats en terme de perplexité sont présentés dans le tableau 5.9. Les fichiers WEBJOURNAUX.C sont constitués des extraits de journaux : France-soir, le Figaro, le Monde, le Point, le Parisien, l'Express, Libération, Nouvel-obser. et Ouest France. Nous avons remarqué que leur utilisation pour l'apprentissage d'un modèle de langage a permis un gain de perplexité de 27.4 (un passage de $253,0$ à $225,6$, ce qui correspond aux lignes 1 et 2 du tableau 5.9).

Les fichiers WEBDIVERS.C qui sont constitués de données prise de quelques sites Web : l'internaut.com, voila.fr, yahoo.com, agoravox.fr et des magazines ou des quotidiens en ligne : slate.fr, 20minutes.fr, etc. ont une importance proche de celle des journaux-web puisqu'ils ont permis un gain de $26,4$ en terme de perplexité.

De même, les fichiers WEBRADIOTV.C extraits des émissions de radio-web de France2, info-France24, ici-tf1.fr, etc. sont moins importants que les fichiers précédemment utilisés (voir les lignes 4, 6 et 7 du tableau 5.9) tandis que les fichiers WIKINEWS.C n'apportent aucune contribution positive pour l'apprentissage des modèles de langage (lignes 7 et 8 du tableau 5.9).

La taille des données du Web est présentée dans le tableau 5.10. Pour une première analyse, nous constatons que les résultats obtenus peuvent être influencés par les quantités de données disponibles par chaque bloc. Les 162 M de mots du WEBJOURNAUX ont amélioré considérablement nos modèles de langage tandis que la faible quantité (de 1,7 M) des WIKINEWS était négligable.

LM	**ppl** $DevLM$	**ppl** $TestLM$
$LM - Tr$	215,6	253,0
$LM - (Tr + Web_{Journaux.C})$	191,8	**225,6**
$LM - (Tr + Web_{Divers.C})$	192,7	226,6
$LM - (Tr + Web_{Radiotv.C})$	197,7	233,9
$LM - (Tr + Web_{WikiNews.C})$	209,8	246,2
$LM - (Tr + Web_{Journ.+Div.})$	188,6	221,7
$LM - (Tr + Web_{Journ.+Div.+Radiotv})$	187,4	220,7
$LM - (Tr + Web_{complet})$	**187,4**	**220,7**

TABLE 5.9 – Perplexités des ML appris sur les données Tr et Web utilisées.

$Fichiersdesdonnes$	$Taille(mots[M])$
WEBJOURNAUX.C	162
WEBDIVERS.C	129
WEBRADIOTV.C	40
WIKINEWS.C	1,7

TABLE 5.10 – Taille par mots (sans <s> et </s> des fichiers Web.

Données textuelles Np

En analysant les résultats précédents obtenu par les données WEBJOURNAUX.C, nous nous attendons à ce que l'utilisation des données Np extraites des journaux : le Monde et l'Humanité auront un impact intéressant sur l'apprentissage des modèles de langage. Ce qui est étonnant est que malgré la quantité importante des données Np (526 M de mots contre seulement 162 M de WEBJOURNAUX) le gain en perplexité est seulement de $1,8$ (un passage de 220,7 à 218,9 en terme de perplexité, voir la ligne 1 et la ligne 7 du tableau 5.11).

Ce phénomène est probablement dû au fait que les données disponibles sur le Web sont ciblés et concentrés sur les sujets traités. Elles ressemblent plus aux données manuellement transcrites d'émissions de radio-télévision que ceux disponibles sur les journaux écrits (ou on trouve beaucoup de récits littéraires (blabla ! !).

Aussi pour répondre à des questions du genre :

— Quels type de données sont les plus pertinents pour l'apprentissage des modèles de langage dédiés à la tâche ciblée : la transcription des émissions radiodiffusées, ceux tirés du journal le Monde ou le journal l'Humanité ?

— Les données de quelle période temporelle sont les plus intéressants pour l'apprentissage des modèles de langage ? Ceux de la période ancienne (1987-1997, désignée par l'extension ".A"), moyenne (1998-2005, désignée par l'extension ".B") ou récente (2006-2011), désignée par l'extension ".C") ?

Quelques expérimentations sont réalisées et leurs résultats sont récapitulés dans le tableau 5.11.

L'apprentissage du modèle de langage en utilisant les données extraites du journal le Monde (385 M de mots) est plus intéressant que celles extraites du journal l'Humanité

(141 M de mots), (voir respectivement la ligne 2 et la ligne 3 du tableau 5.11). Ce résultat est sûrement relatif à la quantité disponible de données.

Concernant les trois périodes considérées, l'ancienne (avec presque 266 M de mots) est plus intéressante que la moyenne (231.5 M de mots) et la récente (28.5 M de mots), selon les résultats du tableau 5.11 (sur les lignes 4, 5 et 6).

Les données du WEBJOURNAUX qui ont contribué à de bons résultats sont récoltées récemment, donc nous s'attendant à ce que (probablement) la période récente (.C) des fichiers Np soit la plus intéressante. Cependant vu la petite quantité de données disponibles pour cette période (28.5 M) le résultat était contraire à ce que nous prévoyant.

LM	**ppl** $DevLM$	**ppl** $TestLM$
$LM-(Tr+Web)$	**187,4**	**220,7**
$LM-(Tr+Web+Np_(LeMonde-complet))$	186,2	**219,3**
$LM-(Tr+Web+Np_(LHumanite-complet))$	186,8	220,0
$LM-(Tr+Web+Np.A)$	186,6	**219,7**
$LM-(Tr+Web+Np.B)$	186,5	219,8
$LM-(Tr+Web+Np.C)$	187,1	220,4
$LM-(Tr+Web+Np\_complet)$	185,7	218,9

TABLE 5.11 – Perplexités et les données Tr, Web et Np utilisées pour l'apprentissage des MLs.

Aussi, vu que ces résultats peuvent être influencés par les quantités de données disponibles pour chaque type, leur taille est présentée dans le tableau 5.12. Dans cette phase,

$Fichiersdedonnes$	$taille(Mots[M])$
LEMONDE.A	204,5
LEMONDE.B	166,5
LEMONDE.C	14,0
LEMONDE_complet	**385**
LHUMANITE.A	61,5
LHUMANITE.B	65,0
LHUMANITE.C	14,5
LHUMANITE_complet	**141**

TABLE 5.12 – Taille par mots (sans <s> et </s> des fichiers Np.

nous avons essayé d'ignorer les fichiers WEB (y compris les fichiers WEBJOURNAUX.C) lors de l'apprentissage des modèles de langage et de considérer seulement les fichiers Np avec les données de Tr. Nous remarquons que le gain possible en terme de perplexité est de 19.3 (passer de 253 à 233.7, voir le tableau 5.13).

Le plus intéressant, dans tout ces résultats, est que l'influence des données textuelles extraites des journaux écrits sur les modèles de langage n'est pas aussi intéressante que celle des données extraites du Web.

ML	**ppl** $MLDev$	**ppl** $MLTest$
$ML - Tr$	215,6	253,0
$ML - (Tr + Np)$	198,7	233,7

TABLE 5.13 – Perplexités de ML appris avec les données de Tr et Np.

Données textuelles Gw

Vu l'espace mémoire insuffisant sur la machine disponible (i5), il est impossible d'utiliser les 4 sources de données (Tr, Web, Np et Gw) à la fois pour la modélisation du langage. nous nous sommes contentés d'utiliser les deux sources Tr et Gw. Notre but est d'avoir une idée sur l'influence des différents fichiers du Gw sur l'apprentissage des modèles de langage et de répondre sur les questions suivantes :

— Quelles sources de données contribuent le mieux pour la modélisation du langage : l'AFP (*Agence France Presse*), l'APW (*Association Press World stream, French service*) ou les GigaHead ?
— Quelle est la période des données du corpus Gw la plus pertinente : l'ancienne (.A), la moyenne (.B) ou la récente (.C)) ?

Quelques expériences sont menées et leurs résultats sont récapitulés dans le tableau 5.14.

ML	**ppl** $MLDev$	**ppl** $MLTest$
$ML - Tr$	215,6	**253,0**
$ML - (Tr + Gw\_AFP)$	206,8	243,3
$ML\_(Tr + Gw\_APW)$	206,3	243,7
$ML\_(Tr + Gw\_(AFP, APW))$	204,2	241,3
$ML\_(Tr + Gw\_(.A))$	210,2	246,8
$ML\_(Tr + Gw\_(.B) sauf - GHead.B))$	206,7	243,2
$ML\_(Tr + Gw\_(.B) y - compris - GHead.B))$	206,3	242,7
$ML\_(Tr + Gw\_(.C))$	207,5	245,5
$ML\_(Tr + Gw\_complet)$	204,0	**241,4**

TABLE 5.14 – Perplexités des ML appris sur les données de Tr et Gw.

Selon les résultats obtenus, les données tirées de l'APW assure une pertinence presque similaire à celle des données de l'AFP pour l'apprentissage des modèles de langage (voir lignes 2 et 3 du tableau 5.14).

Concernant les trois périodes temporelles de récolte des données Gw, la moyenne (avec presque 450 M de mots y compris le GIGAHEAD.B) est plus intéressante que la récente (177 M de mots) et l'ancienne (151 M de mots), selon les résultats du tableau 5.14 (lignes 6 à 9).

Les tailles des différents fichiers du corpus Gw sont présentées dans le tableau 5.15.

