Construction rapide, performante et mutualisée de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues

Kévin Vythelingum

THÈSE DE DOCTORAT DE

LE MANS UNIVERSITÉ
COMUE **U**NIVERSITÉ **B**RETAGNE **L**OIRE

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

Par

Kévin VYTHELINGUM

« Construction rapide, performante et mutualisée de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues »

Thèse présentée et soutenue à LE MANS , le 10 décembre 2019
Unité de recherche : LIUM
Thèse N° : 2019LEMA1035

Composition du jury :

Rapporteurs :	**Martine ADDA-DECKER**, Directrice de recherche, LPP, Université Paris 3 Sorbonne
	Denis JOUVET, Directeur de recherche, LORIA, INRIA Nancy
Examinateurs :	**Sylvain Meigner**, Professeur, LIUM, Le Mans Université
	Jean-François BONASTRE, Professeur, LIA, Avignon Université
	Damien LOLIVE, Maître de Conférences, IRISA, Université de Rennes 1
Directeur de thèse :	**Yannick ESTEVE**, Professeur, LIA, Avignon Université
Co-encadrants de thèse :	**Olivier ROSEC**, Directeur Recherche & Développement, Voxygen
	Antony LARCHER, Maître de Conférences, LIUM, Le Mans Université

Résumé

Nous étudions dans cette thèse la construction mutualisée de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues, avec un objectif de performance et de rapidité de développement. Le développement rapide des technologies vocales pour de nouvelles langues anime des ambitions scientifiques et est aujourd'hui considéré comme stratégique par les acteurs industriels. Cependant, le développement des langues est conduit de manière morcelée par quelques centres de recherche travaillant chacun sur un nombre réduit de langues. Or, ces technologies partagent de nombreux points communs. Notre étude se concentre sur la construction et la mutualisation d'outils pour la création de lexiques, l'apprentissage de règles de phonétisation et l'exploitation de données imparfaites. Nos contributions portent sur la sélection de données pertinentes pour l'apprentissage de modèles acoustiques, le développement conjoint de phonétiseurs et de lexiques de prononciations pour la reconnaissance et la synthèse de la parole, et l'exploitation de modèles neuronaux pour la transcription phonétique à partir du texte et du signal de parole. De plus, nous présentons une approche de détection automatique des erreurs de transcriptions phonétiques dans les bases de données annotées de signal de parole. Cette étude a montré qu'il était possible de réduire de manière importante la quantité de données à annoter manuellement lors du développement de nouveaux systèmes de synthèse de la parole. Cela contribue naturellement à réduire le temps de collecte de données pour la création de nouveaux systèmes. Finalement, nous étudions un cas applicatif en construisant de façon mutualisée un système de reconnaissance et de synthèse de la parole pour une nouvelle langue.

Abstract

We study in this thesis the joint construction of speech recognition and synthesis systems for new languages, with the goals of accuracy and quick development. The rapid development of voice technologies for new languages is driving scientific ambitions and is now considered strategic by industial players. However, language development research is led by a few research centers, each working on a limited number of languages. However, these technologies share many common points. Our study focuses on building and sharing tools between systems for creating lexicons, learning phonetic rules and taking advantage of imperfect data. Our contributions focus on the selection of relevant data for learning acoustic models, the joint development of phonetizers and pronunciation lexicons for speech recognition and synthesis, and the use of neural models for phonetic transcription from text and speech signal. In addition, we present an approach for automatic detection of phonetic transcript errors in annotated speech signal databases. This study has shown that it is possible to significantly reduce the quantity of data annotation useful for the development of new text-to-speech systems. It naturally helps to reduce data collection time in the process of new systems creation. Finally, we study an application case by jointly building a system for recognizing and synthesizing speech for a new language.

Remerciements

Je remercie chaleureusement toutes les personnes qui m'ont aidé durant ma thèse et qui ont participé, de près ou de loin, à cette tranche de vie. Sachez que sans les nombreux échanges que j'ai pu avoir, ce projet n'aurait pas pu aboutir.

Un grand merci à mon directeur de thèse, Yannick Estève, pour ses conseils judicieux et ses idées originales. Nos discussions m'ont beaucoup appris et m'ont aidé à prendre du recul à tous les niveaux. De plus, son rôle fédérateur m'a permis de travailler en équipe sur de nombreux projets innovants. Mes remerciements vont également à mon encadrant chez Voxygen, Olivier Rosec, pour sa clairvoyance, sa grande disponibilité et ses encouragements. Ses propositions se sont toujours révélées intéressantes et il m'a guidé dès le début dans des directions prometteuses. Je remercie également mon encadrant Anthony Larcher, qui m'a apporté son expérience à des moments cruciaux pour la réussite de ce travail de thèse. J'ai beaucoup appris avec eux et je mesure la chance que j'ai eu d'avoir pu travailler dans ces conditions.

Je tiens aussi à remercier Martine Adda-Decker, directrice de recherche au LPP à l'Université Paris 3 Sorbonne, et Denis Jouvet, directeur de recherche au LORIA à l'INRIA Nancy, pour avoir accepté d'être les rapporteurs de ce travail. Merci à Sylvain Meigner, professeur au LIUM à l'Université du Mans, Jean-François Bonastre, professeur au LIA à l'Université d'Avignon et Damien Lolive, maître de conférences à l'IRISA à l'Université de Rennes 1, qui ont accepté d'être membres du jury de thèse. Je remercie également Laurent Besacier, professeur au LIG à l'Université Grenoble Alpes, qui a accompagné Jean-François Bonastre lors des comités de suivi individuel qui ont ponctué ces années de thèse. Merci à tous pour l'intérêt porté à mes travaux.

Je remercie l'ensemble des personnes que j'ai côtoyé chez Voxygen durant cette thèse, pour leur formidable énergie positive et leur compréhension pendant la rédaction du manuscrit : Paul Bagshaw, Sofiane Baloul, Louise Boucheseche, Laure Charonnat, Elizaveta Clouet, Gérard Compain, Guillaume Cueffel, Laurence Duday, Pierre-Yves Jolivet, Marion Le Dissez, Chiara Mazza, Brigitte O'Rorke, Éric Renaud, Romain Saillard, Christian Sassady et Philippe Vinci. C'est un vrai plaisir de travailler avec eux.

Je n'oublie pas l'ensemble des personnes que j'ai rencontré au LIUM, à qui j'adresse mes plus vifs remerciements pour leur aide précieuse et leur bonne humeur. Je pense particulièrement à Rajoua Anane, Walid Aransa, Adrien Bardet, Amira Barhoumi, Loïc Barrault, Emmanuelle Billard, Abdessalam Bouchekif, Fethi Bougares, Pierre-Alexandre Broux, Ozan Caglayan, Nathalie Camelin, Antoine Caubrière, Paul Deléglise, Nicolas Dugué, Grégor Dupuy, Anne Cécile Erreau, Mercedes García Martínez, Sahar Ghannay, Mélanie Hardy, Bruno Jacob, Vincent Jousse, Malik Koné, Carole Lailler, Antoine Laurent, Daniel Luzzati, Salima Mdhaffar, Sylvain Meignier, Étienne Micoulaut, Simon Petitrenaud, Manon Pinel, Dominique Py, Anthony Rousseau, Edwin Simonnet, Marie Tahon, Natalia Tomashenko, Jane Wottawa. Ce sont beaucoup de bons moments partagés qui ont ponctué cette thèse. Je tiens également à souligner le formidable cadre de travail que le LIUM offre aux doctorants. C'est réellement une chance de pouvoir réaliser une thèse dans un environnement de cette qualité.

Merci enfin à ma famille et mes amis pour leur soutien sans faille et leur gentillesse. Un remerciement particulier à Pauline qui a toujours été à mes côtés ces dernières années. Merci pour ta patience et pour toutes les concessions que tu as consenties à faire.

Merci à tous.

Table des matières

Table des figures

Liste des tableaux

Acronymes

CMLLR Constrained Maximum Likelihood Linear Regression. 44

CNN Réseau de Neurones Convolutif (Convolutional Neural Network). 25–29, 31, 35, 44

CTC Connectionist Temporal Classification. 35–37, 45, 87

DCT Transformée en Cosinus Discrète (Discrete Cosine Transform). 12

DNN Réseau de Neurones Profond (Deep Neural Network). 21, 29, 31, 44, 55

fMLLR feature-space Maximum Likelihood Linear Regression. 41, 44, 72

fMPE feature-space Minimum Phone Error. 41

GMM Modèles de Mélanges Gaussiens (Gaussian Mixture Model). 29–31, 43, 44, 75

GRU Gated Recurrent Unit. 25, 84

HMM Modèles de Markov Cachés (Hidden Markov Model). 24, 29–32, 43, 44, 55, 56

LSTM Long Short-Term Memory. 25

MAP Maximum A Posteriori Estimation. 44

MCR Manual Checking Rate. 90

MER Matching Error Rate. 67

MFCC Mel-Frequency Cepstral Coefficient. 41, 54

MLLR Maximum Likelihood Linear Regression. 44

MLP Perceptron multicouches (Multi Layer Perceptron). 22–27, 44, 46

PER Phone Error Rate. 82, 84

PLP Perceptual Linear Prediction. 41, 72

RNN Réseaux de Neurones Récurrents (Recurrent Neural Network). 24, 25, 28, 31, 33, 35, 36, 46

SRAP Système de Reconnaissance Automatique de la Parole. xiii, 40, 41, 44–47, 64–68, 71–74, 76–78

TDNN Time Delay Neural Network. 44, 75

WER Word Error Rate. 67, 72

Introduction

LES technologies vocales jouent un rôle de plus en plus important dans la société actuelle, et de nouveaux usages sont sans cesse imaginés. Cependant, leur déploiement est restreint à certaines zones géographiques, dans la mesure où elles ne sont développées que pour certaines langues. En effet, le développement des technologies vocales s'est fait de manière morcelée par quelques centres de recherche travaillant chacun sur un nombre réduit de langues. La problématique du développement de langues – plus précisément la rationalisation et l'accélération du processus de développement de langues – ne s'est in fine posée que relativement tardivement. Parmi ces technologies, la reconnaissance et la synthèse de la parole sont particulièrement touchées par ces inégalités de développement. Cette problématique de développement rapide de langues est maintenant considérée comme stratégique par les acteurs industriels, l'enrichissement rapide d'un catalogue de langues contribuant naturellement à ouvrir de nouveaux marchés.

Il existe actuellement plus de 7000 langues (ou dialectes) parlées dans le monde, dont environ 330 sont chacune parlées par plus d'un million de personnes. Parmi toutes ces langues, seules quelques-unes ont été effectivement étudiées pour la reconnaissance ou la synthèse de la parole, et même parmi celles-ci les performances des systèmes peuvent être très variables d'une langue à l'autre en fonction de la quantité et de la qualité des ressources sonores et linguistiques disponibles. De plus, ces ressources linguistiques ont un coût non négligeable : même pour des langues européennes bien étudiées (par exemple l'allemand, l'italien ou l'espagnol) le coût des données accessibles sur catalogue peut être rédhibitoire pour un usage industriel. Ainsi, de nombreuses applications liées aux technologies vocales ne peuvent pas voir le jour ou ne peuvent pas s'adapter aux populations qu'elles visent.

Les développements de langues sont actuellement conduits séparément pour la reconnaissance et la synthèse de parole. Or, ces technologies partagent beaucoup de points communs. Des ressources linguistiques proches (lexiques, données textuelles annotées) pourraient par exemple être mises en commun. En outre, le développement d'un phonétiseur pourrait être entrepris de façon conjointe, même si la phonétisation revêt des objectifs différents pour la reconnaissance et la synthèse vocale. Il semble donc possible de mutualiser les développements de langues pour la reconnaissance et synthèse de la parole. De plus, nous observons ces dernières années un regain d'intérêt pour les modèles neuronaux, notamment dans les domaines de traitement de la parole. Ceci s'explique par les progrès algorithmiques réalisés, la puissance de calcul actuelle avec l'utilisation de processeurs graphiques, et des quantités de données exploitables conséquentes grâce au Web. L'utilisation de ces modèles a largement permis de faire progresser les systèmes vocaux, bien que leur potentiel ne soit pas encore totalement exploité. En effet, ils permettent plus facilement le partage de données d'apprentissage car des systèmes conçus pour des tâches différentes

peuvent avoir des représentations numériques communes. Aussi, les modèles neuronaux simplifient les chaînes de traitements et réduisent les connaissances linguistiques nécessaires en favorisant l'apprentissage à partir de données, ce qui accélère naturellement le développement de nouveaux systèmes.

C'est dans ce contexte que l'entreprise Voxygen et le Laboratoire d'Informatique de l'Université du Mans ont collaboré à travers cette thèse pour explorer de nouvelles voies de recherche et apporter de nouvelles solutions pour la construction rapide, performante et mutualisée de systèmes multilingues pour la reconnaissance et la synthèse de la parole. La cible principale de ce travail de thèse concerne la définition, le développement et la mise en œuvre d'un processus de développement de langues pour la reconnaissance et la synthèse de la parole. L'objectif est notamment de mutualiser au maximum les développements et ressources nécessaires à la création d'une nouvelle langue. Sur le plan de la linguistique, l'objectif est de concevoir un ensemble d'outils pour la création de lexiques et l'apprentissage de systèmes de phonétisation. Sur le plan acoustique, la thèse vise à améliorer le processus d'apprentissage de modèles acoustiques en exploitant des données imparfaites.

Ce document est organisé en deux parties. Dans la première, nous présentons un état de l'art de la reconnaissance et de la synthèse de la parole, avec les techniques de modélisation qui leur sont propres. L'objet de la deuxième partie est de développer le travail réalisé durant cette thèse, à savoir la construction d'un système de reconnaissance de la parole pour la langue arabe à partir de données imparfaites, la transcription phonétique à partir du texte et du signal de parole ainsi que la détection automatique d'erreurs de phonétisation.

Nous décrivons dans le premier chapitre notre objet d'étude, la parole, dans sa dimension acoustique et linguistique. Nous montrons alors qu'elle peut être représentée par des séquences de valeurs numériques et symboliques.

Dans le deuxième chapitre, nous passons en revue différents algorithmes permettant de modéliser des séquences. Ainsi, nous posons une base théorique pour aborder la construction de systèmes de reconnaissance et de synthèse de la parole. Nous montrons par ailleurs que ces deux tâches ont de nombreuses similitudes au niveau de la modélisation.

Les troisième et quatrième chapitres présentent respectivement un historique de la reconnaissance et de la synthèse de la parole. Les modèles utilisés dans ce travail de thèse sont plus largement discutés, ainsi que les méthodes d'évaluation des systèmes.

Le cinquième chapitre aborde la première partie de nos contributions. Il s'agit de la construction d'un système de reconnaissance automatique de la parole pour la langue arabe sur la base de données imparfaites. Le système développé a été présenté à une campagne d'évaluation restreignant les données d'apprentissage à un corpus d'émissions de télévision. Nous proposons différents types de modélisation acoustique, plusieurs stratégies pour la création d'un lexique phonétisé, et une technique de sélection de données pour l'apprentissage des modèles.