Encore une fois, nous pouvons noter que l'apprentissage des modèles de langage est fortement influencé par les quantités de données textuelles disponibles. Sachant que ces dernières ne sont pas équilibrées en taille sur les différentes périodes temporelles, il n'est pas évident de trancher sur ce point et confirmer quel type de données est le mieux

$Fichiersdesdonnes$	$Taille(Mots[M])$
GWAFPFRE.A	115,5
GWAFPFRE.B	317,0
GWAFPFRE.C	100,5
GWAPWFRE.A	37,0
GWAPWFRE.B	110.,5
GWAPWFRE.C	77,6
GIGAHEAD.C	23,5

TABLE 5.15 – Taille par mots sans <s> et </s> des fichiers Gw.

adapté pour nos modèles de langage dédiés à la transcription des émissions radiodiffusés. Cependant, il est intéressant d'utiliser les quatre sources principales en totalité.

Une telle analyse donne une stratégie claire pour enrichir les corpora textuels de LORIA. Malgré que l'acquisition des données textuelles n'est pas une tâche simple mais il est utile d'essayer de combler le manque de données existant en point de vue : sources et période pour disposer d'une base de données plus équilibrée et plus riche. Aussi, ces remarques peuvent être utile lors de la récolte des données pour d'autres systèmes de transcription dédiés à d'autres tâches.

6.7 Nouveaux critères proposés pour la sélection des données textuelles

Pour la sélection des données textuelles, les critères utilisés sont : l'aléatoire, à base de la perplexité *ppl*, la différence du log-probabilité *dLog* et la différence de l'entropie croisée $dXent$ proposée dans [Moore and Lewis, 2010].

Nous rappelons que la sélection par le critère de la perplexité *ppl* des phrases estimée sur le $ML - Tr$ donne un rendement moins important que la simple sélection aléatoire. Cependant le critère du log-probabilité *dLog* donne des résultats encourageants. Le sélection à base de la différence d'entropie croisée appliquée sur n'importe quel corpus donne les meilleurs résultats relativement aux deux modèles de langage utilisés comme domaine spécifique et domaine général (dites modèles de langage : *in-domain* et *non-domain*).

Dans cette section, nous testons d'autres critères de sélection, en considérant toujours le modèle de langage $ML - Tr$ comme modèle représentant le domaine spécifique mais différents choix sont à explorer pour le modèle de langage représentant le domaine général. Limités par nos capacités matérielles, nous nous contentons d'utiliser seulement 2 sources de données à la fois.

Le premier critère que nous proposons est la différence de perplexité $dPpl$, avec le principe d'évaluer la perplexité des phrases du corpus à sélectionner, sur deux modèles de langage $ML - Tr$ (comme modèle *in-domain*) et un deuxième modèle de langage général (comme *non-domain*).

Formellement, pour chaque phrase s du corpus à sélectionner, nous calculons une différence de perplexité évaluée entre les deux modèles de langage *in-domain* et *non-domain*, avec la formule :

$$dPpl(s) = ppl_{LM-in-domain}(s) - ppl_{LM-non-domain}(s) \tag{5.5}$$

A chaque fois on fixe une valeur pour le critère $dPpl$, un modèle de langage est entraîné avec les phrases sélectionnées avec des valeurs de $dPpl$ inférieure au seuil précédemment fixé.

Sélection de données textuelles à base de $MSDP$

Pour le deuxième critère, au lieu d'utiliser une simple différence entre les scores, nous nous sommes inspirés de la formule d'erreur quadratique utilisée lors de l'apprentissage des réseaux de neurones.

Rappelons que pour la classification par les réseaux de neurones, l'ensemble d'apprentissage est constitué de données pour lesquels la sortie désirée l (pour label) est une étiquette relative à la classe d'appartenance des données. La fonction de coût la plus utilisée est l'erreur quadratique sur la base d'apprentissage : elle consiste à minimiser la somme des carrés des erreurs entre la sortie du réseaux et la valeure réelle de la sortie supervisée. Elle est considérée comme la fonction *objectif* de l'algorithme de rétroporpagation du réseau de neurones multi-couches (MLP), dont l'équation est :

$$MSE(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i(w) - l_i)^2 \tag{5.6}$$

Cette fonction de coût est issue du principe de maximum de vraisemblance avec une hypothèse gaussienne sur la distribution des sorties.

Notre but est d'exploiter ce critère pour la sélection des données textuelles pour la modélisation du langage. Nous considérons que le score estimé sur les données de Tr est la sortie désirée et celui estimé sur l'une des autres sources Web, Np ou Gw sont les sorties réelles obtenues, à chaque fois.

Pour chaque phrase s du corpus à sélectionner, nous calculons une erreur moyenne quadratique $MSDP$ entre les log-probabilités de cette phrase évaluées sur les deux modèles de langage *in-domain* et *non-domain* (voir la figure 5.10), avec la formule suivante :

$$MSDP(s) = \frac{1}{2}(logProb_{LM-in-domain}(s) - logProb_{LM-non-domain}(s))^2 \tag{5.7}$$

A Chaque fois nous fixons une valeur pour le $MSDP$, un modèle de langage interpolé est formé à l'aide des sous-ensembles des phrases sélectionnées ayant des $MSDP$ inférieures ou égales au seuil considéré.

Sélection sur les données textuelles Web

Sachant que le corpus Web à un poids 0.246 dans la combinaison linéaire du modèle réferentiel présenté dans le tableau 5.1, nous effectuons une sélection sur ce corpus par :

1. la sélection aléatoire (*random*) sur le corpus Web ;
2. la sélection par le critére de perplexité estimé sur le $ML - Tr$;
3. la sélection par le critère de la différence de l'entropie croisée $dXent$ estimée pour toutes les phrases du Web par les deux modèles de langage $ML - Tr$ et $ML - WebTiny$ en utilisant la formule :

$$dXent(s) = H_{(ML-Tr)}(s) - H_{(ML-WebTiny)}(s) \tag{5.8}$$

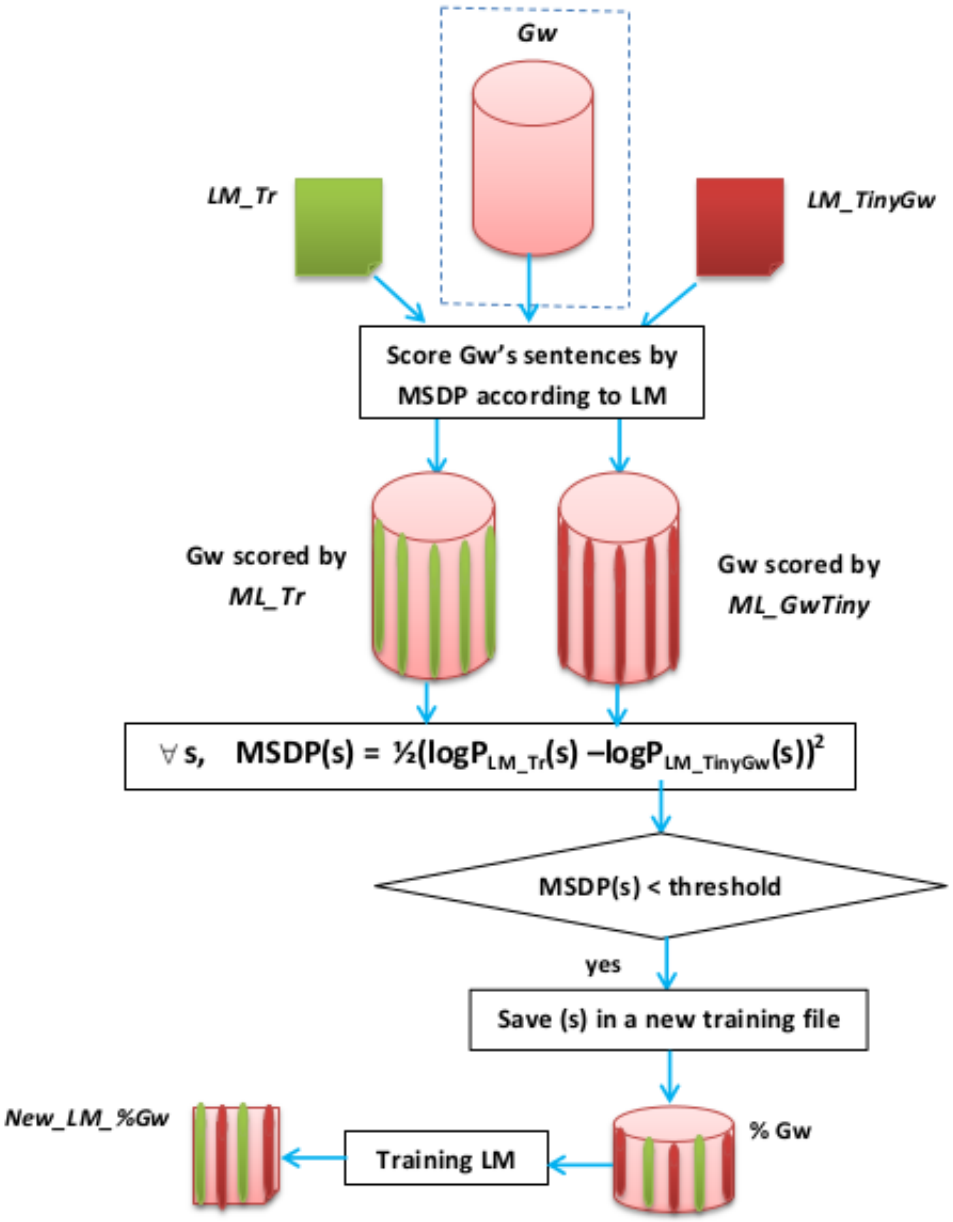

FIGURE 5.10 – Principe de la sélection des données par le critère de $MSDP$

4. la sélection par l'erreur moyenne quadratique $MSDP$ de la log-probabilité de toutes les phrases du Web estimées sur les deux modèles de langage $ML-Tr$ et $ML-WebTiny$, par la formule suivante :

$$MSDP(s) = \frac{1}{2}(logprob_{(LM-WebTiny)}(s) - logprob_{(LM-Tr)}(s))^2 \qquad (5.9)$$

Cependant, il reste faible d'où nous nous contentons de le tester que pour ce corpus.

Les résultats de la sélection par le critère de $MSDP$ a conduit à un minimum de perplexité (de **308,47**) tandis que le critère de $dXent$ à éteint un minimum de **309.15**.

Aussi nous remarquons que l'allure des deux courbes est presque similaire. Nous pouvons justifier cela par le fait que les données du Web ont une distribution probabiliste très proches des données de Tr, ce qui a permit au critère $MSDP$ d'obtenir des résultats proches et même meilleurs que ceux obtenus par la $dXent$.

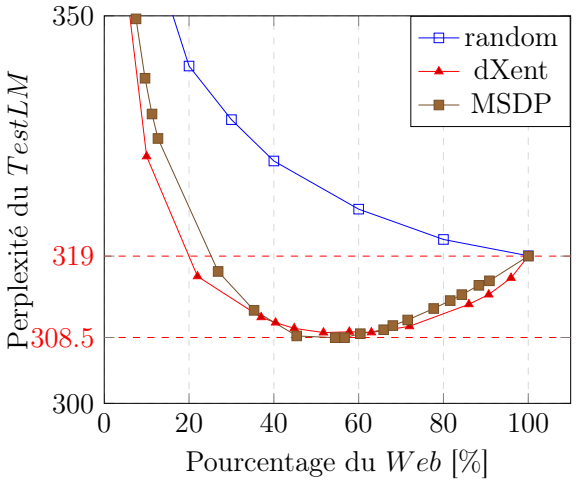

FIGURE 5.11 – Perplexité du ML-Test appris sur les données sélectionnées du Web.

La sélection est appliquée sur les données du Web par : (1) la sélection aléatoire (random), (3) la différence de l'entropie croisée $dXent$ calculée sur modèles $ML-Tr$ et $ML-WebTiny$, (5) la différence quadratique du log-probabilité $MSDP$ calculée entre les modèles $ML-Tr$ et $ML-WebTiny$.

Sélection des données textuelles Np

Sachant que le corpus Np à un poids de $0,062$ dans la combinaison linéaire du modèle de *Baseline*, nous effectuons une sélection sur ce corpus par différentes façons (voir la figure 5.12) :

1. la sélection aléatoire (random) sur le corpus Np ;
2. la sélection par $dXent$ estimée pour toutes les phrases du Np par le deux modèles de langage : $ML-Tr$ développé par les données du Tr (114 M de mots) et ML_NpTiny développé par 114 M de mots extraites aléatoirement du Np (parmi les 526 M mots), en utilisant la formule suivante :

$$dXent(s) = H_{(LM-Tr)}(s) - H_{(LM-NpTiny)}(s) \tag{5.10}$$

3. la sélection par la formule d'erreur moyenne quadratique $MSDP$ de la probabilité des deux MLs $LM-Tr$ et $LM-NpTiny$, estimée pour toutes les phrases du Np par la formule suivante :

$$MSDP(s) = \frac{1}{2}(logprob_{(LM-Tr)}(s) - logprob_{(LM-NpTiny)}(s))^2 \tag{5.11}$$

Les résultats de la sélection sur le corpus Np par le critère de $MSDP$, illustrés dans la figure **??**, conduisent à un minimum de perplexité de $404,35$ pendant que le critère de $dXent$ atteint un minimum de $376,23$. Aussi nous remarquons que l'allure des deux courbes est similaire de 35% à 100% du corpus Np.

Sélection des données textuelles du Gw

Sachant que le corpus Gw à un poids de 0.007 dans la combinaison linéaire du modèle référentiel, nous effectuons une sélection sur ce corpus par :

1. la sélection aléatoire (random) sur le corpus Gw ;

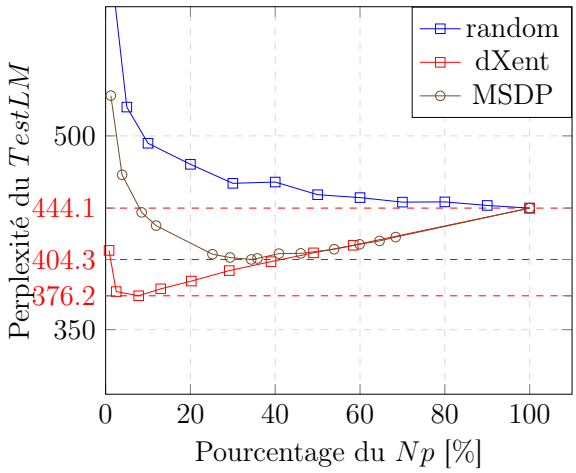

FIGURE 5.12 – Perplexité du ML-Test appris sur les données sélectionnées du Np.

La sélection est appliquée sur les données du Np par : (1) la sélection aléatoire (random), (2) la différence de l'entropie croisée $dXent$ calculée sur modèles $ML-Tr$ et $ML-NpTiny$, (3) la différence quadratique du log-probabilité $MSDP$ calculée entre les modèles $ML-Tr$ et $ML-NpTiny$.

2. la sélection par $dXent$ estimée pour toutes les phrases du Gw par le deux MLs : $ML-Tr$ développé par les données du Tr (114 M de mots) et le $ML-GwTiny$ ($Tiny$ pour préciser que ce modèle de langage est réduit ou petit) développé par 114 M de mots extraites aléatoirement du corpus Gw (parmi les 783 M mots), en utilisant la formule suivante :

$$dXent(s) = H_{(LM-Tr)}(s) - H_{(LM-GwTiny)}(s) \tag{5.12}$$

3. la sélection par la formule d'erreur moyenne quadratique $MSDP$ des log-probabilités des phrases Gw évaluées sur les deux modèles de langage $LM-Tr$ et $LM-GwTiny$, par la formule :

$$MSDP(s) = \frac{1}{2}(logprob_{(LM-GwTiny)}(s) - logprob_{(LM-Tr)}(s))^2 \tag{5.13}$$

Les résultats de la sélection sur le corpus GW par le critère de $MSDP$ a conduit à un minimum de perplexité (de 532.3) pendant que le critère de $dXent$ à éteint un minimum de 454.7. Aussi nous remarquons que l'allure des deux courbes est similaire (sur l'axe des abscisses) de 30% à 100% du corpus Gw, à vrai dire la sélection par $MSDP$ est meilleure que celle par $dXent$.

Selon Xu [Xu et al., 2005], le critère $MSDP$ est un critère populaire dans la formation des tout les systèmes adaptatifs, y compris les réseaux de neurones artificiels. Les deux principales raisons de ce choix sont la traçabilité analytique et l'hypothèse selon laquelle les phénomènes aléatoires de la vie réelle peuvent être suffisamment décrits par des statistiques de second ordre. La fonction de densité de probabilité gaussienne (pdf) est déterminée uniquement par ses statistiques de premier et second ordre, et l'effet des systèmes linéaires sur les statistiques d'ordre faible est bien connu.

Sous ces hypothèses de linéarité et de gaussianité, appuyée par le théorème de limite centrale, $MSDP$, qui limite uniquement les statistiques de second ordre, serait capable d'extraire toute information possible d'un signal dont les statistiques sont uniquement définies par sa moyenne et sa variance. D'autre part, MSDP peut extraire toutes les informations dans les données fournies que le système dynamique est linéaire et le bruit

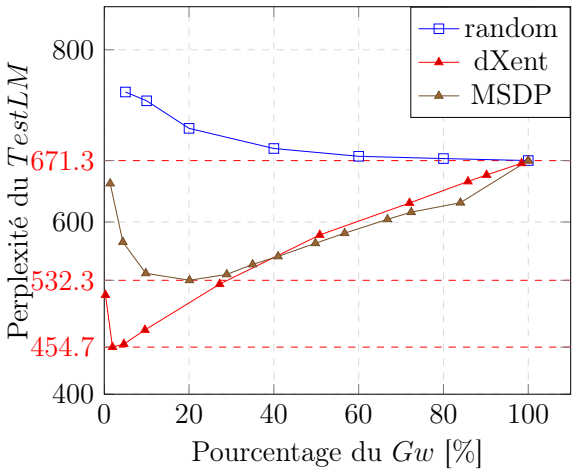

FIGURE 5.13 – Perplexité du ML-Test appris sur les données sélectionnées du Gw.

La sélection est appliquée sur les données du Gw par : (1) la sélection aléatoire (random), (2) la différence de l'entropie croisée $dXent$ calculée sur modèles $ML-Tr$ et $ML-GwTiny$, (3) la différence quadratique du log-probabilité $MSDP$ calculée entre les modèles $ML-Tr$ et $ML-GwTiny$.

est Gaussien distribué. Cependant, lorsque le système devient non linéaire et que la distribution de bruit est non gaussienne, la $MSDP$ ne parvient pas à capturer toutes les informations dans les séquences d'erreur. Dans ce cas, un critère alternatif nécessaire pour atteindre l'optimalité est l'entropie croisée. Cette dernière est une extension naturelle au-delà de $MSDP$ puisque l'entropie est une fonction de la fonction de densité de probabilité (pdf), qui considère toutes les statistiques d'ordre élevé.