Dans le sixième chapitre, nous proposons une approche de transcription phonétique exploitant à la fois le texte et le signal de parole. Nous proposons également des modèles entièrement neuronaux pour la phonétisation à partir du texte et à partir du signal. Ensuite, nous décrivons une méthode pour détecter automatiquement des

erreurs de phonétisation dans les bases de données utilisées pour la synthèse de parole. Nous montrons alors qu'en utilisant la reconnaissance de la parole, il est possible d'accélérer l'annotation de données nécessaire à la construction de systèmes de synthèse tout en améliorant leur qualité. Nous évaluons notre méthode sur des données en français dans un premier temps, puis nous étendons notre travail à la création de voix pour la synthèse de la parole dans d'autres langues. Nous illustrons avec l'italien et l'arabe comment peuvent être faites de nouvelles voix pour des langues déjà traitées, avant de décrire la création d'une voix pour une nouvelle langue avec l'exemple du turc.

Finalement, le dernier chapitre résume les éléments importants de ce travail de thèse. Il présente également quelques perspectives de recherche liées aux problématiques soulevées dans le document.

Première partie

État de l'art

Chapitre 1

La parole : description acoustique et transcription symbolique

Sommaire

La parole, notre objet d'étude, résulte de l'articulation de sons produits par l'appareil phonatoire humain. Elle est le moyen de communication le plus naturel entre des personnes. Ainsi, elle a pour but de véhiculer un message porteur de sens au moyen d'un signal acoustique. Ce signal sera ensuite perçu par l'appareil auditif, chargé de transmettre le signal au cerveau en vue de son décodage.

Avant de parler de reconnaissance ou de synthèse automatique de la parole, dont l'objet est respectivement de transcrire la parole en mots ou de la générer automatiquement à partir du texte, nous nous attachons à décrire la parole en tant que signal acoustique et comme composante du langage. Tout d'abord, nous décrivons les éléments physiologiques en œuvre dans la production et la perception de la parole. La compréhension du fonctionnement de ces éléments nous guidera ensuite sur la manière de paramétriser le signal de parole pour traiter efficacement les informations qu'il contient. Enfin, nous verrons que la parole est composée d'une séquence structurée de sons élémentaires prononcés avec une intensité, une intonation et un rythme particuliers. Nous montrerons alors comment traduire la parole de manière symbolique en préservant autant que possible ses détails de prononciation.

1.1 Physiologie

1.1.1 Appareil phonatoire

L'appareil phonatoire désigne l'ensemble des éléments anatomiques en jeu dans la production de parole. Chacun des éléments agit de manière complémentaire. Tout d'abord, les poumons créent un flux d'air qui est conduit par la trachée jusqu'au larynx, où les cordes vocales régulent le débit d'air sortant (figure 1.1).

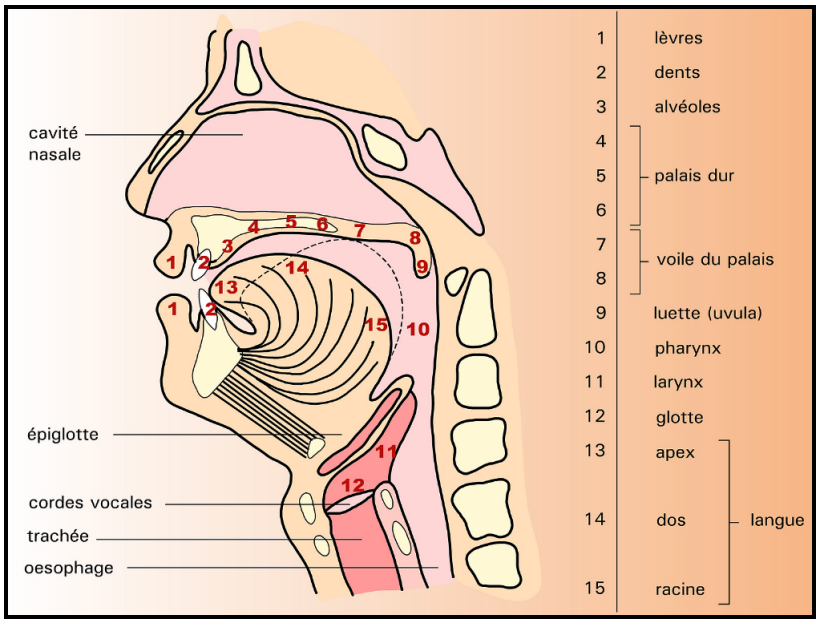

FIGURE 1.1 – Coupe sagittale schématique des organes de la parole [Encyclopaedia Universalis France]

Pour des pressions suffisantes, les cordes vocales peuvent adopter un cycle d'ouverture-fermeture par effet Bernoulli, introduisant un régime vibratoire au flux d'air.

La fréquence de vibration, appelée fréquence fondamentale $F0$, donne la hauteur du son. On dit alors que le signal glottique est périodique, et on parle de parole *voisée*, comme c'est le cas pour les voyelles et certaines consonnes. Dans le cas contraire, lorsque l'air traverse le larynx sans vibrer, les sons produits sont dits *non-voisés*. La taille des cordes vocales étant différente pour chaque individu, la fréquence fondamentale du signal de parole varie d'une personne à l'autre. En particulier, elle varie entre 100 et 150Hz pour les hommes, et entre 140 et 240Hz pour les femmes (CALLIOPE, 1989).

Ensuite, de nombreuses cavités (pharyngeales, buccales, nasales, ...), agissent comme autant de résonnateurs sur le signal glottique. Finalement, l'air sort par la bouche et/ou le nez. On parle respectivement de conduit oral et nasal. Ce dernier présente un ensemble d'articulateurs, comme la langue, la mandibule, les lèvres ou le voile du palais, permettant de modifier la forme et le volume des cavités ainsi que le parcours du flux d'air. L'appareil phonatoire permet donc la production de parole en contrôlant la propagation de l'air dans le conduit vocal.

1.1.2 Appareil auditif

L'appareil auditif permet la perception des sons qui nous entourent. Il est en particulier bien adapté pour percevoir la parole. L'appareil auditif est composé des différentes parties de l'oreille et du système nerveux transmettant l'information perçue au cerveau en vue de son traitement (figure 1.2).

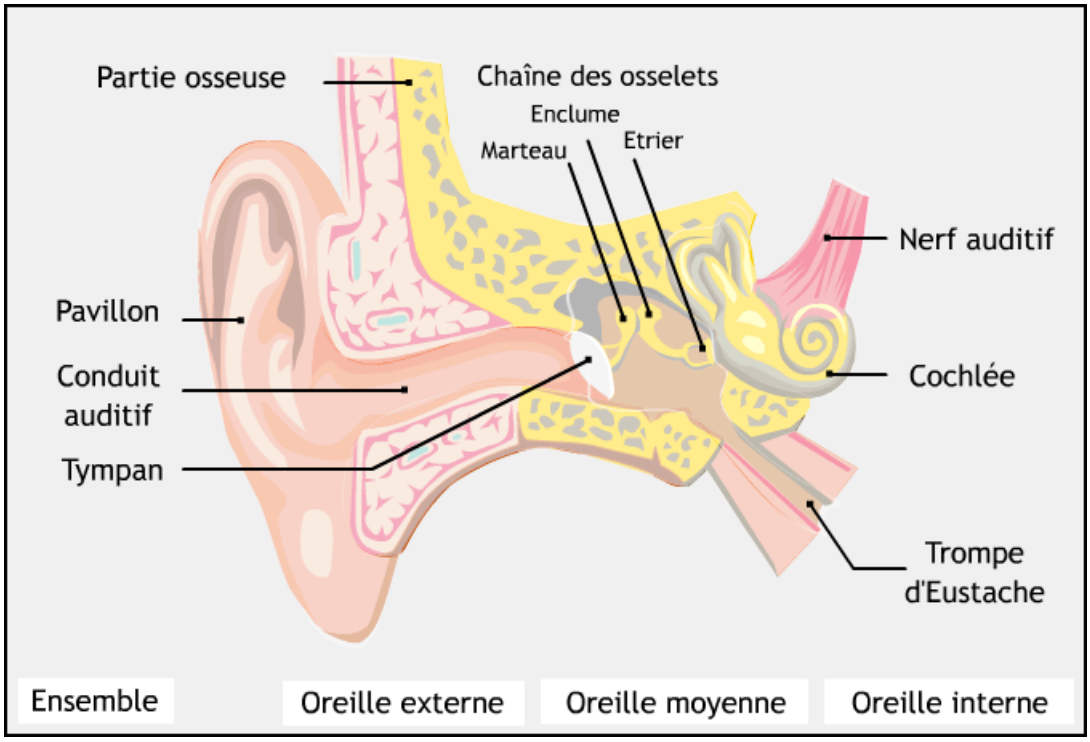

FIGURE 1.2 – Schéma de l'appareil auditif [www.audition.fr]

L'oreille est composée du canal auditif où se propagent les ondes sonores par vibration de l'air jusqu'au tympan. Les vibrations de cette membrane sont transmises

jusqu'à l'oreille interne par les osselets. A partir de la cochlée, les ondes sonores sont converties en influx nerveux pour acheminer le signal jusqu'au cerveau.

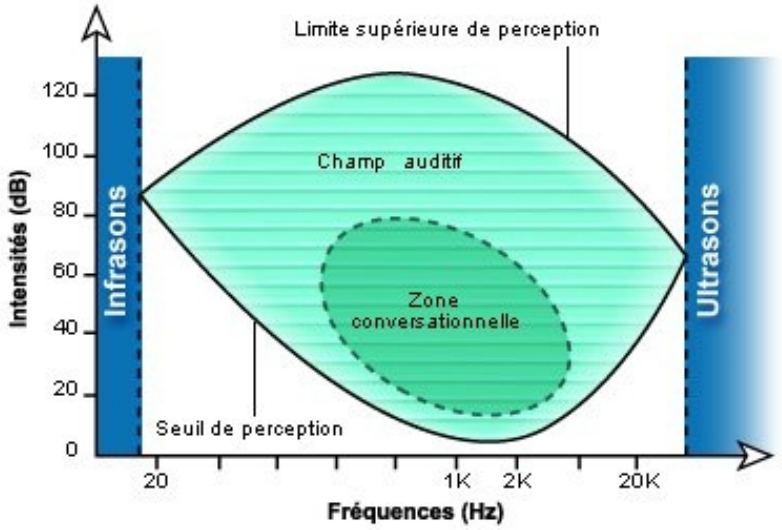

FIGURE 1.3 – Champ auditif humain et zone conversationnelle

Les sons sont perçus entre 20 Hz et 20 kHz pour une intensité comprise entre 0 et 130 dB au delà de laquelle l'appareil auditif subit des dommages irréversibles. Cependant, les sons ne sont pas perçus de manière égale selon leur fréquence et leur intensité : le champ auditif se situe entre un seuil et une limite supérieure de perception non linéaires (figure 1.3). De plus, une analyse spectrale de l'appareil auditif humain montre qu'il existe une zone restreinte où les sons sont plus facilement discriminés en fréquence et en intensité. Cette zone, appelée *zone conversationnelle*, est celle utilisée pour la communication parlée. Ceci nous permet de mieux paramétrer l'analyse du signal de parole, en tenant compte des spécificités de la perception humaine.

1.2 Paramétrisation du signal de parole

La parole est un signal acoustique continu porté par les vibrations de l'air, d'énergie finie et pouvant présenter des périodicités (CALLIOPE, 1989). Lors de son enregistrement, elle est numérisée au format PCM (Pulse Code Modulation) selon une certaine fréquence d'échantillonage. Il est possible de représenter numériquement le signal directement, en représentant chaque échantillon indépendemment, ou en utilisant un modèle paramétrique de type source/filtre en faisant des hypothèses sur la nature acoustique de la parole. Ceci nous permettra d'obtenir une représentation du signal spécifique à la parole grâce aux connaissances acquises sur les appareils phonatoire et auditif.

1.2.1 Représentation directe du signal

La parole est un signal acoustique non stationnaire, mais considéré comme quasi stationnaire pour des intervalles de temps assez courts, de l'ordre de 10 à 30 ms.

Pour décrire l'évolution temporelle du signal de parole, le signal acoustique est ainsi découpé en fenêtres d'analyse de l'ordre de 20 ms, sur lesquels sont appliqués des transformées de Fourier discrètes. Ainsi, on obtient le spectre du signal sonore pour chaque fenêtre d'observation. En pratique, l'algorithme de tranformée de Fourier rapide (FFT, Fast Fourier Transform) est utilisé sur une fenêtre glissante, donnant une information sur des intervalles de temps fixes appelés *trames*, correspondant au pas de déplacement de la fenêtre, soit généralement 10 ms. Pour limiter les effets de bord induits par le découpage arbitraire du signal, une fonction de fenêtrage est appliquée : la fenêtre de Hamming, donnée par l'équation 1.1 pour une fenêtre d'observation $[0, T]$, est couramment utilisée, mais il existe également les fenêtres de Hann, Blackman ou encore Bartlett qui donnent des résultats similaires.

$$w(t) = \begin{cases} 0,54 - 0,46cos(2\pi\frac{t}{T}) & \text{si } t \in [0, T] \\ 0 & \text{sinon} \end{cases} \tag{1.1}$$

Le spectrogramme est un outil permettant de visualiser le spectre d'amplitude du signal de parole en fonction du temps. Il permet en effet de décomposer chaque trame de signal en un ensemble de N_{fft} sinusoïdes. Soit un signal $s(n)$ composé de N échantillons et une fonction de fenêtrage w, le spectre $S_m(k)$ de la trame m de T échantillons est donné par l'équation 1.2, où k est proportionnel à la fréquence du signal :

$$S_m(k) = \sum_{n=0}^{N} s(n)w(n-m)e^{-j\frac{2\pi k}{N_{fft}}n} \tag{1.2}$$

Le spectre d'amplitude est obtenu en prenant le module de l'équation 1.2 pour chaque trame. Autrement dit, $|S_m(k)|$ donne l'amplitude de la sinusoïde de fréquence $2\pi k/N_{fft}$ pour la trame m. En pratique, le temps se trouve en abscisses, les fréquences en ordonnées, et l'amplitude est représentée en niveau de gris (figure 1.4).

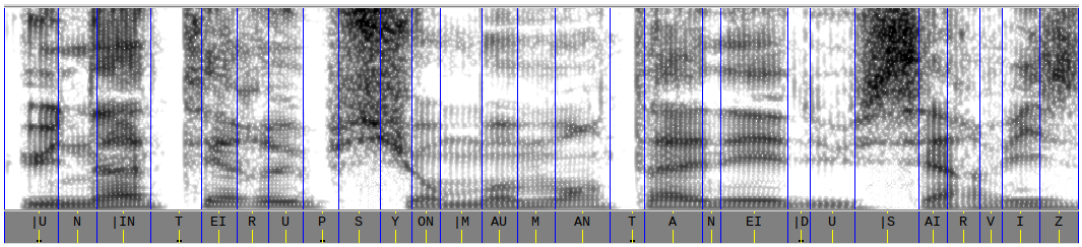

FIGURE 1.4 – Exemple de spectrogramme d'un signal de parole et sa transcription en phonèmes

Des zones plus foncées peuvent être observées dans le spectrogramme. Il s'agit de maxima énergétiques aux fréquences de résonnance du conduit vocal. Ces résonnances, appelées formants, permettent notamment de caractériser la parole dans son timbre : repérer les formants et décrire leur évolution au cours du temps est une première paramétrisation de la parole.