Nous pouvons justifier les résultats obtenus par le fait que, une part des données du NP et du Gw ont une distribution probabilistique non linéaire par rapport aux données de Tr. Il est difficile au critère $MSDP$ d'obtenir des résultat meilleurs (tout au long des courbes 5.13) par rapport aux sélections obtenues par la $dXent$.

Combinaison de multi-sources textuelles

Les meilleurs modèles de langage obtenus en combinant les données sélectionnées sur les différentes sources (Web, Np et Gw) au niveau des meilleurs points de sélection sont récapitulés dans les tableaux 5.16 5.17 pour la sélection à base de $dXent$ et dans les tableaux 5.18 et 5.19 pour la sélection à base de $MSDP$. Dans la plupart des cas les résultats de sélection obtenus par le critère de $MSDP$ sont similaires (et quelques fois inférieurs) à ceux obtenus par le critère de $dXent$.

Sources	**ML individuel**	**ML interpolé**		
	ppl $DevLM$	**poids**	**ppl** $DevLM$	**ppl** $TestLM$
Tr 100%	215,6	0,6795	187,165	220,468
Web 100%	264,7	0,2840		
Np 8%	312,34	0,0152		
Gw 2%	372,2	0,0213		

TABLE 5.16 – ML développé par 100% du Tr, 100% du Web, 8% du Np et 2% du Gw sélectionné par la différence de l'entropie croisée $dXent$

Sources	**ML individuel**	**ML interpolé**		
	ppl *DevLM*	**poids**	**ppl** *DevLM*	**ppl** *TestLM*
Tr 100%	215,6	0,6734	185,736	218,822
Web 100%	264,7	0,2453		
Np 100%	364,2	0,0628		
Gw (2%)	372,2	0,0186		

TABLE 5.17 – ML développé par 100% de *Tr*, 100% du *Web*, 100% du *Np* et 2% *Gw* sélectionnés par le critère de *dXent*.

Sources	**ML individuel**	**ML Interpolé**		
	ppl *DevLM*	**poids**	**ppl** *DevLM*	**ppl** *TestLM*
Tr 100%	215,6	0,6838	186,785	220,072
Web 100%	264,7	0,2702		
Np 34%	336,2	0,0375		
Gw 20%	419,2	0,0085		

TABLE 5.18 – ML développé par 100% du *Tr*, 100% du *Web*, 34% du *Np* et 20% du *Gw* sélectionnés par le critère de *MSDP*.

Sources	**ML individuel**	**ML interpolé**		
	ppl *DevLM*	**poids**	**ppl** *DevLM*	**ppl** *TestLM*
Tr 100%	215,6	0,6837	185,757	218,908
Web 100%	264,7	0,2476		
Np 100%	364,2	0,0632		
Gw 20%	419,2	0,0055		

TABLE 5.19 – ML développé par 100% du *Tr*, 100% du *Web*, 100% du *Np* et 20% du *Gw* sélectionnés par le critère *MSDP*.

Nous avons pu obtenir des modèles de langage à perplexités similaires au ML référentiel *Baseline*, avec des tailles inférieures 815, 3 Mo pour le ML présenté dans le tableau 5.17 et 880, 1 Mo pour le ML présenté dans le tableau 5.19 au lieu de 1.2 Go du ML référentiel présenté dans 5.1 (entrainé sur la totalités des données textuelles : environ 2 milliards de mots).

Aussi nous avons exploité le $MSDP$ comme un nouvel critère pour la sélection des données et nous avons comparé ses performances au critère de $dXent$ de Moore.

Sélection multi-sources textuelles

Nous avons essayé d'appliquer la sélection à base de $MSDP$ sur les données des 4 sources (Tr+Web+Np+Gw) à la fois, par la formule :

$$MSDP(s) = \frac{1}{2}(logprob_{(LM-Gw)}(s) - logprob_{(LM-TrWebNp)}(s))^2 \tag{5.14}$$

ou en appliquant la sélection seulement sur les données du Gw par la formule :

$$MSDP(s) = \frac{1}{2}(logprob_{(LM-Gw)}(s) - logprob_{(LM-TrWebNp)}(s))^2 \tag{5.15}$$

et les combiner aves les autres sources (Tr, Web, Np) complètes. Malheureusement, l'insuffisante matérielle vu l'importance du corpus complet et la complexité du déroulement du script gène l'exploitation des données en totalité.

7 Conclusion

Il existe plusieurs méthodes statistiques pour améliorer les systèmes de RAP : l'utilisation des mesures de confiance, l'analyse des erreurs, l'utilisation de l'apprentissage profond pour l'apprentissage des modèles acoustiques et/ou les modèles de langage, la sélection des données audio et/ou textuelles, etc.

A notre niveau, nous avons exploité la richesse, la complexité et les avantages du domaine de la RAP, et plus précisément du LVCSR [11]. Nos travaux se sont focalisés sur les modèles de langage dédiés à la transcription des émissions de radio et de télévision de la compagne d'évaluation ETAPE. Aussi, nous avons présenté et analysé la sélection de données multi-sources pour l'apprentissage des modèles de langage dédiés à cette tâche.

La perplexité du test pour le modèle de langage entraîné sur les corpora de transcriptions manuelles, données du web, les données des journaux et le Gigaword français noté respectivement ($Tr + Web + Np + Gw$), qui est notre modèle de langage référentiel de base, est de 218.9. Nous avons remarqué que les corpus Tr et Web sont les plus proches de notre tâche et malheureusement l'énorme corpus Gw contient beaucoup de données hétérogènes et non pertinentes. Le fait de conserver les trois sources de données (Tr, Web et Np) et de sélectionner les données du corpus Gw avec la différence d'entropie croisée $dXent$ conduit à de meilleurs résultats que lors du choix de données aléatoires ou avec une sélection à base de différence d'entropie croisée sur toutes les données (Tr, Web, Np et Gw) à la fois.

Aussi, une perplexité optimale de 210.5 est obtenue avec un modèle de langage construit à partir de 55.4 % des données sélectionnées du corpus All = ($Tr + Web + Np + Gw$).

11. http ://www.speech.sri.com/projects/lvcsr/, consulté janv. 2017

La meilleure amélioration est d'environ 8.3 en termes de perplexité, et cela se traduit par une réduction de 0.2 % absolue en termes de WER pour le système de transcription.

Cette partie du travail mène à plusieurs conclusions intéressantes. Tout d'abord, le choix des modèles qui représentent les données du domaine considéré (*in-domain*) et du domaine général (*non-domain*) est important. Les résultats sont très différents selon les l'approches considérées 1, 2 et 3.

Il semble que le corpus Gigaword Gw n'est pas très utile pour la modélisation du langage pour notre tâche. Nous supposons que ce n'est pas vrai en raison de la grande couverture de ce corpus. Par conséquent, il faut explorer de façons différentes les données de ce corpus Gw afin d'améliorer les performances de la modélisation du langage.

Il est important de noter que lors de la manipulation des données textuelles, nous avons remarqué que les phrases du Gw sont longues en les comparant avec les phrases des autres corpora (Tr, Np et Web). En considérant que la longueur du contexte peut influencer la modélisation du langage, il est intéressant de considérer des n-grammes plus importants : 4,5,6 ou 7 pour une meilleure modélisation ou se baser sur les modèles de langage neuronaux. Cependant, le système de transcription de LORIA standard est basé sur des modèle de langage 3-gramme, d'où il sera inutile d'avancer dans ce sens.

Par la suite, deux critères pour la sélection des données textuelles sont proposés et utilisés : la différence de la perplexité $dPpl$ et l'erreur moyenne quadratique des log-probabilités $MSDP$. Le premier critère $dPpl$ est relativement intéressant mais reste faible devant la différence de l'entropie croisée $dXent$. Le $MSDP$ est un critère populaire dans la formation des réseaux de neurones artificiels. Selon les résultats obtenus, ce critère $MSDP$ ne parvient pas à capturer toutes les informations utiles mais il se rapproche des compétences du critère $dXent$.

Chapitre 6

Conclusion générale

1 Conclusion

Dans un système de reconnaissance de la parole (SRAP), les *modèles acoustiques* sont sensés de modéliser la totalité de l'espace acoustique et la modéles de langage sont utilisés pour résumer les contraintes linguistiques liées à une langue naturelle. Avec l'utilisation de larges vocabulaires, les systèmes LVCSR (*Large Vocabulary Conversational Speech Recognition*) actuels sont plus puissants que de simple SRAP : ils sont entrainés sur des centaines d'heures de parole et des milliards de mots de texte. Cependant, ils ne sont encore pas robustes aux conditions réélles de test et ils restent loin des capacités des auditeurs humains. Il existe plusieurs méthodes statistiques pour contribuer à l'amélioration des performances de ces systèmes [1] : l'utilisation des mesures de confiance, l'analyse des erreurs, l'utilisation de l'apprentissage profond pour l'apprentissage des modèles acoustiques et/ou les modèles de langage, la sélection des données audio et/ou textuelles, etc.

Le grand développement dans ce domaine est réalisé essentiellement par les groupes de recherche industriels : Google, Facebook, Yahoo, etc. et il est fortement lié au développement matériel : le parallélisme, l'utilisation des processeurs GPU en plus des CPU, et les énormes bases de données audio et/ou textuelles disponibles. Les laboratoires universitaires viennent par la suite, en deuxième position, pour contribuer avec des idées théoriques et des concepts pratiques importants.