1.2.2 Modélisation source-filtre

Considérant les différents éléments le composant, l'appareil phonatoire peut être modélisé comme la réponse d'un filtre à un signal source : c'est le modèle source-filtre (FANT, 1970). La source correspond alors au signal glottique, et le filtre aux cavités du conduit vocal et aux articulateurs. Le signal de parole s s'écrit ainsi en fonction du temps discrétisé n comme la convolution du signal source e et du filtre h :

$$s(n) = e(n) * h(n) \tag{1.3}$$

L'équation 1.3 peut s'écrire dans le domaine fréquentiel comme :

$$X(f) = E(f) \cdot H(f) \tag{1.4}$$

Le signal source correspond à un bruit auquel s'ajoute une composante périodique dans le cas des sons voisés, qui se matérialise par un spectre de raies harmoniques espacées de la fréquence fondamentale $F0$. Le filtre comprend quant à lui l'enveloppe spectale du signal qui contient l'information utile à la reconnaissance de la parole.

Les travaux cherchant à mieux comprendre la perception de la parole par l'humain montrent que l'appareil auditif n'est pas sensible à toutes les fréquences de façon égale. Pour imiter cette perception, les fréquences composant le spectre d'amplitude sont représentées selon l'échelle de Mel (STEVENS, VOLKMANN et NEWMAN, 1937), qui cherche à placer à égale distance les fréquences perçues comme étant à égale distance. Une manière de convertir la valeur des fréquences d'une échelle linéaire à l'échelle de Mel est donnée par la formule 1.5.

$$F_{Mel}(f) = 2595 \times \log_{10}(1 + \frac{F_{Hz}(f)}{700}) \tag{1.5}$$

Ensuite, le spectre Mel est compressé par une fonction logarithmique pour imiter la perception humaine de la dynamique du signal.Enfin, une Transformée en Cosinus Discrète (Discrete Cosine Transform) (DCT) est utilisée sur le logarithme de $|X_{Mel}(f)|$ pour obtenir le cepstre (relation 1.6).

$$C(s(n)) = DCT(log(|X_{Mel}(f)|)) \tag{1.6}$$

Les coefficients résultant de la DCT sont appelés coefficients MFCC pour *Mel-Frequency Cepstral Coefficient*. Seuls les premiers coefficients sont conservés pour chaque trame car ils sont représentatifs de notre perception du signal. Les dérivées premières et secondes des coefficients sont ajoutées pour tenir compte de la dynamique de l'évolution de ces coefficients.

Ainsi, les connaissances sur la production et la perception de la parole, associées à des notions de traitement du signal, ont permis de transformer un enregistrement de parole en une séquence de vecteurs, dits *vecteurs acoustiques*. Cette représentation du signal à l'avantage de compresser l'information, en ne conservant que des données pertinentes pour l'analyse de la parole. Les séquences de vecteurs acoustiques nous

permettront de construire des modèles pour la reconnaissance et la synthèse de la parole. Il est important de noter que la fréquence fondamentale n'est pas conservée par cette analyse, et qu'elle peut être importante pour la synthèse de la parole ou la reconnaissance dans des langues à tons. Dans ce cas, ce paramètre peut être ajouté aux vecteurs acoustiques décrivant le signal de parole.

1.3 Transcription symbolique de la parole

La parole permet de véhiculer un message, une idée. Transcrire la parole de manière symbolique, c'est donner du sens en tant qu'élément de langage à sa représentation acoustique. Nous cherchons par ce moyen à lier la réalisation acoustique d'un énoncé à la séquence de mots qui le compose. Cependant, la parole n'est pas seulement limitée à une séquence de mots : elle porte, en plus des informations linguistiques (mots, grammaire), des éléments paralinguistiques (sexe et âge du locuteur, émotion, attitude, niveau social, etc.). Tous ces éléments sont porteurs de sens et peuvent modifier l'interprétation du discours. On parle ainsi du *double codage de la parole* (LÉON, 1993), en distinguant le contenu linguistique du contenu phonostylistique. Pour la reconnaissance et la synthèse de la parole, il nous est nécessaire de modéliser la parole dans son ensemble. En effet, il s'agit respectivement de transcrire en mots un énoncé quelle que soit la manière dont il est prononcé, ou de vocaliser avec une prononciation adéquate un message écrit. Nous allons décrire une manière de transcrire la parole de manière symbolique en préservant les éléments de langage qu'elle contient.

1.3.1 Vers une transcription des sons en phonèmes

Les premiers éléments que l'on distingue dans un flux de parole sont les mots. Cependant, pour transcrire la parole avec un grand vocabulaire, il n'est généralement pas possible d'attribuer un symbole unique à chaque mot possible, étant donné la très grande variété de mots. La langue écrite utilise donc un ensemble plus restreint de symboles, et les combinent pour former les mots. Les mots sont alors structurés en séquences selon une grammaire afin de véhiculer une information linguistique. À l'aide de la ponctuation et d'interjections, il est possible d'introduire des éléments d'expressivité. Cependant, la langue écrite reste largement ambiguë : des mots de sens différent et avec une prononciation différente peuvent être écrits de la même manière, et des mots d'orthographes différentes peuvent se prononcer de la même manière. De plus, le nombre de lettres étant généralement plus petit que le nombre de sons possibles, des sons peuvent correspondre à des combinaisons de lettres. Il devient donc nécessaire de dresser l'inventaire des sons distinctifs des langues parlées et de leur associer un symbole unique.

Des tentatives pour établir l'inventaire des sons des langues existantes ont été présentées dès le XVIIème siècle (SCHWEITZER, DODANE et LAZAR, 2018). L'objectif premier était d'obtenir un système d'écriture permettant de transcrire la prononciation de n'importe quelle langue. La principale difficulté est de disposer d'un ensemble de symboles suffisamment grand pour décrire précisément les subtilités du langage parlé, mais assez restreint pour rendre aisées l'écriture et la lecture. L'alphabet figuré de Montmignon (MONTMIGNON, 1785), conçu pour le français, enrichit l'alphabet latin de symboles décrivant les propriétés acoustiques des sons, mais

suppose deux transcriptions pour chaque mot. L'alphabet organique de Charles de Brosses (DE BROSSES, 1765) et le Visible Speech d'Alexander Melville Bell (BELL, 1867) reposent quant à eux sur une description schématique des mouvements articulatoire dans la parole, ce qui engendre un système d'écriture complètement nouveau et difficile à apprendre. En 1877, Henri Sweet améliore le Visible Speech de Bell en s'inspirant des caractères latins pour proposer l'alphabet romique. Il propose notamment d'utiliser deux alphabets, l'un générique à toutes les langues, et l'un spécifique à chaque langue permettant de décrire leurs spécificités phonétiques. Ainsi, il pose les bases de la phonétique, où seules les différences distinctives des sons, qui induisent un changement de sens, sont représentées. Aujourd'hui, l'alphabet phonétique international (API), présenté en 1888 et en constante évolution, hérite de ces réflexions et nous permet d'associer un phonème à chaque unité distinctive des langues existantes. Nous utilisons en pratique la transcription X-SAMPA (1995) de l'API, qui n'utilise que des caractères ASCII en s'affranchissant de la limite d'un seul symbole par son.

Dans l'API (figure 1.5), les consonnes sont décrites selon les articulateurs en œuvre dans leur production, tandis que les voyelles sont représentées sous la forme d'un triangle vocalique. La position des voyelles dans le triangle vocalique est corrélée à la position relative des deux premiers formants (FANT, 1970). Les phonèmes permettent ainsi de décrire la séquence de sons d'un signal de parole dans sa dimension linguistique. Cependant, les informations paralinguistiques comme la durée, l'intensité et l'intonation, ne sont pas représentés. Il est donc nécessaire d'enrichir cette symbolique pour obtenir une représentation plus complète de la parole.

1.3.2 Prosodie

La prosodie est la représentation formelle des éléments expressifs de la parole. Parmi ceux-ci, on retrouve les tons, le rythme, l'accentuation et l'intonation des sons du langage (DI CRISTO, 2000). La prosodie dépend du contexte de chaque élément dans le message oralisé : c'est une caractéristique suprasegmentale. De plus, elle a trait au sens de l'énoncé, selon que ce soit une question, une exclamation, une implication, etc. (DELATTRE, 1966).

Le rôle de la prosodie dans la parole est fondamental. Par exemple, elle permet en français de structurer l'énoncé et en anglais de désambiguïser certains homonymes (FÓNAGY, 2003). Par ailleurs, la prosodie dépend également de l'attitude du locuteur, de son état émotionnel, de son niveau social, de son origine géographique, etc. La prosodie est porteuse de la sémantique d'un message au même titre que la séquence de mots qui le compose.

Concrètement, il est possible d'attribuer à chaque phonème d'un énoncé des éléments quantitatifs pour certains aspects de la prosodie, comme sa durée, sa fréquence fondamentale (F0) s'il est voisé ou son intensité. Cependant, la modélisation conjointe de ces paramètres et leur interpréation est une tâche complexe liée à l'expressivité, qui reste difficile à représenter symboliquement. En pratique, chaque phonème est accompagné d'un ensemble de descripteurs symboliques permettant de conserver au mieux les informations transmises par la parole, à défaut de pouvoir en transcrire le sens. Un vecteur peut donc être associé à chaque phonème, dont les valeurs sont déterminées en fonction des informations dont on dispose d'après

nos connaissances de la langue traitée. Ces séquences de vecteurs, représentent la parole dans sa dimension linguistique.

1.4 Conclusion

Notre compréhension des appareils phonatoires et auditifs chez l'humain nous a permis de représenter efficacement la parole de manière numérique dans sa dimension acoustique. De plus, les connaissances sur l'élocution ont permis de définir une symbolique liée au contenu phonétique du message. Le passage d'une représentation à l'autre, soit des vecteurs acoustiques aux vecteurs linguistiques, ou inversement, est le coeur de la reconnaissance et de la synthèse de la parole. Chez l'humain, cette tâche met en œuvre des processus cognitifs complexes ayant pour objectif la compréhension du message transmis. Nous allons dans le prochain chapitre poser les bases algorithmiques de la modélisation de séquences, mécanisme fondamental de notre étude.

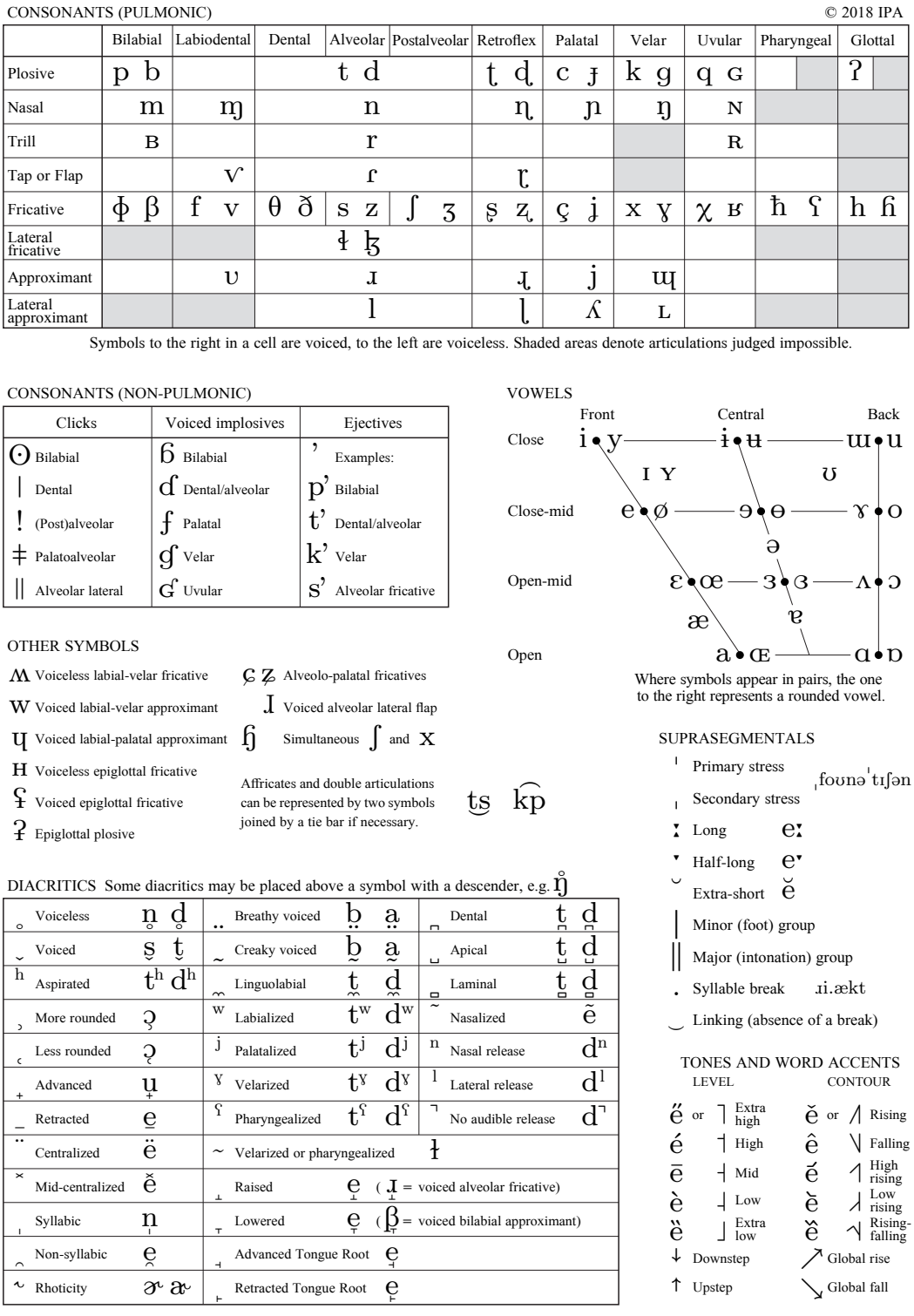

FIGURE 1.5 – L'Alphabet Phonétique International (révision 2018) : http ://www.internationalphoneticassociation.org/content/ipa-chart

Chapitre 2

Modélisation de séquences

Sommaire

NOUS avons vu qu'une production de parole peut être décrite par une séquence temporelle finie. L'un des problèmes fondamentaux en traitement de la parole est de construire des modèles de telles séquences. Ces derniers cherchent à décrire les lois que suit la production des séquences observées afin de pouvoir reconnaître les éléments sous jacents à la production d'une séquence ou de prédire la production d'une séquence à partir d'éléments connus. Ceci amène naturellement à des applications pratiques, comme la reconnaissance et la synthèse de la parole.

Pour décrire les séquences produites dans le cadre de la parole, il est possible de se fonder sur des connaissances ou d'exploiter des données. Dans le premier cas, les séquences produites sont supposées suivre un ensemble de règles. Ces règles sont déterminées de manière experte d'après la connaissance de la nature des séquences modélisées, et certains paramètres sont calculés sur un ensemble d'observations. Dans le deuxième cas, la production de séquences est supposée suivre un processus stochastique dont les paramètres sont estimés statistiquement à partir d'observations. Bien que la modélisation selon des règles montre des résultats satisfaisants pour décrire la parole, nous nous attachons dans ce chapitre à la présentation de modèles statistiques. Nous reviendrons sur les autres approches dans les chapitres dédiés aux tâches de reconnaissance et de synthèse de la parole.

La première hypothèse de modélisation est de considérer la parole comme un processus stochastique où des symboles sont émis à chaque instant, la probabilité d'émission d'un symbole dépendant des symboles émis précédemment. Selon la nature des séquences traitées, issues d'une paramétrisation de la parole sur les plans acoustiques et linguistiques, la modélisation présente des spécificités. D'une part, les vecteurs acoustiques ne sont pas tirés d'un alphabet fini, comme le peuvent être les symboles linguistiques. D'autre part, les dimensions acoustiques et linguistiques de la parole sont liées. Il est donc nécessaire de modéliser de façon jointe les séquences acoustiques et linguistiques, bien qu'elles aient des tailles différentes et ne suivent pas la même échelle de temps.

Nous présentons dans ce chapitre les principes de la modélisation de séquences adaptée au traitement de la parole dans le cadre de la reconnaissance et de la synthèse de celle-ci. Nous nous attachons ainsi à décrire les modèles de séquences simples, aussi bien sur les plans acoustiques et linguistiques, avant de discuter de la modélisation de séquences jointes.

2.1 Modélisation de séquences simples

La parole, aussi bien dans sa représentation acoustique que linguistique, peut être considérée comme un processus aléatoire où à chaque instant sont émis des symboles. Pour simplifier la présentation, nous considérons que ces symboles sont issus d'un ensemble fini S. Nous verrons ensuite que certains modèles peuvent être généralisés pour des points d'un espace continu.

Dans le cas le plus simple, l'émission d'un symbole ne dépend pas de l'historique d'émission du processus aléatoire : il s'agit alors d'une source sans mémoire. Cependant, dans le cadre de la parole, l'hypothèse faite est que le processus suit une chaîne de Markov, c'est-à-dire que chaque symbole produit dépend des symboles

déjà émis. Étant donné une séquence d'observations $X = (x_1, x_2, ..., x_T)$, la probabilité $P(X)$ d'observer une telle séquence s'écrit comme le produit des probabilités d'observer chaque élément de la séquence sachant les éléments précédents :

$$P(X) = P(x_1) \prod_{t=2}^{T} P(x_t | x_1, ..., x_{t-1}) \tag{2.1}$$

Construire un modèle statistique de la parole revient alors à estimer la probabilité d'observer toute séquence d'éléments possibles.

2.1.1 Modèles n-grammes

Un modèle n-grammes repose sur l'hypothèse que la probabilité d'observer un symbole x_t d'une séquence $X = (x_1, x_2, ..., x_T)$, avec $t \leq T$, dépend uniquement des $n-1$ symboles précédents. Ainsi, la relation 2.1 devient :

$$P(X) = P(x_1) \prod_{t=2}^{T} P(x_t | x_{t-(n-1)}, ..., x_{t-1}) \tag{2.2}$$

En limitant l'historique, cela permet de limiter l'ensemble des séquences possibles à $card(S)^n - 1$ possibilités. Ainsi, un graphe des séquences possibles peut être établi, où chaque nœud représente un symbole possible et où chaque liaison est pondérée par la probabilité d'accéder au nœud à partir des nœuds précédents. Un modèle n-gramme est alors vu comme un automate pondéré à états finis (figure 2.1).

Ces probabilités sont estimées statistiquement en maximisant la vraisemblance d'un ensemble de séquences observées, constituant un corpus d'apprentissage. Par exemple, dans le cadre de la modélisation du langage, la probabilité d'observer une séquence de n mots est estimée en calculant la fréquence d'une telle séquence de mots dans un texte. Comme il n'est pas possible d'avoir des données d'apprentissage contenant l'ensemble des séquences de symboles possibles, des techniques de lissage permettent d'attribuer une probabilité non nulle aux séquences non observées pendant l'apprentissage. Le détail de l'estimation de ces probabilités est donné dans (CHEN et GOODMAN, 1999).

Bien qu'ils soient encore très utilisés grâce à leur efficacité algorithmique, les modèles n-grammes font l'objet de plusieurs limitations. Premièrement, la taille de l'historique doit être fixée et est nécessairement de taille réduite. En effet, la modélisation repose sur la description de l'espace des séquences possibles. Même avec des techniques d'élagage, il est par exemple difficile d'avoir des valeurs de n supérieures à 3 ou 4 pour un ensemble de 150 000 symboles. L'hypothèse de départ (relation 2.2) devient alors très réductrice dans la mesure où les dépendances longues, c'est-à-dire dépassant la fenêtre d'observation, ne seront pas modélisées. Deuxièmement, les probabilités de transition d'un symbole à un autre sont estimées pour chaque symbole uniquement en fonction de leur fréquence d'observation dans un certain contexte. Par conséquent, les poids ne sont pas partagés entre les symboles partageant le même contexte. Cela est problématique dans la mesure où les symboles ne sont généralement pas uniformément représentés dans les données d'apprentissage et certains symboles sous représentés sont donc difficilement modélisés.

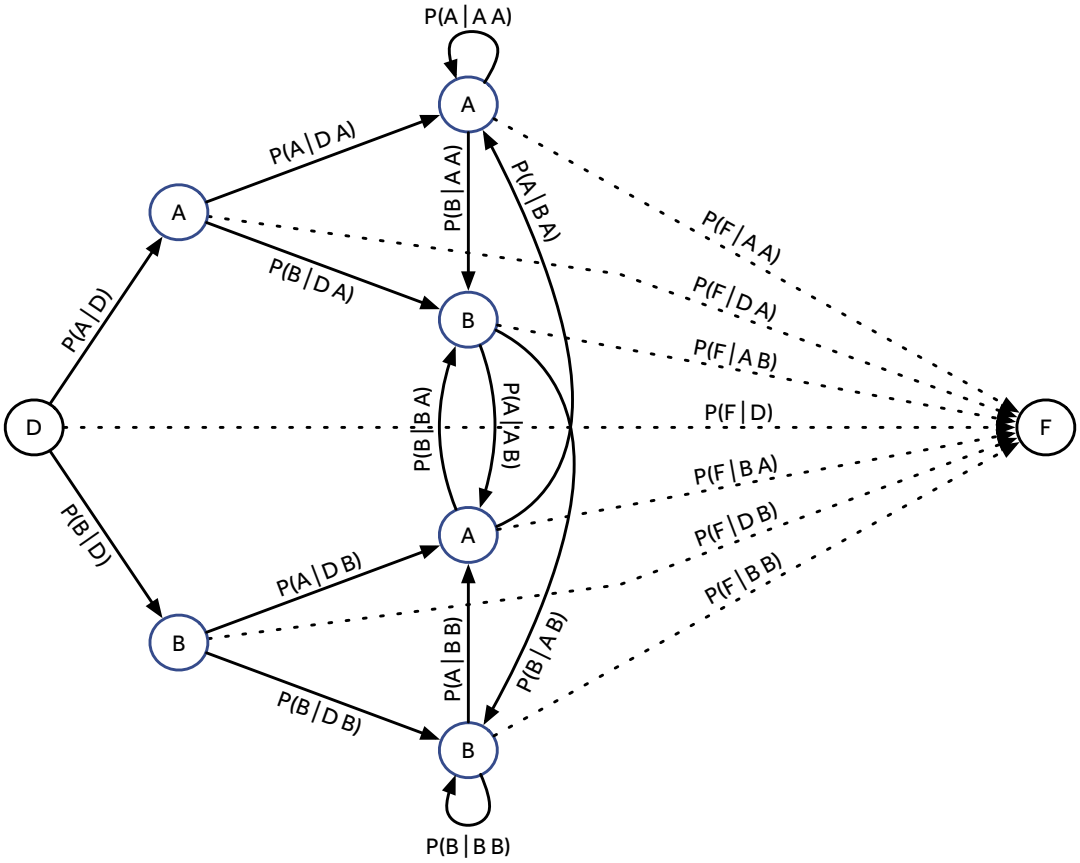

FIGURE 2.1 – Graphe d'un modèle trigramme pour un ensemble de symboles $S = \{A, B\}$. D et F sont respectivement les marqueurs de début et de fin de séquence, les liaisons sont pondérées par les probabilités de transition d'un nœud à un autre.

2.1.2 Réseaux de neurones

Les neurones formels et leur organisation en réseaux ont été proposés dans les années 1950 comme un modèle mathématique simplifié des neurones biologiques. L'objectif était de concevoir des modèles simples pouvant être combinés en des structures complexes afin d'approximer n'importe quelle fonction mathématique à partir d'exemples. Leur capacité de généralisation permettent ainsi aux réseaux de neurones de capturer la distribution de données observées pour estimer leur probabilité d'observation. La complexité de l'architecture d'un réseau de neurones étant liée à la complexité des données à modéliser, il a été longtemps difficile de les utiliser pour des applications pratiques. Aujourd'hui, grâce aux progrès algorithmiques, à la facilité pour collecter des données, et aux machines suffisamment puissantes pour mener à bien les calculs requis, on observe un regain d'intérêt pour ce type de modèles. Ainsi, de nombreuses limites ont été dépassées dans des domaines aussi variés que la classification d'images (LECUN, KAVUKCUOGLU et FARABET, 2010), la modélisation du langage (SCHWENK, 2007, MIKOLOV et al., 2010), la modélisation acoustique (HINTON et al., 2012 ; AMODEI et al., 2016) ou la synthèse de la parole (OORD et al., 2016 ; WANG et al., 2017 ; SHEN et al., 2018 ; ARIK et al., 2017 ; GIBIANSKY et al., 2017 ; KALCHBRENNER et al., 2018). Pour avoir une compréhension détaillée des réseaux de neurones, nous invitons le lecteur à consulter (GOODFELLOW et BENGIO, 2016),

livre sur lequel nous nous appuyons pour écrire cette partie.

Principe

Un neurone formel à n entrées est une fonction mathématique définie comme :

$$f : \left| \begin{array}{ccc} \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ \mathbf{e} & \longmapsto & \phi(\mathbf{w} \cdot \mathbf{e}^\top + b) \end{array} \right. \tag{2.3}$$

avec :

— ϕ une fonction d'activation
— $\mathbf{w} = (w_1, ..., w_n)$ un vecteur de poids associé à l'entrée $\mathbf{e} = (e_1, ..., e_n)$
— b un nombre réel appelé *biais* agissant comme un seuil d'activation dans le cas du *perceptron* (Rosenblatt 1957).

Le choix de la fonction d'activation dépend du type de résultat souhaité. Par exemple, ce peut être une sigmoïde, une tangente hyperbolique ou une unité de rectification linéaire (ReLU) pour une classification binaire, ou une fonction *softmax* dans le cas d'une classification multiclasses. Une combinaison de ces fonctions peut également être utilisée pour calculer la sortie d'un neurone. Elle correspond au potentiel d'activation des neurones biologiques.

Un réseau de neurones est un ensemble de neurones interconnectés organisé en plusieurs couches (figure 2.2). Chaque neurone étant une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$, une couche L de m neurones est définie comme :

$$L : \left| \begin{array}{ccc} \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ \mathbf{e} & \longmapsto & (\phi(W_1 \cdot \mathbf{e}^\top + b_1), ..., \phi(W_m \cdot \mathbf{e}^\top + b_m)) \end{array} \right. \tag{2.4}$$

avec :

— ϕ une fonction d'activation
— W une matrice de poids de taille $m \times n$, où W_i est le vecteur de poids associé à l'entrée $\mathbf{e}$ pour le i-ème neurone de la couche L
— $\mathbf{b} = (b_1, ..., b_m)$ le vecteur de biais [1] associé à la couche L

Le nombre de couches du réseau donne la *profondeur* de celui-ci, ce terme étant à l'origine de l'appelation Réseau de Neurones Profond (Deep Neural Network) (DNN). Cette organisation en couches permet au réseau de neurones d'approximer n'importe quelle fonction mathématique (CYBENKO, 1989 ; HORNIK, 1991) La première couche du réseau est appelée *couche d'entrée* et la dernière couche est appelée *couche de sortie*. Les couches intermédiaires sont appelées *couches cachées* dans la mesure où leur comportement n'est pas décrit par les données d'apprentissage.

L'apprentissage d'un réseau de neurones est de type *supervisé*, c'est-à-dire que les données d'apprentissage sont constituées de couples $(\mathbf{x}, \mathbf{y})$, avec $\mathbf{y}$ la sortie attendue pour l'entrée $\mathbf{x}$. Il consiste à estimer les paramètres du modèle, constitués des poids pour les différentes couches, ainsi que les biais. Après une initialisation aléatoire des paramètres, il se fait par itérations sur les données d'apprentissage en plusieurs étapes :

1. Dans la suite, nous ne mentionnerons pas systématiquement le biais, bien qu'il soit généralement appliqué à chaque couche.

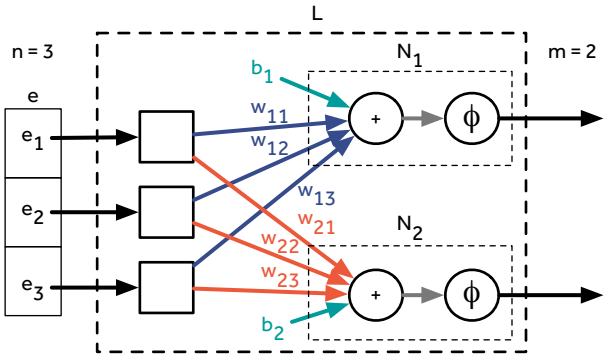

FIGURE 2.2 – Couche L de deux neurones N_1 et N_2 acceptant en entrée un vecteur **e** de trois coefficients.

— la couche de sortie du réseau de neurones est calculée en fonction de l'entrée et des paramètres du modèle
— une fonction de coût est évaluée en comparant la prédiction du modèle et la sortie attendue
— le gradient de chaque paramètre est calculé d'après le coût observé
— les paramètres du modèle sont mis à jour selon un algorithme d'optimisation (descente de gradient stochastique, Adam (KINGMA et BA, 2014), Adadelta (ZEILER, 2012), Adagrad (DUCHI, HAZAN et SINGER, 2011))

Nous nous intéressons ici à trois architectures pour la modélisation de séquences : le perceptron multicouches, le réseau de neurones convolutif et le réseau de neurones récurrent. Des architectures plus complexes spécifiques à la reconnaissance et à la synthèse de parole seront présentées ensuite.

Perceptron multicouches

Un Perceptron multicouches (Multi Layer Perceptron) (MLP) est un réseau de neurones composé d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie. Une particularité du MLP est que les neurones de deux couches successives sont tous connectés et que l'information circule en sens unique, de l'entrée vers la sortie (figure 2.3).

La modélisation de séquences avec un MLP suit le même principe que la modélisation de n-grammes (section 2.1.1). Elle a par exemple été proposée par (SCHWENK, 2007) pour la modélisation du langage. Estimer la probabilité d'observer une séquence de vecteurs $X = (x_1, ..., x_T)$ revient alors à estimer les probabilités $P(x_t|x_{t-(n-1)}, ..., x_{t-1})$ pour tout élément x_t de X (relation 2.2).