Dans cette thèse nous nous sommes intéressés, dans un cadre applicatif, à la modélisation du langage pour la reconnaissance automatique de la parole spontanée et conversationelle à grand vocabulaire. Nous avons commencé par présenter la science de la linguistique avec ses différentes branches et domaines qui coïncident fortement avec le domaine de la reconnaissance automatique de la parole. Ensuite, nous avons exposé l'état de l'art du domaine la reconnaissance automatique de la parole à large vocabulaire, qui rencontrent actuellement une forte popularité auprès de la communauté industrielles et scientifique. Nous avons introduit particulièrement les nouvelles approches de modélisation et d'adaptation statistiques utilisées dans ce domaine.

Nous avons étudié les différentes stratégies possibles de modélisation de langage avec les modèles standards n-gramme. nos travaux été présentées lors d'une conférence internationale [Mezzoudj et al., 2015b]. Aussi, nous nous sommes intéressés aux modèles de langage avancés à base de réseaux de neurones et l'apprentissage profond, le fruit des ces travaux est publié dans un article (Mezzoudj et Benyettou, 2018 en presse). Les travaux réalisés et décrits se basent sur des données diffusées lors de la campagne d'évaluation

1. http ://www.speech.sri.com/projects/lvcsr/, consulté en fev. 2017

ETAPE, qui constitue un contexte réel de l'étude des systèmes de la reconnaissance automatique de la parole spontanée et/ou conversationnelle à grand vocabulaire.

Aussi, nous avons exploité la sélection des données textuelles pour améliorer la modélisation du langage dédiée à la reconnaissance de la parole en utilisant différents critères, et plus précisément le critère de l'entropie croisée. Grâce aux expériences que nous avons réalisées, nous avons pu démontrer l'efficacité de cette sélection pour l'amélioration du système de reconnaissance, par les résultats obtenues. Cette partie de travail a fait l'objet d'une contribution à la conférence internationnale ICNLSP (*International Conference on Natural Language and Speech Processing, Algeria*) [Mezzoudj et al., 2015a].

Enfin, nous portons l'accent sur l'importance de disposer de données audio et textuelles énormes et pertinentes et du matériels puissants. Ces deux clés permettent de se lancer dans le domaine de la reconnaissance automatique de la parole continue et spontanée à grand vocabulaire.

2 Perspectives

À partir des contributions exposées ici et des axes de recherches définis en introduction, plusieurs perspectives de recherche peuvent être envisagées.

— La poursuite des recherches dans le cadre de la sélection de données textuelles et/ou audio pour la modèlisation de la parole, basée sur différents critères ainsi que dans le cadre de l'acquisition de ces données via l'apprentissage non-supervisé.

— Sachant qu'il est coûteux et difficile de se procurer de nouveaux corpora de parole audio et textuelles de taille suffisante, afin de construire des systèmes de LVSCR standard. Nous envisageant de consacrer les compétences que nous avons acquis durant la préparation de cette thèse pour contribuer à la réalisation d'un système de transcription des émissions radiophoniques et télévisées pour la langue Arabe.

Contribution Personnelle

F. Mezzoudj and A. Benyettou. On the optimization of multiclass support vector machines dedicated to speech recognition. In T. Huang, Z. Zeng, C. Li, and C. Leung, editors, Neural Information Processing, volume 7664 of Lecture Notes in Computer Science, pages 1–8. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-34480-0. doi : 10.1007/978-3-642-34481-7_1.

F. Mezzoudj, M. Loukam, and A. Benyettou. On an empirical study of smoothing techniques for a tiny language model. In Proceedings of IPAC 15 November 23-25, Batna, Algeria, pages 67–80. ACM, 2015. dx.doi.org/10.1145/2816839. 2816878.

F. Mezzoudj, D. Langlois, D. Jouvet, and A. Benyettou. Textual data selection for language modelling in the scope of automatic speech recognition. In International Conference on Natural Language and Speech Processing, Algeria, 2015.

F. Mezzoudj, D. Langlois, D. Jouvet, and A. Benyettou. Textual data selection for language modelling in the scope of automatic speech recognition. In Procedia Computer Science (2018). 10-APR-2018. pp. 55-64. DOI information : 10.1016/j.procs.2018.03.008

F. Mezzoudj and A. Benyettou. Textual data selection based on mean square difference probability for language medelling. 3rd CITIM, 9-10 October 2018. Mascara. Algeria.

F. Mezzoudj and A. Benyettou. 'An empirical study of statistical language models : n-grams language models vs. neural network language models'. In International Journal of Innovative Computing and Applications, Vol. 9, No.4, (2018). pp. 189-202. Doi : 10.1504/I-JICA. 2018.10016827.

Bibliographie

A. Acero. *Acoustical and environmental robustness in automatic speech recognition*. PhD thesis, Carnegie Mellon University Pittsburgh, 1990.

M. Adda-Decker. Corpus pour la transcription automatique de l'oral. *Revue française de linguistique appliquée 1/2007 (Vol. XII) , p. 71-84*, 2007.

N. Alcaraz Meseguer. Speech analysis for automatic speech recognition. 2009.

M. Alghamdi. *Arabic Phonetics, en Arabe.* phonetic. Attaoobah, Riyadh, 2000. URL `https://books.com`.

A. Allauzen and J.-L. Gauvain. Misea jour automatique du modele de langage d'un systeme de transcription. *Proc. of XXIVièmes Journée d'Etude sur la Parole*, pages 305–308, 2002.

A. Allauzen and J.-L. Gauvain. Construction automatique du vocabulaire d'un système de transcription. *Journées d'Etude sur la Parole*, 2004.

C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. Openfst : A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer, 2007.

T. Alumäe and M. Kurimo. Efficient estimation of maximum entropy language models with N-gram features : an SRILM extension. In *Proceedings of Interspeech 2010*, Chiba, Japan, September 2010.

T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 1137–1140. IEEE, 1996.

T. Anastasakos, J. McDonough, and J. Makhoul. Speaker adaptive training : A maximum likelihood approach to speaker normalization. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1043–1046. IEEE, 1997.

J. Andrés-Ferrer, M. Sundermeyer, and H. Ney. Conditional leaving-one-out and cross-validation for discount estimation in kneser-ney-like extensions. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5013–5016. IEEE, 2012.

X. L. Aubert. A brief overview of decoding techniques for large vocabulary continuous speech recognition. In *ASR2000-Automatic Speech Recognition : Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

X. L. Aubert. An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech & Language*, 16(1) :89–114, 2002.

A. Axelrod, X. He, and J. Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics, 2011.

J. K. Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1) :S132–S132, 1979.

M. Barkat. Détermination d'indices acoustiques robustes pour l'identification automatiques des parlers arabes, thèse de doctorat, université lumière lyon, 2000.

C. Barras, E. Geoffrois, Z. Wu, and M. Liberman. Transcriber : a free tool for segmenting, labeling and transcribing speech. In *First international conference on language resources and evaluation (LREC)*, pages 1373–1376, 1998.

L. Barrault. *Diagnostic pour la combinaison de systèmes de reconnaissance automatique de la parole.* PhD thesis, Université d'Avignon, 2009.

L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In O. Shisha, editor, *Inequalities III : Proceedings of the Third Symposium on Inequalities*, pages 1–8, University of California, Los Angeles, 1972. Academic Press.

L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, pages 1554–1563, 1966.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, pages 164–171, 1970.

T. Bazillon. *Transcription et traitement manuel de la parole spontanée pour sa reconnaissance automatique.* PhD thesis, Université du Maine, 2011.

F. Béchet. Lia phon : un systeme complet de phonétisation de textes. *Traitement automatique des langues*, 42(1) :47–67, 2001.

N. Béchet. *Extraction et regroupement de descripteurs morpho-syntaxiques pour des processus de Fouille de Textes.* PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2009.

J. R. Bellegarda. Statistical language model adaptation : review and perspectives. *Speech communication*, 42(1) :93–108, 2004.

Y. Bengio and S. Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. In *NIPS*, volume 99, pages 400–406, 1999.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. In *NIPS, 2000*, pages 933–938. IEEE, 2001.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb) :1137–1155, 2003.

R. Besson. *Sono et prise de son- 3e édition.* Audio-Vidéo. Dunod, version 1 disponible à UHBC F02-09-007, 2004. ISBN 2 10 004351 X. URL `https://books.google.dz/books?id=I0W5EY_CuMC`.

J. Billa, M. Noamany, A. Srivastava, D. Liu, R. Stone, J. Xu, J. Makhoul, and F. Kubala. Audio indexing of arabic broadcast news. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–5. IEEE, 2002.

J. Blitzer, K. Q. Weinberger, L. K. Saul, and F. C. Pereira. Hierarchical distributed representations for statistical language modeling. *Advances in Neural Information Processing Systems*, 17 :185–192, 2005.

P. Boersma and V. van Heuven. Speak and unspeak with praat. *Glot International*, 5 (9-10) :341–347, 2001.

G. Bohas. *Développements récents en linguistique arabe et sémitique.* Presses de l'Ifpo, 2014.

R. Boite. *Traitement de la parole- 3e édition.* électricité. Presses polytechniques et universitaires romandes, 2000. URL `https://books.google.dz/books?id=I0W5EY_CuMC`.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

L. Bottou. Stochastic gradient descent tricks. In *Neural Networks : Tricks of the Trade*, pages 421–436. Springer, 2012.

T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* Citeseer, 2007.