Tout d'abord, il est nécessaire de définir une manière de représenter un symbole $x_t \in S$. Afin de donner un poids identique à chaque symbole vis-à-vis du MLP, il peut être représenté par un vecteur *one-hot*. Autrement dit, le i-ème symbole de S est représenté par la i-ème ligne de la matrice identité $I_{card(S)}$. Ensuite, pour estimer les probabilités $P(x_t|x_{t-(n-1)}, ..., x_{t-1})$, le MLP doit prendre en entrée la séquence d'observations $(x_{t-(n-1)}, ..., x_{t-1})$. Pour cela, la séquence d'observation est représentée par la concaténation des représentations vectorielles des symboles de la séquence. La couche d'entrée du MLP accepte alors cette séquence grâce à $n - 1 \times card(S)$

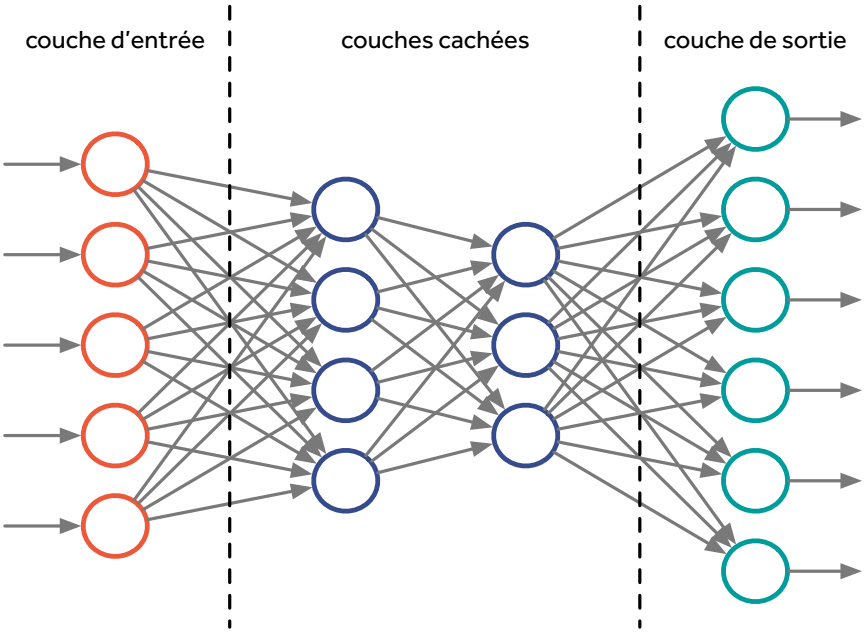

FIGURE 2.3 – Perceptron multicouches à deux couches cachées, six neurones de sortie, et acceptant des vecteurs d'entrées de cinq coefficients.

neurones. Enfin, la couche de sortie est composée de $card(S)$ neurones, le i-ème neurone donnant une estimation de $P(x_t = i|x_{t-(n-1)}, ..., x_{t-1})$ en appliquant la fonction $softmax$ à ses entrées.

Une limite de la représentation one-hot est de représenter les symboles dans un espace discret. Cela limite les possibilités d'interpolation pour estimer la probabilité d'observation de séquences absentes du corpus d'apprentissage. Ainsi, de nombreuses méthodes cherchent à représenter des symboles discrets dans un espace continu (SCHWENK, 2007 ; BENGIO et HEIGOLD, 2014 ; MIKOLOV et al., 2013 ; GIBIANSKY et al., 2017). Ces représentations sont couramment appelées *embeddings* et conduisent à représenter les symboles par des vecteurs à coefficients réels. Ils sont obtenus en multipliant la représentation one-hot par une matrice de projection (figure 2.4). Les coefficients de cette matrice sont soit estimés en même temps que les autres paramètres du réseau de neurones pendant la phase d'apprentissage, soit appris sur une autre tâche. Cela offre par exemple la possibilité de transférer des connaissances entre plusieurs modèles neuronaux. Un autre avantage de cette représentation est que les embeddings ont une taille inférieure aux vecteurs one-hot. Ceci permet de réduire le nombre de neurones de la couche d'entrée du réseau et donc d'avoir moins de paramètres à estimer pour projeter l'entrée sur la première couche cachée dans le cadre du MLP.

Finalement, un MLP permet de modéliser une séquence comme un modèle $n-grammes$ sans expliciter l'ensemble des séquences de symboles possibles. Ainsi, un historique plus long des séquences à traiter peut être considéré. De plus, le problème de l'estimation de probabilités pour les séquences de symboles qui ne sont pas observées pendant l'apprentissage est géré par une projection dans un espace continu.

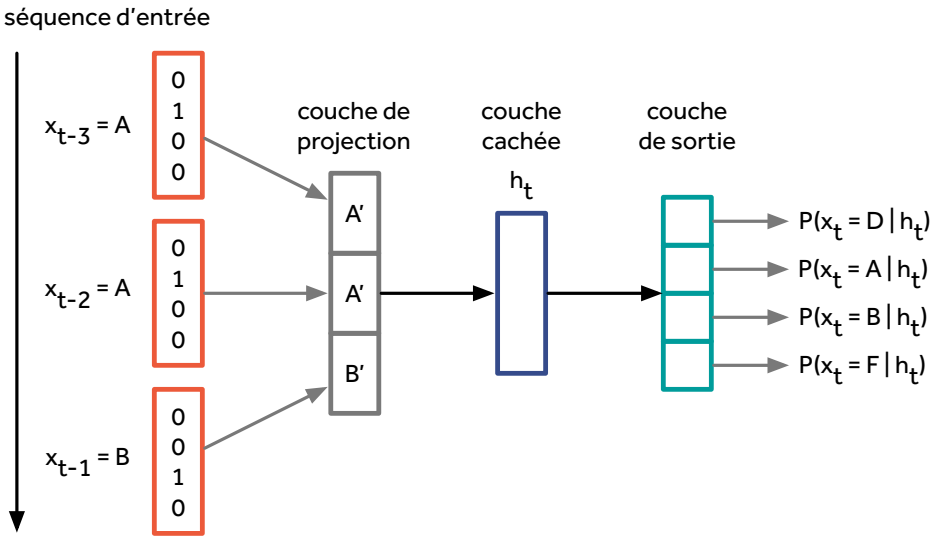

FIGURE 2.4 – Perceptron multicouches pour la modélisation 4-grammes de séquences d'éléments de $S = \{A, B\}$. Les éléments de la séquence d'entrée, représentés par des vecteurs *one-hot*, sont projetés dans un espace continu avant d'alimenter une couche cachée. La couche de sortie donne une distribution de probabilité sur les éléments pouvant poursuivre la séquence. D et F sont respectivement les marqueurs de début et de fin de séquence.

Ceci permet une meilleure généralisation et rend possible l'exploitation et la combinaison de représentations de différentes nature (MIKOLOV et al., 2013 ; PENNINGTON, SOCHER et MANNING, 2014 ; GHANNAY, 2017). Cependant, les MLPs considèrent de façon égale l'ensemble de la séquence, sans prendre en compte d'éventuelles dépendances plus locales, et ne permettent pas de gérer un historique de grande taille, qui dépend de la taille de la couche d'entrée. Il ne sont donc pas adaptés pour la modélisation de séquences longues, comme un signal de parole, et doivent pour cela être combinés avec des Modèles de Markov Cachés (Hidden Markov Model)s (HMMs).

Réseaux de neurones récurrents

Les Réseaux de Neurones Récurrents (Recurrent Neural Network) (RNN) ont été proposés par (JORDAN, 1986 ; ELMAN, 1990) pour répondre au problème de modélisation de séquences temporelles de tailles variables. En effet, lorsqu'un MLP traite une séquence, la couche d'entrée comporte autant de neurones que d'éléments dans la séquence. Il est donc nécessaire de traiter des séquences de taille fixe ou d'appliquer une fenêtre glissante. Dans ce dernier cas, il n'est pas possible de capturer des dépendances situées au-delà de la fenêtre d'observation. De plus, les paramètres du réseau de neurones sont dépendants de la position de chaque élément dans la séquence, ce qui devient problématique dans le cadre de fenêtres glissantes ou lorsque l'ordre des éléments de la séquence de sortie est indépendant de l'ordre des éléments de la séquence en entrée. Avec un RNN, les même paramètres sont partagés par tous les éléments de la séquence. Le principe est de présenter au RNN les éléments de la séquence dans l'ordre chronologique et de calculer la sortie, non seulement selon l'entrée, mais aussi selon les sorties précédentes (figure 2.5). Pour accentuer le poids

des éléments dans le contexte proche de l'élément courant, une fenêtre d'observation peut être utilisée, en concaténant les représentations de l'élément courant et des éléments voisins. Dans tous les cas, l'entrée du RNN est représentée par un vecteur, qui peut résulter de la concaténation d'une séquence de vecteurs.

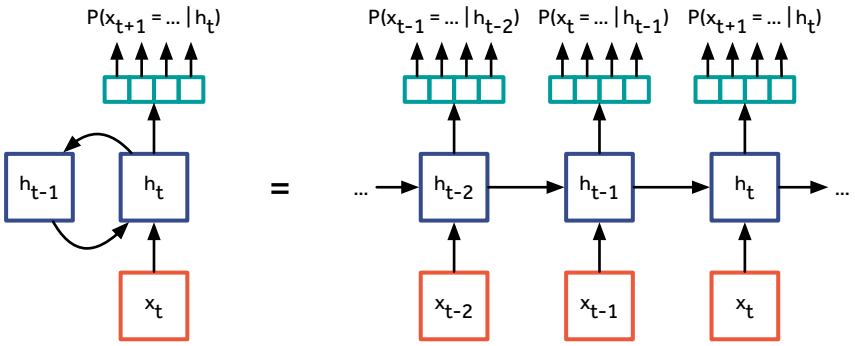

FIGURE 2.5 – Modélisation d'une séquence à l'aide d'un réseau de neurones récurrent à une couche cachée. Les éléments de la séquence d'entrée, représentés par des vecteurs *one-hot* ou des *embedding*, sont présentés un par un dans l'ordre chronologique. La couche de sortie donne une distribution de probabilité sur les éléments pouvant poursuivre la séquence, l'élément suivant est déterminé en prenant l'indice de l'élément obtenant la plus grande probabilité.

Le problème d'une telle architecture est que le gradient d'erreur peut ou disparaître, ou exploser, lors de la rétropropagation. La raison a été identifiée comme étant liée aux valeurs que prennent les dérivées des fonctions d'activation et à la profondeur du réseau. En effet, un RNN à une couche est équivalent à un MLP avec une couche par intervalle de temps. La profondeur réelle d'un RNN peut donc devenir très importante pour des séquences longues. Une solution est de considérer des unités Long Short-Term Memory (LSTM) (HOCHREITER et SCHMIDHUBER, 1997) pour calculer l'état interne des couches cachées (figure 2.6). Ces unités sont une architecture particulière de couche neuronale. Elles possèdent des portes associées à des fonctions d'activation permettant de conserver ou non la valeur des états intermédiaires. Depuis leur utilisation, de nombreuses variantes ont été proposées, et notamment les unités Gated Recurrent Unit (GRU) (CHO et al., 2014) qui possèdent une organisation des portes plus simple.

De plus, pour tenir compte des dépendances passées et futures, le réseau peut être bidirectionnel, c'est-à-dire que la séquence est présentée dans les deux sens à la fois.

Les RNNs forment une architecture neuronale particulièrement adaptée pour la modélisation de séquences. Cependant, pour traiter des séquences très longues, il est parfois plus judicieux de considérer une architecture de Réseau de Neurones Convolutif (Convolutional Neural Network) (CNN).

Réseaux de neurones convolutifs

Les CNNs sont des réseaux de neurones conçus pour modéliser des données structurées en grille, comme les pixels d'une image ou des vecteurs acoustiques dans le plan temps-fréquence par exemple. Une séquence $X = (x_1, ..., x_T)$ sera ainsi vue comme une matrice où chaque colonne correspond à la représentation vectorielle

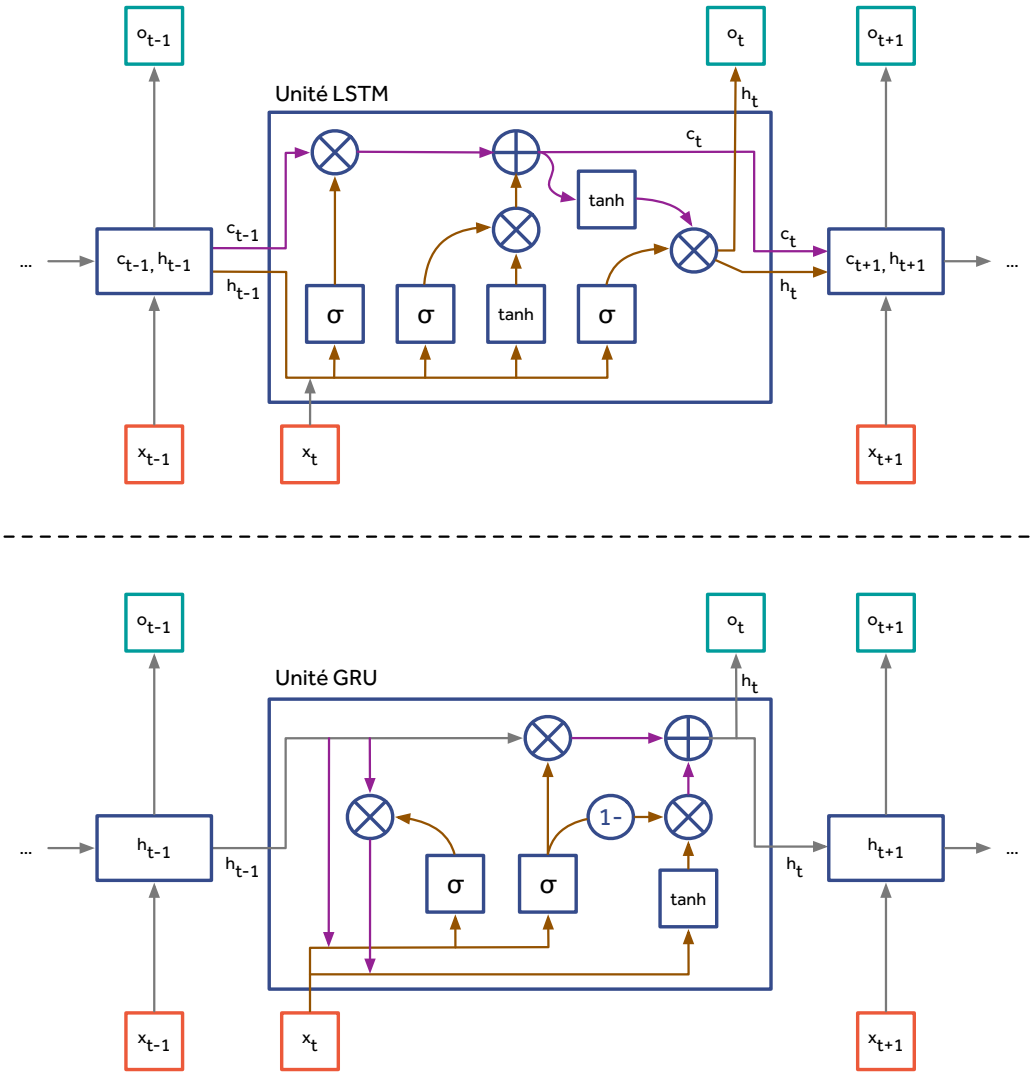

FIGURE 2.6 – Détail d'une unité LSTM et d'une unité GRU dans un réseau de neurones récurrent

d'un symbole x_t. La particularité des CNNs est d'utiliser le produit de convolution à la place du produit matriciel pour pondérer les entrées d'un neurone. La convolution est mise en œuvre à l'aide d'un filtre appliqué comme une fenêtre glissante sur la séquence d'entrée. Ceci permet de réduire le nombre de connexions entre deux couches neuronales, de partager les paramètres de pondération pour toute la séquence d'entrée et d'avoir une modélisation indépendante de la position. Le principe des CNNs s'inspire notamment de l'organisation et du fonctionnement du cortex visuel (HUBEL et WIESEL, 1962).