J. Brousseau, J.-F. Beaumont, G. Boulianne, P. Cardinal, C. Chapdelaine, M. Comeau, F. Osterrath, and P. Ouellet. Automated closed-captioning of live tv broadcast news in french. In *INTERSPEECH*, 2003.

P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4) :467–479, 1992.

A. Brun, C. Cerisara, D. Fohr, I. Illina, D. Langlois, O. Mella, and K. Smaïli. Ants : le système de transcription automatique du loria. In *Journées d'Etude sur la Parole-Journée d'Etude sur la Parole'04*, pages 4–p, 2004.

P. Cardinal, A. M. Ali, N. Dehak, Y. Zhang, T. Al Hanai, Y. Zhang, J. R. Glass, and S. Vogel. Recent advances in asr applied to an arabic transcription system for al-jazeera. In *INTERSPEECH*, pages 2088–2092, 2014.

J. Catineau. *Études de linguistique arabe*, volume 2. Librairie C. Klincksiek, 1960.

C. Chelba and F. Jelinek. Structured language modeling. *Computer Speech & Language*, 14(4) :283–332, 2000.

S. F. Chen. Building probabilistic models for natural language. *arXiv preprint cmp-lg/9606014*, 1996.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4) :359–393, 1999.

S. F. Chen, D. Beeferman, and R. Rosenfeld. Evaluation metrics for language models. 1998.

K. Cho. Natural language understanding with distributed representation. *arXiv preprint arXiv :1511.07916*, 2015.

N. Chomsky. *Structures syntaxiques*, volume 98. Editions du SEUIL, 1969.

J. Cohen, T. Kamm, and A. G. Andreou. Vocal tract normalization in speech recognition : Compensating for systematic speaker variability. *The Journal of the Acoustical Society of America*, 97(5) :3246–3247, 1995.

R. Collobert and J. Weston. A unified architecture for natural language processing : Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12 (Aug) :2493–2537, 2011.

B. Cormons. *Analyse et désambiguïsation : Une approche à base de corpus(Data-Oriented Parsing) pour les représentations lexicales fonctionnelles*. PhD thesis, 2014.

Cours. phonétique en ligne., 2013. URL `http://www.unil.ch/ling/page13431.html`.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4) :303–314, 1989.

G. Dahl, A.-r. Mohamed, G. E. Hinton, et al. Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems*, pages 469–477, 2010.

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

N. Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. *International Journal for Advance Research in Engineering and Technology*, 1(6) :1–4, 2013.

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

F. De Saussure. *Cours de linguistique générale : Édition critique*, volume 1. Otto Harrassowitz Verlag, 1989.

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

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

L. Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3 :e2, 2014.

L. Deng and D. Yu. Deep learning. *Signal Processing*, 7 :3–4, 2014.

L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications : An overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603. IEEE, 2013.

R. Dufour. *Transcription automatique de la parole spontanée*. PhD thesis, Université du Maine, 2010.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2) :179–211, 1990.

A. Emami and F. Jelinek. Random clusterings for language modeling. In *ICASSP (1)*, pages 581–584, 2005.

S. Essid. *Classification automatique des signaux audio-fréquences : reconnaissance des instruments de musique*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2005.

Y. Estève. *Intégration de sources de connaissances pour la modélisation stochastique du langage appliquée à la parole continue dans un contexte de dialogue oral homme-machine*. PhD thesis, Université d'Avignon et des Pays de Vaucluse, 2002.

Y. Esteve, T. Bazillon, J.-Y. Antoine, F. Béchet, and J. Farinas. The epac corpus : Manual and automatic annotations of conversational speech in french broadcast news. In *LREC*, 2010.

P. G. et de Calmès M. Bdlex lexical data and knowledge base of spoken and written french. In *European Conference on Speech Technology (ECST), Édimbourg, Écosse, Royaume-Uni*, pages 1393–1396, 1987.

P. Fiala. Marie-anne paveau et georges-elia sarfati, les grandes théories de la linguistique. de la grammaire comparée à la pragmatique. *Mots. Les langages du politique*, (75) : 129–132, 2004.

É. Ficquet and A. Mbodj-Pouye. Cultures de l'écrit en afrique. anciens débats, nouveaux objets. In *Annales. Histoire, Sciences Sociales*, volume 64, pages 751–764. Éditions de l'EHESS, 2009.

D. Filimonov and M. Harper. A joint language model with fine-grain syntactic tags. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 3-Volume 3*, pages 1114–1123. Association for Computational Linguistics, 2009.

G. D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3) :268–278, 1973.

G. Foster, C. Goutte, and R. Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459. Association for Computational Linguistics, 2010.

P. Fousek, L. Lamel, and J.-L. Gauvain. On the use of mlp features for broadcast news transcription. In *International Conference on Text, Speech and Dialogue*, pages 303–310. Springer, 2008.

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

C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy, and D. Suendermann-Oeft. Comparing open-source speech recognition toolkits, 2014.

F. Gaillard, F. Berthommier, G. Feng, and J.-L. Schwartz. Une méthode modifiée de détection de pitch par passages par zéro en milieu interférant. In *16 Colloque sur le traitement du signal et des images, FRA, 1997*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1997.

M. J. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2) :75–98, 1998.

O. Galibert, J. Leixa, G. Adda, K. Choukri, and G. Gravier. The etape speech processing evaluation. *Proc of LREC, ELRA, Reykjavik, Iceland*, 2014.

O. Galibert, J. Kahn, and S. Rosset. Comparaison de listes d'erreurs de transcription automatique de la parole : quelle complémentarité entre les différentes métriques ? In *Journée d'Etude sur la Parole, JEP-TALN-RECITAL*, 2016.

S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Interspeech*, pages 1149–1152, 2005.

S. Galliano, E. Geoffrois, G. Gravier, J.-F. Bonastre, D. Mostefa, and K. Choukri. Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *Proceedings of LREC*, volume 6, pages 315–320, 2006.

S. Galliano, G. Gravier, and L. Chaubard. The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *Interspeech*, volume 9, pages 2583–2586, 2009.

J. Gao, H.-F. Wang, M. Li, and K.-F. Lee. A unified approach to statistical language modeling for chinese. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1703–1706. IEEE, 2000.

J. Gao, J. Goodman, M. Li, and K.-F. Lee. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1) :3–33, 2002.

J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *Speech and audio processing, ieee transactions on*, 2(2) :291–298, 1994.

J.-L. Gauvain, L. Lamel, G. Adda, and M. Jardino. Recent advances in transcribing television and radio broadcasts. In *EUROSPEECH*, 1999.

J.-L. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. *Speech communication*, 37(1) :89–108, 2002.

J.-L. Gauvain, G. Adda, L. Lamel, F. Lefèvre, and H. Schwenk. Transcription de la parole conversationnelle. *Actes des 25emes Journées d'Etudes sur la Parole (JEP)*, 2005.

Y. Goldberg. A primer on neural network models for natural language processing. *arXiv preprint arXiv :1510.00726*, 2015.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4) :237–264, 1953.

J. T. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, 15 :403–434, 2001.

R. Gopinath. Constrained maximum likelihood modeling with gaussian distributions. In *Proceedings Broadcast news transcription and understanding workshop*, 1998.

A. Gorin. *Structuration du modele acoustique pour améliorer les performances de la reconnaissance automatique de la parole.* PhD thesis, Université de Lorraine, 2014.

A. Gorin and D. Jouvet. Component structuring and trajectory modeling for speech recognition. In *Interspeech*, 2014.

A. Gorin, D. Jouvet, E. Vincent, and D. Tran. Investigating Stranded GMM for Improving Automatic Speech Recognition. In *4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA 2014)*, Nancy, France, May 2014. URL `https://hal.inria.fr/hal-01003054`.

G. Gravier and G. Adda. Evaluation en traitement automatique de la parole (etape), evaluation plan etape 2011 version 2.0.

G. Gravier, J.-F. Bonastre, E. Geoffrois, S. Galliano, K. McTait, and K. Choukri. The ester evaluation campaign for the rich transcription of french broadcast news. In *LREC*, 2004.

G. Gravier, G. Adda, N. Paulson, M. Carré, A. Giraudel, and O. Galibert. The etape corpus for the evaluation of speech-based tv content processing in the french language. In *LREC-Eighth international conference on Language Resources and Evaluation*, 2012.

A. Grognuz. Audition et surdité : Informations pour les enseignants, 2012.

R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 13–16. IEEE, 1992.

T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grézl, A. El Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan. Transcribing meetings with the amida systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2) :486–498, 2012.

F. Hamza and B. Halima. Etude comparative entre les librairies de reconnaissance vocale.

J.-P. Haton, C. Cerisara, D. Fohr, Y. Laprie, and K. Smaïli. *Reconnaissance automatique de la parole-Du signal à son interprétation : Du signal à son interprétation.* Dunod-disponible à UHBC 10718/1 Electronique, 2006.

D. O. Hebb. *The organization of behavior.* Science Edition,, 1961.

G. Heigold, W. Macherey, R. Schlüter, and H. Ney. Minimum exact word error training. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 186–190. IEEE, 2005.

H. Hermansky and L. A. Cox Jr. Perceptual linear predictive (plp) analysis-resynthesis technique. In *Applications of Signal Processing to Audio and Acoustics, 1991. Final Program and Paper Summaries., 1991 IEEE ASSP Workshop on*, pages 0_37–0_38. IEEE, 1991.

H. Hermansky and N. Morgan. Rasta processing of speech. *Speech and Audio Processing, IEEE Transactions on*, 2(4) :578–589, 1994.

T. Hézard. *Production de la voix : exploration, modèles et analyse/synthèse*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2013.