Un CNN est un réseau de neurones constitué de plusieurs couches de convolution successives, où l'information circule en sens unique comme pour les MLPs. Le passage d'une couche à l'autre se fait en trois étapes : une transformation affine, l'application d'une fonction d'activation non linéaire et une opération de sous-échantillonnage. Contrairement aux MLPs, les neurones de deux couches successives ne sont pas tous interconnectés : chacun d'entre eux possède un champ récepteur, c'est-à-dire qu'il prend en entrée seulement une partie des sorties des neurones de la couche précédente (figure 2.7). Mis à part ces différences avec l'architecture

d'un MLP, le problème de modélisation d'une séquence est traité de la même manière, soit comme une modélisation de n-grammes neuronale.De plus, s'agissant de séquences, nous utilisons un cas particulier des CNNs à convolution causale : l'état des neurones reliés à l'élément courant de la séquence ne dépend pas des éléments futurs.

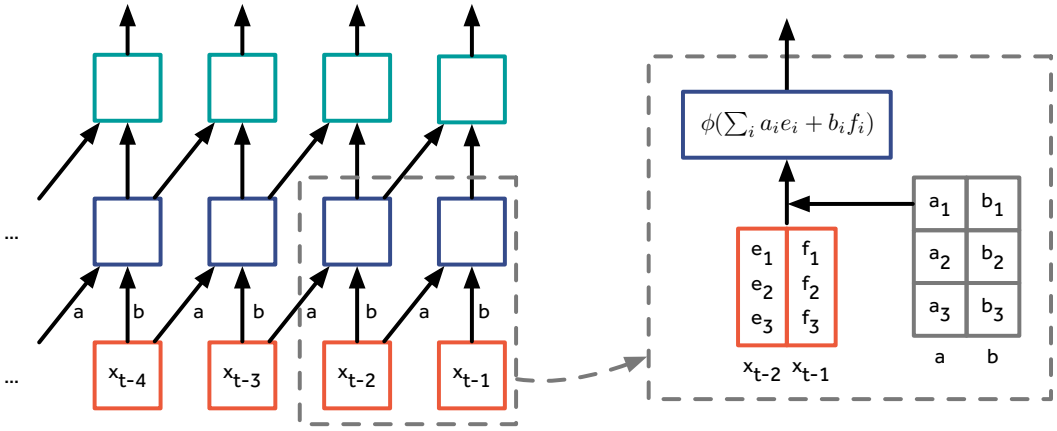

FIGURE 2.7 – CNN pour la modélisation de séquences

L'étape de transformation affine consiste à convoluer la matrice d'entrée avec un filtre mobile dont les paramètres sont estimés lors de l'apprentissage du CNN. Le produit de convolution est une opération mathématique commutative qui à deux fonctions f et g associe une fonction $f * g$ définie par :

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(t-a)g(a)\mathrm{d}a \tag{2.5}$$

Cette opération peut être réalisée dans le domaine discret à l'aide du produit matriciel. Ainsi, pour une matrice de données d'entrée E à deux dimensions et un filtre H de taille $M \times N$, la matrice de sortie S est donnée par :

$$\begin{aligned} S_{i,j} &= (E * H)_{i,j} &&= \sum_{m=1}^{M}\sum_{n=1}^{N} E_{m,n}H_{i-m,j-n} \\ &= (H * E)_{i,j} &&= \sum_{m=1}^{M}\sum_{n=1}^{N} E_{i-m,i-n}H_{m,n} \end{aligned} \tag{2.6}$$

Généralement, le filtre a beaucoup moins de paramètres que l'entrée, ce qui permet de réduire considérablement le nombre de paramètres à estimer pendant l'apprentissage du CNN par rapport à un MLP. C'est pourquoi, nous préférons généralement utiliser la deuxième ligne de l'équation 2.6.

Après avoir appliqué une fonction d'activation non linéaire sur le résultat de la convolution, une autre caractéristique des CNNs est d'effectuer une opération de sous-échantillonnage (*pooling*). Cette opération, qui n'est pas systématique, permet de mettre en évidence certaines caractéristiques de l'entrée tout en éliminant certaines valeurs non pertinentes. Elle permet également de réduire le nombre de paramètres, ce qui induit un temps d'apprentissage plus court, une vitesse de décodage plus grande et une empreinte mémoire plus faible.

Les CNNs sont utilisés dans différentes tâches de traitement de la parole, comme en reconnaissance de la parole (PEDDINTI, POVEY et KHUDANPUR, 2015) ou en synthèse de la parole, aussi bien pour la prédiction de paramètres acoustiques (PING et al., 2017) que pour le codage d'un signal de parole (OORD et al., 2016). En outre, ils permettent de réduire le nombre d'éléments d'une séquence en effectuant des regroupements en vue de la présenter à un RNN par exemple (AMODEI et al., 2016). En somme, les CNN permettent de limiter le nombre de paramètres à estimer lors de l'apprentissage tout en bénéficiant d'un décodage en parallèle. Nous verrons dans la suite comment les architectures de MLP, RNN et CNN sont combinées dans un même réseau de neurones pour la modélisation de séquences complexes, comme les séquences jointes.

2.2 Modélisation de séquences jointes

Jusqu'à maintenant, nous avons considéré la modélisation d'une seule séquence. Or, la reconnaissance et la synthèse de la parole ont pour objet la transformation d'une séquence en une autre. Cette problématique de conversion de séquences apparaît à plusieurs niveaux, comme la transcription d'un signal de parole, la génération d'une séquence de vecteurs de paramètres acoustiques à partir d'un texte, ou la phonétisation de mots. Il devient alors nécessaire de modéliser conjointement des séquences.

L'hypothèse faite pour la conversion d'une séquence X en une séquence Y est celle de la classification bayésienne, c'est-à-dire qu'on considère que la séquence Y à trouver est celle qui est la plus probable parmi l'ensemble des séquences possibles Y^*.

$$Y = \underset{Y' \in Y^*}{\operatorname{argmax}}\, P(X, Y') \tag{2.7}$$

Cette équation peut se réécrire à l'aide de probabilités conditionnelles :

$$Y = \underset{Y' \in Y^*}{\operatorname{argmax}}\, P(X|Y')P(Y') = \underset{Y' \in Y^*}{\operatorname{argmax}}\, P(Y'|X) \tag{2.8}$$

X étant connu, $P(X)$ est constante quelle que soit Y et n'intervient donc pas dans la relation 2.8. Ainsi, plusieurs solutions permettent de déterminer la séquence Y associée à X :

— modéliser X et Y conjointement pour pouvoir estimer $P(X, Y)$
— estimer séparément $P(X|Y)$ à l'aide d'un modèle qui joint les séquences et $P(Y)$ avec un modèle pour séquences simples
— estimer plus directement $P(Y|X)$

Les différentes solutions ne sont pas systématiquement applicables à tous les cas d'usage, et le choix dépend de la nature des séquences. Nous nous intéressons ici aux modèles généralement utilisés en reconnaissance et en synthèse de la parole.

2.2.1 Modèles n-grammes joints

L'idée des modèles n-grammes joints est de modéliser la séquence (X, Y) comme une séquence simple d'éléments joints. Ceci se fait en deux étapes :

— aligner les éléments de X et Y afin de former des paires $(x, y) \in X^* \times Y^*$
— modéliser les séquences de paires $(x, y) \in X^* \times Y^*$ avec un modèles de séquences simples comme vu précédemment

L'alignement peut être réalisé à l'aide d'une table de correspondances, comme dans (Bagshaw, 1998), ou à l'aide d'algorithmes de programmation dynamique. Dans ce cas, le principe est de procéder par itération pour trouver le minimum de paires (x, y) nécessaires pour décrire les données d'apprentissage.

Historiquement, cette approche a été utilisée en traduction automatique (KOEHN et al., 2007) et en phonétisation (BISANI et NEY, 2008) Récemment, cette approche a été utilisée en synthèse de la parole pour conditionner un CNN (OORD et al., 2016) par une séquence source pour générer une séquence cible. Le conditionnement d'un CNN consiste simplement à concaténer un vecteur de conditionnement à chaque vecteur de la séquence d'entrée, elle même correspondant au passé de la séquence cible. Ces vecteurs de conditionnement représentent les éléments de la séquence source.

Une limite de cette approche est de nécessiter un alignement strict entre les séquences cible et les séquences sources. Selon les cas, cet alignement peut ne pas exister et est donc l'objet d'approximations. De plus, des erreurs d'alignement nuisent à l'apprentissage du modèle de séquences. Nous décrivons dans la suite des méthodes de modélisation autorisant un alignement plus souple.

2.2.2 Modèles de Markov Cachés

Les HMM sont des modèles statistiques de séquences temporelles permettant de traiter des séquences d'éléments indénombrables. Ils peuvent alors convenir pour modéliser des séquences de vecteurs acoustiques. Les HMMs ont été initialement proposés en reconnaissance de la parole pour la modélisation acoustique (RABINER, 1989), avant d'être appliqués par analogie à la synthèse de la parole (TOKUDA, KOBAYASHI et IMAI, 1995 ; MASUKO et al., 1996). Notre description de ces modèles s'appuie sur (VIRTANEN, SINGH et RAJ, 2012), dont le chapitre 2 pose les bases de la modélisation par HMMs pour la reconnaissance de la parole.

Principe

Un HMM modélise une séquence temporelle comme étant générée par une séquence d'états émetteurs, ces états formant une chaîne de Markov. Un processus aléatoire composé d'une chaîne de Markov transite d'un état à un autre, ou reste dans le même état, selon une certaine probabilité de transition. À chaque état est associée une fonction aléatoire générant une observation, représentée par un vecteur de paramètres. Les états d'un HMM sont dits cachés car un observateur extérieur ne peut voir que les sorties de ces fonctions aléatoires. Ces dernières sont modélisées par des Modèles de Mélanges Gaussiens (Gaussian Mixture Model) (GMM) ou des modèles neuronaux comme les DNN. Le modèle fait l'hypothèse que les observations sont indépendantes et que la probabilité que le modèle soit dans un état ne dépend que de l'état précédent.

Soit une séquence d'observations $X = (x_1, ..., x_T)$ générée par une séquence d'états $Q = (q_1, ..., q_T)$. Un HMM est défini par :

— un ensemble d'états émetteurs dont le nombre N est fixé empiriquement
— une distribution de probabilités initiales π, où $\pi(i)$ est la probabilité que l'état i soit l'état initial
— une matrice de transition A, où les coefficients $a_{i,j} = P(q_{t+1} = j|q_t = i)$ donnent la probabilité de transiter successivement de l'état i à l'état j
— un ensemble $B = \{b1, ..., b_N\}$ des densités de probabilité d'émission associées à chaque état, où $b_j(x_i) = P(x_i|q_t = j)$ est la probabilité qu'une observation x_i soit générée à l'état j au temps t

La figure 2.8 montre un HMM gauche-droit à 5 états dont 3 états sont émetteurs, avec un état d'entrée et un état de sortie non émetteurs. La figure montre également un exemple de séquence d'observations : l'état initial et l'état final, non émetteurs, sont respectivement les états 1 ($\pi(1) = 1$) et 5, et chaque état émetteur génère une ou plusieurs observations.

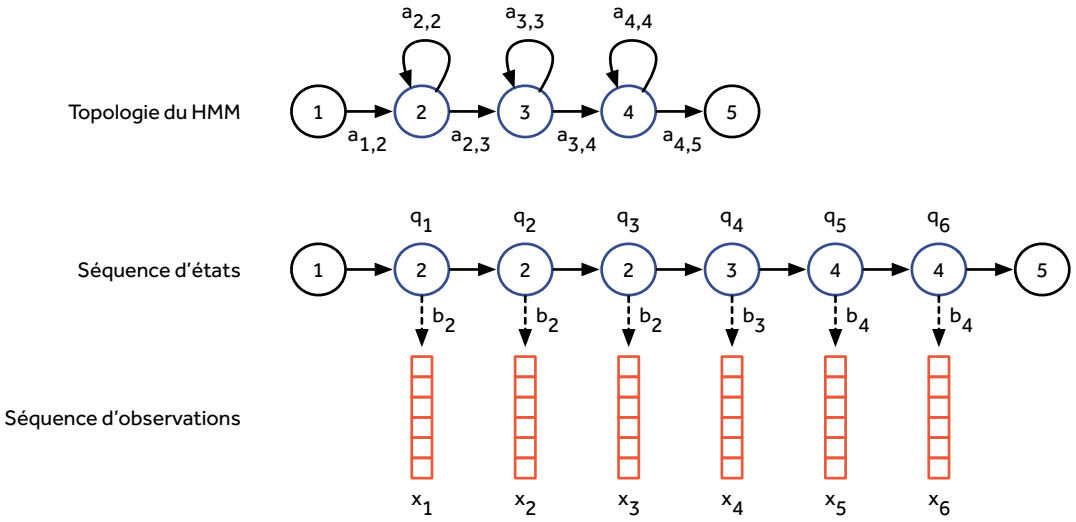

FIGURE 2.8 – Topologie d'un HMM à 5 états avec 3 états émetteurs, et exemple de génération d'une séquence d'observations

Lorsque B est donné par des GMMs, les densités de probabilité sont estimées par des sommes pondérées de densités de probabilité gaussiennes (relation 2.9). On parle alors de modèle HMM-GMM.

$$b_j(x_i) = \sum_{k=1}^{K} w_{j,k} \mathcal{N}(x_i; \mu_{j,k}, \Theta_{j,k}) \tag{2.9}$$

avec $\mathcal{N}(x_i; \mu, \Theta)$ une densité de probabilité gaussienne et $w_{j,k}$ les poids du mélange de gaussiennes. En pratique, K, le nombre de gaussiennes par mélange, est déterminé empiriquement.

Bien qu'ils comportent de nombreux avantages, les GMM montrent des limites dans la modélisation de données dont la distribution est non linéaire, comme c'est le cas pour la parole. Depuis quelques années, grâce aux avancées algorithmiques dans

l'apprentissage automatique et aux progrès techniques en matériel de calcul, les modèles neuronaux ont montré de meilleures capacités à exploiter les quantités de données disponibles aujourd'hui (HINTON et al., 2012). Ceci a conduit à la généralisation des DNN pour différentes tâches de modélisation, et donc au développement de modèles hybrides HMM-DNN. Dans cette approche, les HMM caractérisent la dynamique des séquences modélisées, alors que les DNN permettent d'estimer les probabilités a posteriori des états émetteurs des HMM. Les réseaux de neurones prennent en entrée des vecteurs représentant les observations, sur lesquels sont ensuite appliqués une suite de transformations non linéaires. D'autres types de modèles neuronaux peuvent être utilisés, comme les CNN ou les RNN que nous avons décrits précédemment.

Apprentissage

L'apprentissage d'un HMM consiste à estimer les paramètres du modèle de manière maximiser un certain critère. Il s'agit par exemple de maximiser la probabilité d'observation des séquences tirées de données d'apprentissage. Les paramètres à estimer sont la distribution de probabilités initiales π, les probabilités de transition A et les densités de probabilités d'émission B. Ces dernières sont, dans le cas de GMM, paramétrées par les moyennes et variances de chaque gaussienne ainsi que les poids de chacune d'elles dans le mélange. Lorsqu'un DNN est utilisé, il remplace les GMM d'un modèle HMM-GMM déjà estimé et l'apprentissage, que nous allons aborder dans la suite, consiste à estimer les poids du réseau.

Il n'est pas possible de trouver de manière analytique les paramètres maximisant la probabilité d'observer les séquences d'un corpus d'apprentissage. C'est pourquoi l'estimation des paramètres des HMM est réalisée de manière itérative selon un algorithme *Expectation-Maximization*. Après, une initialisation aléatoire des paramètres $\theta = (A, B, \pi)$, plusieurs itérations des étapes suivantes sont effectuées sur des échantillons $x_{1:T}$ d'un corpus d'apprentissage :

1. les probabilités $\alpha_i(t) = P(x_{1:t}, q_t = i|\theta)$ et $\beta_j(t) = P(x_{t+1:T}, q_t = j|\theta)$ sont calculées de manière récursive d'après A, B et π définis précédemment. $\alpha_i(t)$ est la probabilité que le modèle soit à l'état i au temps t après avoir généré la séquence d'observations $x_{1:t}$. $\beta_j(t)$ est la probabilité que le modèle génère la séquence d'observations futures $x_{t+1:T}$ en partant de l'état j au temps t.
2. en posant $\gamma_i(t) = \frac{P(x_{1:T}, q_t=i)}{P(x_{1:T})} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N}\alpha_j(t)\beta_j(t)}$ la probabilité a posteriori que le processus soit à l'état i au temps t pour une séquence d'observation $x_{1:T}$ du corpus d'apprentissage, et $\gamma_{i,j}(t) = P(q_t = i, q_{t+1} = j|x_{1:T})$ la probabilité que le processus soit à l'état i au temps t puis à l'état j au temps $t+1$ pour $x_{1:T}$, les nouvelles valeurs des paramètres $\theta = (A, B, \pi)$ sont calculés en fonction de $\gamma_i(t)$ et $\gamma_{i,j}(t)$.

Pour estimer directement la probabilité $P(x_{0:T})$ d'observer une séquence $x_{0:T}$, il est nécessaire de considérer l'ensemble des séquences d'états possibles. Les probabilités $\alpha_i(t)$ et $\beta_i(t)$ définies précédemment rendent ce calcul possible en sommant sur les états du HMM :

$$\begin{aligned} P(x_{0:T}) &= \sum_{i=1}^{N} P(x_{0:T}, q_t = i) \\ &= \sum_{i=1}^{N} \alpha_i(t)\beta_i(t) \end{aligned} \tag{2.10}$$

Pour aller plus loin, le détail des calculs est donné dans le chapitre 2 de (VIRTANEN, SINGH et RAJ, 2012).

Modélisation conjointe

Nous avons vu qu'une séquence X pouvait être modélisée comme étant une séquence d'observations émises par les états cachés d'un HMM. Lorsque la séquence X est beaucoup plus grande que la séquence Y, ce HMM peut être vu comme la composée de HMMs. Ceci est par exemple le cas en reconnaissance de la parole quand X est une séquence de vecteurs acoustiques et Y une séquence de phonèmes. Chaque valeur que peut prendre un élément de Y est alors modélisée par un HMM dont les états émetteurs produisent des éléments de X. La séquence X est ainsi modélisée comme résultant d'émissions d'un HMM défini comme la composée de HMMs élémentaires d'éléments de Y (figure 2.9).

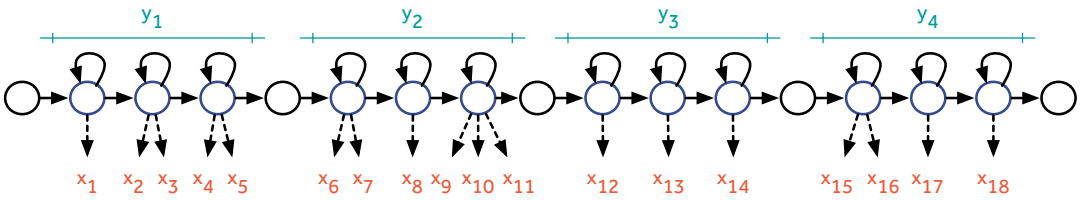

FIGURE 2.9 – Modélisation de séquences jointes par composition de modèles HMM élémentaires

Lorsque les séquences X et Y sont connues a priori, la modélisation conjointe permet un alignement des deux séquences : on parle d'alignement forcé. En revanche, lorsqu'il s'agit de déterminer la séquence $\hat{Y}$ la plus probable sachant X, cela revient à rechercher la séquence d'états la plus probable $\hat{Q} \in Q^*$ parmi les séquences d'états possibles dans le HMM modélisant X. Ensuite, il reste à déterminer les valeurs de $\hat{Y}$ d'après les densités de probabilité d'émission des états retenus et de la probabilité $P(Y)$ d'observer une telle composée de HMM :

$$\hat{Y} = \underset{Y \in Y^*}{\operatorname{argmax}} \left[\max_{Q \in Q^*} P(X, Q|Y)P(Y) \right] \tag{2.11}$$

La séquence d'états la plus probable $\hat{Q}$ est donnée par l'exécution d'un algorithme de Viterbi en s'appuyant sur les probabilités de transition dans le graphe des séquences d'états possibles.

En somme, la modélisation par HMMs permet d'estimer la probabilité d'observer une séquence Y sachant une séquence X, en alignant les éléments des deux séquences à l'aide d'états cachés émetteurs. Ceci permet de convertir une série temporelle en une autre, qui trouve des applications en reconnaissance et en synthèse

de la parole. Cependant, des contraintes importantes sont liées à l'utilisation de ces modèles, comme la possibilité d'estimer les paramètres d'un modèle pour chaque valeur possible de la séquence d'entrée, et l'hypothèse d'indépendance entre les éléments de la séquence de sortie. De plus, il est difficile de prendre en compte des dépendances longues dans la séquence d'entrée, ce qui est généralement le cas dans la modélisation de la parole. Nous allons voir que l'utilisation de réseaux de neurones permet de proposer des solutions alternatives à la modélisation de séries temporelles, avec des hypothèses de travail souvent moins contraignantes.

2.2.3 Architecture Encodeur-Décodeur

Nous avons vu que les RNNs permettent d'associer une distribution de probabilité pour chaque élément d'une séquence en entrée. Cette distribution peut servir à associer un nouveau symbole à chaque entrée. Ainsi, pour une séquence $X = (x_1, ..., x_T)$ en entrée, il est possible de déterminer une séquence $Y = (y_1, ..., y_T)$ de même longueur en sortie. Par ailleurs, en ne considérant que l'état de la couche cachée, on obtient un unique vecteur de taille fixe après avoir présenté en entrée une séquence X. De la même façon, un RNN peut générer une séquence à partir d'un unique vecteur de taille fixe. La combinaison de ces propriétés donne la possibilité de convertir une séquence $X = (x_1, ..., x_{T_x})$ en une séquence $Y = (y_1, ..., y_{T_y})$, respectivement de longueurs T_x et T_y différentes. C'est le principe de l'architecture encodeur-décodeur (figure 2.10) (CHO et al., 2014).

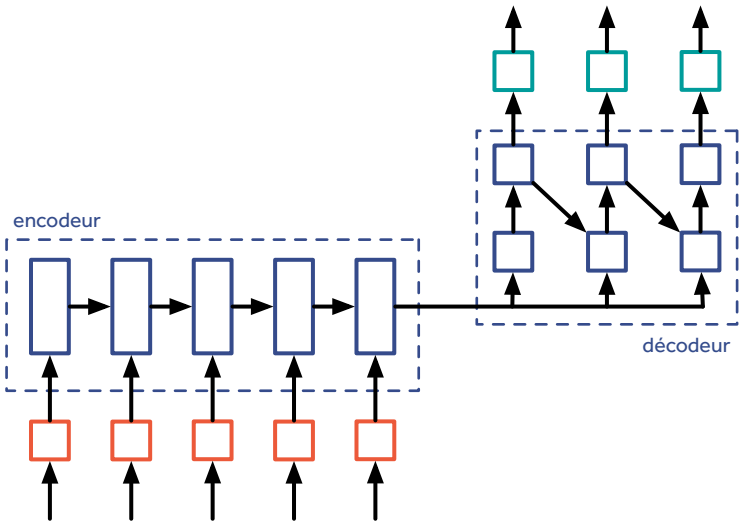

FIGURE 2.10 – Architecture encodeur-décodeur

L'encodeur est un RNN associant à la séquence d'entrée un vecteur de taille fixe. Ce RNN peut être bidirectionnel afin de considérer les deux sens de lecture de la séquence. Dans ce cas, les couches cachées associées à chaque sens de lecture sont concaténées (GOODFELLOW et BENGIO, 2016 ; II.10.2-3). Ce vecteur représentatif de la séquence d'entrée est ensuite donné à un décodeur. Ce dernier est également un RNN qui associe une séquence de distributions de probabilité à son entrée de taille fixe. Autrement dit, il permet de calculer à chaque instant t une probabilité $P(y_t|X)$

pour chaque symbole y_t possible. Ainsi, il est possible d'obtenir en sortie une séquence Y de taille différente de la séquence d'entrée X, la taille de la séquence de sortie étant déterminée par le décodeur :

$$Y = \underset{Y'}{\operatorname{argmax}}\, P(Y'|X) = \underset{y_t}{\operatorname{argmax}} \prod_{t=1}^{T_y} P(y_t|X) \tag{2.12}$$

Une limite de cette architecture est de modéliser inégalement les séquences d'entrées selon leur longueur. De plus, il est difficile pour le décodeur de donner une probabilité importante au symbole de fin de séquence sans informations supplémentaires. Dans (BAHDANAU, CHO et BENGIO, 2014), les auteurs proposent une méthode pour calculer une représentation de la séquence d'entrée pour chaque état du décodeur. Ceci est réalisé en pondérant les états de l'encodeur à l'aide de poids déterminés d'après l'état courant du décodeur : c'est le mécanisme d'attention (figure 2.11).

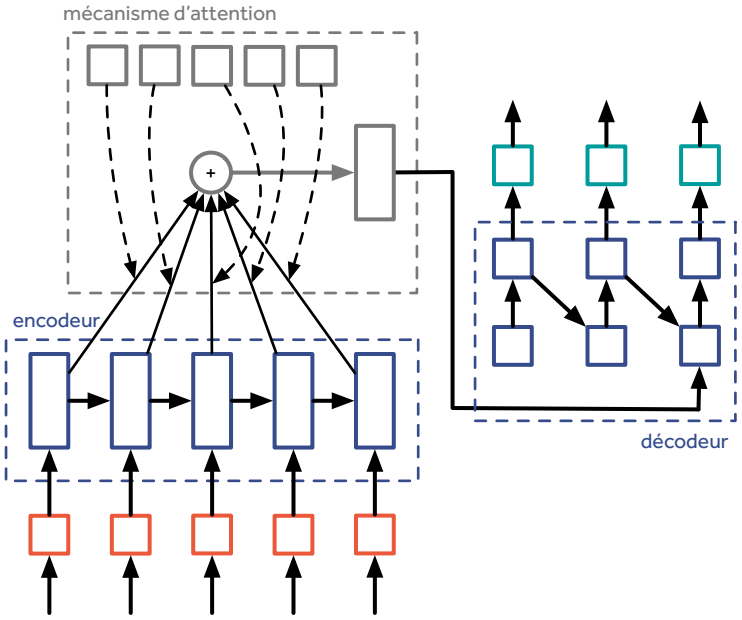

FIGURE 2.11 – Architecture encodeur-décodeur avec mécanisme d'attention

Soit $\mathbf{h_1}, ..., \mathbf{h_{T_x}}$ les états de l'encodeur associés à la séquence d'entrée $X = (x_1, ..., x_T x)$ et $\mathbf{s_1}, ..., \mathbf{s_{T_y}}$ les états du décodeur associés à la séquence de sortie $Y = (y_1, ..., y_{T_y})$. Le principe est de calculer pour chaque s_i un vecteur $\mathbf{e_i} = (e_{i1}, ..., e_{i,T_x})$ tel que $e_{ij} = \sigma(s_{i-1}, h_j)$. En normalisant $\mathbf{e_i}$, on obtient des coefficients $a_1, ..., a_{T_x}$, appelés poids d'attention, qui vérifient :

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.13}$$

Ces poids d'attention permettent alors de calculer $\mathbf{c_i}$, le vecteur représentatif de la séquence d'entrée à l'état i du décodeur :

$$\mathbf{c_i} = \sum_{j=1}^{Tx} a_{ij}\mathbf{h_j} \tag{2.14}$$

On obtient ainsi une séquence $C = (c_1, ..., c_{T_y})$ représentative de la séquence d'entrée, mais de longueur identique à la séquence de sortie.

L'architecture encodeur-décodeur est très utilisée pour les tâches de conversion de séquences, notamment la phonétisation (RAO et al., 2015 ; YAO et ZWEIG, 2015), la traduction automatique (BAHDANAU, CHO et BENGIO, 2014), la reconnaissance de la parole (LU et al., 2015) ou la synthèse de la parole (WANG et al., 2017 ; SHEN et al., 2018). Il est important de noter que du fait de son rôle, l'encodeur peut facilement être substitué par un CNN, comme dans (PING et al., 2017) où les auteurs proposent un système de synthèse de la parole entièrement neuronal.

2.2.4 Classification Temporelle Connexionniste

La classification Connectionist Temporal Classification (CTC) a été proposée dans (GRAVES et al., 2006) comme une fonction de coût permettant d'entraîner un réseau de neurones récurrent à associer des séquences Y à des séquences X sans nécessiter d'alignement entre les éléments des séquences jointes. Elle permet ainsi de d'estimer une probabilité $P(Y|X)$ pour des séquences X et Y de taille différente. Nous avons vu qu'un RNN estimait un élément de sortie pour chaque élément en entrée. La fonction CTC utilise un symbole ϵ ajouté aux symboles possibles en sortie pour agir comme un délimiteur, et la fusion des étiquettes en sortie selon des règles de regroupement pour retrouver la séquence de sortie correcte.