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

G. E. Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5) :359–366, 1989.

X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The sphinx-ii speech recognition system : an overview. *Computer Speech & Language*, 7(2) :137–148, 1993.

D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, A. Rudnicky, et al. Pocketsphinx : A free, real-time continuous speech recognition system for hand-held devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE, 2006.

J.-w. Hung, H.-m. Wang, and L.-s. Lee. Comparative analysis for data-driven temporal filters obtained via principal component analysis (pca) and linear discriminant analysis (lda) in speech recognition. In *INTERSPEECH*, pages 1959–1962, 2001.

I. Illina. Contributions à la reconnaissance robuste de la parole. *Habilitation à diriger des recherches, Université de Nancy*, 2, 2005.

I. Illina, D. Fohr, and D. Jouvet. Grapheme-to-phoneme conversion using conditional random fields. In *12th Annual Conference of the International Speech Communication Association-Interspeech 2011*, 2011.

H. Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach.* GMD-Forschungszentrum Informationstechnik, 2002.

F. Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13(6) :675–685, 1969.

F. Jelinek. Speech recognition by statistical methods. *Proceedings of the IEEE*, 64 : 532–556, 1976.

F. Jelinek. Interpolated estimation of markov source parameters from sparse data. *Pattern recognition in practice*, 1980.

F. Jelinek. *Statistical methods for speech recognition.* MIT press, 1997.

H. Jiang. Confidence measures for speech recognition : A survey. *Speech communication*, 45(4) :455–470, 2005.

D. Jouvet. *Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques.* PhD thesis, ENST, 1988.

D. Jouvet and D. Fohr. Combining forward-based and backward-based decoders for improved speech recognition performance. In *InterSpeech-14th Annual Conference of the International Speech Communication Association-2013*, 2013.

D. Jouvet and D. Fohr. About combining forward and backward-based decoders for selecting data for unsupervised training of acoustic models. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, 2014.

D. Jouvet, D. Fohr, and I. Illina. Evaluating grapheme-to-phoneme converters in automatic speech recognition context. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4821–4824. IEEE, 2012a.

D. Jouvet, A. Gorin, and N. Vinuesa. Exploitation d'une marge de tolérance de classification pour améliorer l'apprentissage de modèles acoustiques de classes en reconnaissance de la parole. In *Journée d'Etude sur la Parole-TALN-RECITAL 2012*, pages 763–770, Grenoble, France, June 2012b. URL `https://hal.inria.fr/hal-00753394`.

B.-H. Juang and S. Katagiri. Discriminative learning for minimum error classification [pattern recognition]. *Signal Processing, IEEE Transactions on*, 40(12) :3043–3054, 1992.

W. Jun. An overview of automatic speaker diarization systems. 2012.

S. Kapadia. *Discriminative training of hidden Markov models.* PhD thesis, Citeseer, 1998.

M. Karafiát, F. Grézl, P. Schwarz, L. Burget, et al. Robust heteroscedastic linear discriminant analysis and lcrc posterior features in large vocabulary continuous speech recognition.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(3) :400–401, 1987.

D. Klakow. Selecting articles from the language model training corpus. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1695–1698. IEEE, 2000.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE, 1995.

P. Koehn and B. Haddow. Towards effective use of training data in statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 317–321. Association for Computational Linguistics, 2012.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses : Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

R. Kuhn and R. De Mori. A cache-based natural language model for speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 12(6) :570–583, 1990.

N. Kumar and A. G. Andreou. *Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition*. PhD thesis, Johns Hopkins University, 1997.

R. Lacouture. *Au sujet des algorithmes de recherche des systèmes de reconnaissance de la parole à grands vocabulaires*. PhD thesis, Université McGill, 1995.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

A. Laurent. *Auto-adaptation et reconnaissance automatique de la parole*. PhD thesis, Université du Maine, 2010.

H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon. Structured output layer neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5524–5527. IEEE, 2011.

T. LE Manh. Analyse acoustique de sons bien identifiés par un systeme de reconnaissance automatique de la parole. 2007.

Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10) :1995, 1995.

A. Lee, T. Kawahara, and K. Shikano. Julius—an open source real-time large vocabulary recognition engine. 2001.

C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2) :171–185, 1995.

M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–670. ACM, 2014.

H. Lin and J. Bilmes. How to select a good training-data subset for transcription : Submodular active selection for sequences. Technical report, DTIC Document, 2009.

D. Lindley. Book review of the third edition of jeffreys' theory of probability. *Journal of the American Statistical Association*, 57 :922–924, 1962.

F.-H. Liu, R. M. Stern, X. Huang, and A. Acero. Efficient cepstral normalization for robust speech recognition. In *Proceedings of the workshop on Human Language Technology*, pages 69–74. Association for Computational Linguistics, 1993.

X. Liu, M. J. F. Gales, and P. C. Woodland. Use of contexts in language model interpolation and adaptation. *Computer Speech & Language*, 27(1) :301–321, 2013.

A. Ljolje, M. D. Riley, and D. Hindle. The at&t large vocabulary conversational speech recognition system. In *EUROSPEECH*, 1999.

B. Lowerre. The harpy speech understanding system. In *Readings in speech recognition*, pages 576–586. Morgan Kaufmann Publishers Inc., 1990.

N. Madnani. Querying and serving n-gram language models with python. *The Python Papers*, 4(2), 2009.

P. B. d. Mareüil, G. Adda, M. Adda-Decker, C. Barras, B. Habert, and P. Paroubek. Une étude quantitative des marqueurs discursifs, disfluences et chevauchements de parole dans des interviews politiques. *TIPA. Travaux interdisciplinaires sur la parole et le langage*, (29), 2013.

J. Mariani. *Reconnaissance de la parole*. Hermès science, 2002.

J. D. Markel and A. H. Gray. *Linear prediction of speech*. Communication and cybernetics. Springler-Verlag, Berlin, Heidelberg, New York, 1976. ISBN 0-387-07563-1. URL `http://opac.inria.fr/record=b1079660`.

A. Martinet. *La double articulation du langage, dans Éléments de linguistique générale*. linguistic. A. Colin, 1960. URL `https://books.com`.

A. Martinet and A. Colin. *Eléments de linguistique générale*. A. Colin, 1967.

S. Matsoukas, J.-L. Gauvain, G. Adda, T. Colthurst, C.-L. Kao, O. Kimball, L. Lamel, F. Lefevre, J. Z. Ma, J. Makhoul, et al. Advances in transcription of broadcast news and conversational telephone speech within the combined ears bbn/limsi system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5) :1541–1556, 2006.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, 1943.

D. Mercier. *Le livre des techniques du son - Tome 1 - 4e édition - Notions fondamentales*. Audio-Photo-Vidéo. Dunod, version 1 disponible à UHBC 02-06-052, 2010. ISBN 9782100560851. URL `https://books.google.dz/books?id=I0W5EY3_CuMC`.

C. Meunier. Phonétique acoustique. *Les dysarthries*, pages 164–173, 2007.

F. Mezzoudj. *Optimisation des machines à vecteurs de support multiclasses par des méthaheuristiques (SVM-AG)*. Thèse de magister, USTO, Algèrie, 2011.

F. Mezzoudj and A. Benyettou. On the optimization of multiclass support vector machines dedicated to speech recognition. In T. Huang, Z. Zeng, C. Li, and C. Leung, editors, *Neural Information Processing*, volume 7664 of *Lecture Notes in Computer Science*, pages 1–8. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-34480-0. doi : 10.1007/978-3-642-34481-7_1. URL `http://dx.doi.org/10.1007/978-3-642-34481-7_1`.

F. Mezzoudj, D. Langlois, D. Jouvet, and A. Benyettou. Textual data selection for language modelling in the scope of automatic speech recognition. In *International Conference on Natural Language and Speech Processing, Algeria*, 2015a.

F. Mezzoudj, M. Loukam, and A. Benyettou. On an empirical study of smoothing techniques for a tiny language model. In *Proceedings of IPAC 15 November 23-25, Batna, Algeria*, pages 67–80. ACM, 2015b. URL `http://dx.doi.org/10.1145/2816839.2816878`.

T. Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 2012.

T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky. Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201, 2011.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013a.

T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751, 2013b.

A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM, 2007.

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv :1206.6426*, 2012.

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

M. Mohri and M. Riley. A weight pushing algorithm for large vocabulary speech recognition. In *INTERSPEECH*, pages 1603–1606, 2001.

R. C. Moore and W. Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224. Association for Computational Linguistics, 2010.

A. Nádas, D. Nahamoo, and M. A. Picheny. On a model-robust training method for speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36 (9) :1432–1436, 1988.

R. Nemoto, M. Adda-Decker, and I. Vasilescu. Fouille de données audio pour la classification automatique de mots homophones. *EGC'2008*, 2008.

H. Ney and U. Essen. On smoothing techniques for bigram-based natural language modelling. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 825–828. IEEE, 1991.

H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1) :1–38, 1994.

N. Obin. *Apprentissage de la correlation de la F0et de l'enveloppe spectrale : application à la transcription de la voix parlée.* 2006.

S. Oger. Modèles de langage ad hoc pour la reconnaissance automatique de la parole. Avignon, 2011.

I. Oparin, O. Glembek, L. Burget, and J. Cernocky. Morphological random forests for language modeling of inflectional languages. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pages 189–192. IEEE, 2008.

B. Pinkowski. Principal component analysis of speech spectrogram images. *Pattern recognition*, 30(5) :777–787, 1997.

D. Povey and P. C. Woodland. Minimum phone error and i-smoothing for improved discriminative training. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–105. IEEE, 2002.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

G. Pérennou. Le projet bdlex de base de données lexicales et phonologiques. In *1ères journeés du GRECO-PRC CHM, EC2 e d., Paris*, 1998.

L. Rabiner and B.-H. Juang. Historical perspective of the field of asr/nlu. In *Springer Handbook of Speech Processing*, pages 521–538. Springer, 2008.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.

L. R. Rabiner and B. Juang. Statistical methods for the recognition and understanding of speech. *Encyclopedia of language and linguistics*, 2004.

M. G. Rahim, C.-H. Lee, and B.-H. Juang. Discriminative utterance verification for connected digits recognition. *IEEE transactions on speech and audio processing*, 5(3) : 266–277, 1997.

M. Ravishankar, R. Singh, B. Raj, and R. M. Stern. The 1999 cmu 10x real time broadcast news transcription system. In *Proc. darpa workshop on automatic transcription of broadcast news*. Citeseer, 2000.

J. Razik. *Mesure de confiance trame-synchrones et locales en reconnaissance automatique de la parole*. PhD thesis, Université Henri Poincaré-Nancy I, 2007.

G. Rigoll. *Discriminative Training and Acoustic Modeling for Automatic Speech Recognition*. PhD thesis, Citeseer, 2010.

B. Roark, M. Saraclar, and M. Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2) :373–392, 2007.

F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386, 1958.

R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. 1996.

R. Rosenfeld. Two decades of statistical language modeling : Where do we go from here ? 2000.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*, 2016.

D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney. The rwth aachen university open source speech recognition system. In *Interspeech*, pages 2111–2114, 2009.

T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for lvcsr. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8614–8618. IEEE, 2013.

F. Sajous, N. Hathout, and B. Calderone. Glàff, un gros lexique à tout faire du français. In *Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013)*, pages 285–298, 2013.

F. Sajous, N. Hathout, and B. Calderone. Ne jetons pas le wiktionnaire avec l'oripeau du web ! etudes et réalisations fondées sur le dictionnaire collaboratif. In *4ème Congrès Mondial de Linguistique Française (CMLF 2014)*, pages 663–680, 2014.

G. Salvi and N. Vanhainen. The wavesurfer automatic speech recognition plugin. In *LREC*, pages 3067–3071, 2014.

G. Saon, H. Soltau, D. Nahamoo, and M. Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *ASRU*, pages 55–59, 2013.

T. Schultz. Globalphone : a multilingual speech and text database developed at karlsruhe university. In *INTERSPEECH*, 2002.

R. Schwartz and S. Austin. A comparison of several approximate algorithms for finding multiple (n-best) sentence hypotheses. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 701–704. IEEE, 1991.

H. Schwenk. Continuous space language models. *Computer Speech & Language*, 21(3) : 492–518, 2007.

H. Schwenk. Cslm : a modular open-source continuous space language modeling toolkit. In *INTERSPEECH*, 2013.

H. Schwenk, A. Rousseau, and M. Attik. Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proceedings of the NAACL-HLT 2012 Workshop : Will We Ever Really Replace the N-gram Model ? On the Future of Language Modeling for HLT*, pages 11–19. Association for Computational Linguistics, 2012.

F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Interspeech*, pages 437–440, 2011.

M. S. Seigel. *Confidence Estimation for Automatic Speech Recognition Hypotheses*. PhD thesis, Ph. D. thesis, University of Cambridge, 2013.

S. Seng, S. Sam, V.-B. Le, B. Bigi, and L. Besacier. Reconnaissance automatique de la parole en langue khmère : quelles unités pour la modélisation du langage et la modélisation acoustique. 2010.

M. A. B. Shaik, D. Rybach, S. Hahn, R. Schlüter, and H. Ney. Hierarchical hybrid language models for open vocabulary continuous speech recognition using wfst. *Proc. of SAPA*, 2012.

C. E. Shannon. The mathematical theory of communication. 1948.

I. Sheikh, I. Illina, D. Fohr, and G. Linarès. Oov proper name retrieval using topic and lexical context models. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5291–5295. IEEE, 2015.

X. Shen and B. Xu. The study of the effect of training set on statistical language modeling. In *INTERSPEECH*, pages 721–724, 2001.

S. S. K. Shinoda, M. Nakai, and H. Shimodaira. Analytic methods for acoustic model adaptation : A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

O. Siohan. On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 125–128. IEEE, 1995.

O. Siohan and M. Bacchiani. ivector-based acoustic data selection. In *INTERSPEECH*, pages 657–661, 2013.

K. Smaïli. *Conception et réalisation d'une machine à dictée a entrée vocale destinée ux grands vocabulaires : le système MAUD texte imprimé.* PhD thesis, Nancy 1, 1991. Thèse Doctorat : Sciences appliquées.

R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

A. Stolcke. Srilm : an extensible language modeling toolkit. In U. Densver, editor, *Proceeding of the ICSLP*, volume 2, pages 109–904, 2002.

A. Stolcke, J. Zheng, W. Wang, and V. Abrash. Srilm at sixteen : Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, page 5, 2011.

H. Strik and C. Cucchiarini. Modeling pronunciation variation for asr : A survey of the literature. *Speech Communication*, 29(2) :225–246, 1999.

M. Sundermeyer, R. Schlüter, and H. Ney. On the estimation of discount parameters for language model smoothing. In *INTERSPEECH*, pages 1433–1436, 2011.

A. R. Syed. Optimizing data selection for automatic speech recognition in low resource languages. 2015.

H. Tang, J. Keshet, and K. Livescu. Discriminative pronunciation modeling : A large-margin, feature-rich approach. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*, pages 194–203. Association for Computational Linguistics, 2012.

K. Tangigaki, H. Yamamoto, and Y. Sagisaka. A hierarchical language model incorporating class-dependent word models for oov words recognition. In *the Proceedings of the 6th International Conference on Spoken Language Processing*, 2000.

I. Tellier. Introduction au taln et à l'ingénierie linguistique, univ. lile3, 2008. URL `http://www.grappa.univ-lille3.fr/polys/info_ling/index.html`.

E. Thelen, X. Aubert, and P. Beyerlein. Speaker adaptation in the philips system for large vocabulary continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1035–1038. IEEE, 1997.

N. Tomashenko, Y. Khokhlov, A. Larcher, and Y. Estève. Exploration de paramètres acoustiques dérivés de gmm pour l'adaptation non supervisée de modèles acoustiques à base de réseaux de neurones profonds. *Proceedings of the 31éme Journées d'Études sur la Parole (JEP)*, 2016.

A. Toral. Hybrid selection of language model training data using linguistic information and perplexity. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation, Sofia, Bulgaria*, pages 8–12, 2013.

V. Valtchev, J. Odell, P. C. Woodland, and S. J. Young. Mmie training of large vocabulary recognition systems. *Speech Communication*, 22(4) :303–314, 1997.

D. Vaufreydaz. *Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue*. PhD thesis, Grenoble 1, 2002a.

D. Vaufreydaz. Modélisation statistique du langage à partir d'internet pour la reconnaissance automatique de la parole continue. 2002b.

K. Vertanen. An overview of discriminative training for speech recognition. *University of Cambridge*, 2004.

B. Virole. *Phonétique acoustique appliquée en audioprothèse*, 1999.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2) :260–269, 1967.

W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4 : A flexible open source framework for speech recognition. 2004.

K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3311–3315. IEEE, 2014a.

K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes. Unsupervised submodular subset selection for speech data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4107–4111. IEEE, 2014b.

P. J. Werbos. Backpropagation through time : what it does and how to do it. *Proceedings of the IEEE*, 78(10) :1550–1560, 1990.

F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13 (1) :23–31, 2005.

B. Widrow and M. E. Hoff. Adaptive switching circuits. In *Neurocomputing : foundations of research*, pages 123–134. MIT Press, 1988.

B. Widrow and S. D. Stearns. Adaptive signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, 1, 1985.

B. Widrow, M. E. Hoff, et al. Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pages 96–104. New York, 1960.

I. H. Witten and T. C. Bell. The zero-frequency problem : Estimating the probabilities of novel events in adaptive text compression. *Ieee transactions on information theory*, 37(4) :1085–1094, 1991.

Y. Wu, R. Zhang, and A. Rudnicky. Data selection for speech recognition. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 562–565. IEEE, 2007.

X. Xiao, E. S. Chng, and H. Li. On the study of very low-resource language keyword search. In *Proceedings of APSIPA Annual Summit and Conference*, volume 16, 2015.

J.-W. Xu, D. Erdogmus, and J. C. Principe. Minimum error entropy luenberger observer. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1923–1928. IEEE, 2005.

P. Xu and F. Jelinek. Random forests in language modeling. In *Proceedings of EMNLP*, volume 4, pages 325–332, 2004.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. *The HTK book*, volume 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. The htk book (revised for htk version 3.4. 1). *Cambridge University*, 2009.

S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pages 307–312. Association for Computational Linguistics, 1994.

F. Yvon. Sub-word based language modeling of morphologically rich languages for lvcsr.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM, 2001.

P. Zhan and A. Waibel. Vocal tract length normalization for large vocabulary continuous speech recognition. Technical report, DTIC Document, 1997.

X. Zhu, C. Barras, L. Lamel, and J.-L. Gauvain. Speaker diarization : From broadcast news to lectures. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 396–406. Springer, 2006.

A. Zolnay and U. D.-I. R. Haeb-Umbach. *Acoustic feature combination for speech recognition.* PhD thesis, RWTH Aachen University, 2006.