La figure 2.12 donne les différentes étapes de l'algorithme CTC :

1. préparation de la séquence d'entrée (exemple : une séquence de trames acoustiques)
2. production d'une séquence en sortie à partir de cette entrée à l'aide d'un RNN
3. association de chaque entrée à une distribution de probabilité sur l'ensemble des symboles possibles en sortie
4. estimation de la probabilité d'obtenir différentes séquences en sortie d'après l'ensemble de distribution de probabilités
5. projection des séquences générées sur l'espace des séquences cibles pour obtenir les différentes séquences de sortie possibles et l'estimation de leur probabilité d'observation

Autrement dit, la fonction CTC permet d'estimer la probabilité $P(Y|X)$ d'observer une séquence Y connaissant une séquence X sans nécessiter d'alignement explicite :

$$P(Y|X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} P_t(a_t|X) \tag{2.15}$$

avec $A_{X,Y}$ l'ensemble des alignements valides entre les séquences X et Y.

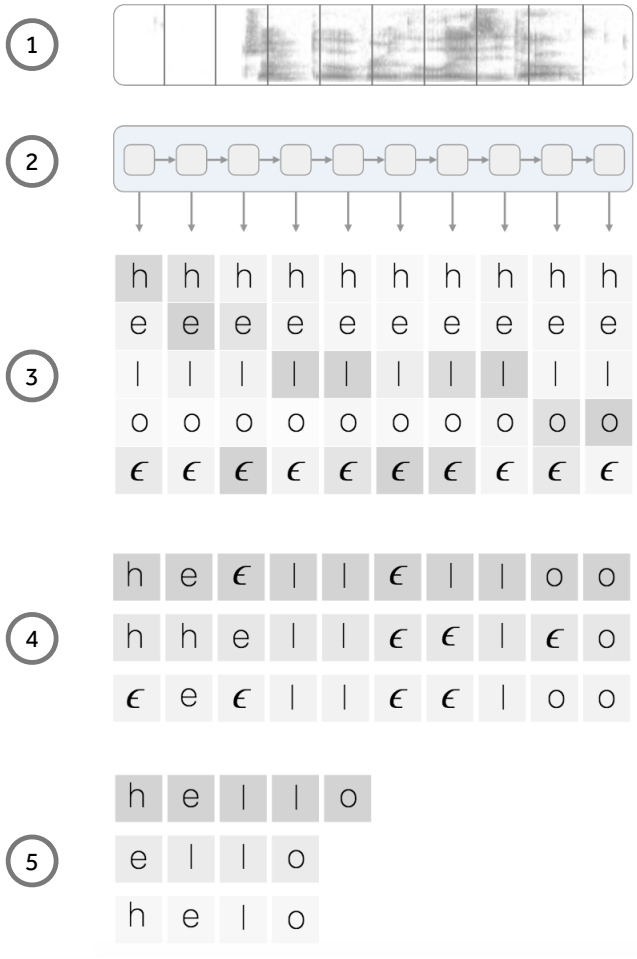

FIGURE 2.12 – CTC (source : Hannun, "Sequence Modeling with CTC", Distill, 2017)

Un alignement est dit valide lorsque la séquence Y correspond à la conversion de la séquence X après application des règles de réduction. Celles-ci consistent à fusionner dans la séquence les symboles identiques qui ne sont pas séparés par un symbole ϵ. Par exemple, [h e ϵ l l ϵ l l o o] et [h h e l l ϵ ϵ l ϵ o] donnent toutes les deux [h e l l o] après réduction. Ainsi, des RNNs simples peuvent être utilisés pour la modélisation de séquences jointes de tailles différentes sans utiliser une architecture encodeur-décodeur. La fonction CTC est particulièrement bien adaptée à la modélisation acoustique, où des séquences de trames acoustiques sont traduites en caractères ou en phonèmes.

2.3 Conclusion

La modélisation de séquences se présente sous deux aspects : la modélisation de séquences simples et la modélisation de séquences jointes. Dans le premier cas, les modèles permettent d'estimer la probabilité d'observer une séquence donnée. Cela peut se traduire par la prédiction de l'élément suivant d'une séquence connue. Dans le deuxième cas, c'est la probabilité d'observer simultanément deux séquences qui est

modélisée. Les modèles peuvent nécessiter un alignement des séquences avant leur apprentissage : lorsqu'il est effectué automatiquement, cela peut conduire à des problèmes de modélisation du fait des erreurs d'alignement. De nouvelles approches neuronales permettent de se passer de cet alignment. Cependant, elles ne sont pas applicables à tous les cas d'usage. En effet, l'architecture encodeur-décodeur nécessite de grandes quantités de données, et la fonction CTC, bien adaptée à la modélisation acoustique, est difficilement compatible avec un mécanisme d'attention. Dans les prochains chapitres, nous allons voir de manière plus approfondie comment ces modèles sont mis en œuvre en reconnaissance et en synthèse de la parole.

Chapitre 3

Reconnaissance automatique de la parole

Sommaire

La reconnaissance automatique de la parole consiste à transcrire un signal de parole en une séquence de mots. Un système réalisant une telle tâche, appelé SRAP, peut aujourd'hui fournir des transcriptions de grande qualité. Cependant, ceci est uniquement possible si le système est adapté au cas d'usage, dans la mesure ou de nombreuses variabilités du langage parlé sont prises en compte. En effet, la prononciation, le vocabulaire et les conditions acoustiques peuvent varier énormément selon le locuteur, la langue, la thématique du discours et son environnement sonore. Diverses techniques permettent de surmonter ces difficultés, mais cela conduit le plus souvent à devoir construire un nouveau SRAP ou adapter un SRAP existant pour chaque nouvelle situation. Par exemple, il est nécessaire de disposer de données de parole transcrite ainsi que de grandes quantités de texte représentatives du domaine d'application, ainsi que des connaissances expertes de la langue traitée. Ces ressources sont difficiles à obtenir, notamment dans un cadre industriel ou pour des langues relativement peu dotées en ressources, ce qui freine le déploiement global des SRAPs. Nous allons dans ce chapitre présenter les principes de base à la contruction de SRAPs, décrire leurs composants essentiels et expliquer la manière de les évaluer. Ainsi, nous comprendrons les efforts qui ont été menés pour accélérer leur développement dans de nouvelles conditions d'utilisation.

3.1 Principes

La reconnaissance de la parole anime des ambitions scientifiques et commerciales depuis de nombreuses années. En effet, chercher à reproduire la capacité humaine à traduire en mots un acte parlé est un enjeu technologique majeur. De plus, de nombreuses applications sont possibles, comme dans les systèmes de dialogues homme-machine, de dictée, de sous-titrage automatique, d'indexation de documents audiovisuels, de classification d'appels téléphoniques, etc. C'est pourquoi la tâche de reconnaissance de la parole a fait l'objet de développements successifs, en particulier des années 1960 à nos jours.

Nous nous plaçons dans le cadre du paradigme de reconnaissance de la parole statistique proposé par (RABINER, 1989), qui est celui le plus largement considéré de nos jours. Ainsi, le problème est de trouver une séquence de mots $W = (w_1, ..., w_N)$ qui maximise la probabilité d'observer une séquence de paramètres acoustiques $X = (x_1, ..., x_T)$ selon des modèles statistiques. Aujourd'hui, il est courant de découper la tâche de reconnaissance de la parole en une chaîne de traitements reposant sur différents modules :

— Un module de segmentation qui extrait du signal les portions contenant de la parole. Ce module peut être complété par un regroupement en locuteurs pour traiter indépendemment les différents intervenants dans le discours.
— Un module d'analyse qui calcule des représentations numériques du signal pertinentes pour le traitement de la parole.
— Un module acoustique qui, à partir des paramètres acoustiques extraits du signal de parole, donne un ensemble d'hypothèses de reconnaissance.
— Un module syntaxique qui intègre des contraintes du langage pour construire la suite de mots la plus probable dans l'espace de recherche construit par le module acoustique.

Le module de segmentation fonctionne différemment selon que le signal est un flux ou un enregistrement déterminé. Dans le premier cas, la reconnaissance de la parole doit être traitée en continu et le signal est découpé en fenêtres d'observations qui sont potentiellement incohérentes en termes de locuteurs ou de contenu linguistique. Dans le second cas, il est possible d'effectuer des opérations plus fines en plusieurs passes. Cette tâche étant un objet d'étude à part entière, nous ne nous étendrons pas dessus dans ce chapitre. Cependant, il est essentiel de garder à l'esprit que ce module conditionne significativement les traitements qui seront effectués ensuite pour déterminer la séquence textuelle.

Le module d'analyse du signal a été discuté en grande partie dans le chapitre 1. Il consiste à convertir le signal en une séquence de vecteurs acoustiques appelés *observations*. En effet, dans le signal de base, beaucoup de données sont superflues pour la reconnaissance de la parole. Il est donc nécessaire d'extraire des paramètres acoustiques pertinents. On distingue ainsi différents coefficients acoustiques, comme les Mel-Frequency Cepstral Coefficient (MFCC) (DAVIS et MERMELSTEIN, 1980) qui sont les plus répandus, mais aussi les coefficients Perceptual Linear Prediction (PLP) (HERMANSKY, 1990). Ces derniers peuvent être enrichis à l'aide de paramètres calculés par différents modèles, afin d'obtenir des SRAPs plus robustes aux changements de locuteurs (feature-space Maximum Likelihood Linear Regression (fMLLR), GALES, 1998) ou minimisant les erreurs de phonèmes (feature-space Minimum Phone Error (fMPE), POVEY et al., 2005).

Ensuite, les modules acoustiques et syntaxiques permettent de calculer la séquence de mots $\hat{W}$ qui maximise la probabilité d'observer X parmi l'ensemble des séquences de mots possibles W^* :

$$\hat{W} = \underset{W \in W^*}{\operatorname{argmax}} P(W|X) \tag{3.1}$$

Or, comme il est difficile de modéliser $P(W|X)$ directement, le théorême de Bayes est généralement appliqué pour obtenir un problème équivalent plus simple à résoudre (JELINEK, 1976) :

$$\hat{W} = \underset{W \in W^*}{\operatorname{argmax}} P(X|W)P(W) \tag{3.2}$$

Il s'agit de traduire la probabilité d'obtenir une séquence de mots W sachant le signal X comme le produit entre (1) la probabilité de générer le signal X avec la séquence de mots W et (2) la probabilité d'observer la séquence de mots W. (1) est estimée à l'aide d'un modèle acoustique, qui est un modèle de séquences jointes, et (2) est estimée par un modèle de langage, qui est un modèle de séquences simples. Ceci explique l'utilisation de modules distincts pour modéliser la composante acoustique et la composante syntaxique du langage.

Dans la plupart des cas, la taille du vocabulaire est trop importante pour pouvoir traiter des séquences de mots directement. C'est pourquoi on préfère alors considérer des unités sous-lexicales qui sont en nombre beaucoup plus restreint. Un module lexical, associant les mots à ces unités, s'ajoute alors au système. Comme nous l'avons vu dans le chapitre 1, l'unité sous-lexicale la plus courante est le phonème. Cependant, il est aussi possible de considérer les lettres elles-mêmes, ou regrouper au sein de mêmes symboles les séquences de lettres les plus fréquentes (SHIBATA et

al., 1999). Dans ce cas, on parle de reconnaissance de la parole à base de caractères. Le choix des unités sous-lexicales dépend des caractéristiques de la langue considérée, selon qu'elle soit phonétique, agglutinative, etc. Nous parlerons de *prononciation* du mot w_n pour la séquence Q_n d'unités sous-lexicales correspondantes.

En notant Q^* l'ensemble des séquences d'unités sous-lexicales possibles, la probabilité $P(X|W)$ est alors donnée par :

$$P(X|W) = \sum_{Q \in Q^*} P(X|Q)P(Q|W) \tag{3.3}$$

La probabilité $P(Q|W)$ de la séquence de phonèmes Q sachant la séquence de mots W est donnée par un modèle de prononciation ou phonétiseur. Généralement, l'ensemble des mots qu'il est possible de reconnaître étant fixé, un dictionnaire de prononciation, aussi appelé lexique, est utilisé. L'équation 3.2 devient donc :

$$\hat{W} = \underset{W \in W^*}{\operatorname{argmax}} \sum_{Q \in Q^*} P(X|Q)P(Q|W)P(W) \tag{3.4}$$

Le dictionnaire de prononciation peut être soit réalisé manuellement, soit généré à l'aide d'un phonétiseur. Dans ce dernier cas, on distingue deux types de phonétiseurs : ceux fondés sur des règles de réécriture et ceux appris sur des données. Les phonétiseurs fondés sur des règles (KAPLAN et KAY, 1994 ; BÉCHET, 2001b) cherchent à la fois à couvrir les règles de prononciation de la langue, mais aussi les exceptions éventuels. Ils demandent donc une bonne connaissance de la langue et des annotations manuelles. Les phonétiseurs appris sur des donnés cherchent à étendre un lexique existant en mettant en œuvre des principes de généralisation. On retrouve ainsi des approches fondées sur des modèles n-grammes joints (GALESCU et ALLEN, 2001 ; GALESCU et ALLEN, 2002 ; BISANI et NEY, 2008), des modèles empruntés à la traduction automatique (LAURENT, DELÉGLISE et MEIGNIER, 2009) ou des modèles neuronaux (RAO et al., 2015 ; YAO et ZWEIG, 2015). Dans tous les cas, les phonétiseurs permettent d'associer à une liste de mots isolés une ou plusieurs prononciations possibles par mot afin de construire un dictionnaire de prononciation.

3.2 Modélisation acoustique

Nous avons vu que la modélisation acoustique cherchait à estimer la probabilité $P(X|W)$ d'observer une séquence de vecteurs acoustiques sachant une séquence de mots. Plus précisemment, ils permettent d'estimer $P(X|Q)$ avec Q la séquence de sous-unités lexicales correspondant à W. Nous nous retrouvons donc dans le cas d'un modèle de séquences jointes comme définit dans le chapitre 2.

En pratique, les unités sous-lexicales sont des phonèmes en contexte, ou triphones. Autrement dit, il s'agit de considérer chaque phonème d'une séquence avec ses voisins gauche et droit afin de prendre en compte la variabilité de prononciation des phonèmes selon leur contexte. Dans la mesure où certains triphones sont peu ou non représentés dans les données d'apprentissage du modèle acoustique, des algorithmes de regroupement sont utilisés pour partager des paramètres entre les différents triphones.

De nombreuses approches ont été proposées pour construire des modèles acoustiques pour la reconnaissance de la parole. Parmi ceux-ci, nous distinguons les modèles fondés sur des HMM, et les modèles composés uniquement d'un réseau de neurones.

3.2.1 Modèles acoustiques fondés sur des modèles de Markov cachés

La reconnaissance de la parole statistique (JELINEK, 1976), et en particulier l'usage de HMM pour la modélisation acoustique (RABINER, 1989), est aujourd'hui considérée comme l'approche conventionnelle pour cette tâche. Les HMMs sont des automates à états finis probabilistes, où la probabilité d'être dans un état dépend uniquement de l'état précédent. Nous avons décrit leur application à la modélisation de séquences jointes dans le chapitre 2. Pour la reconnaissance de la parole, ils sont utilisés pour estimer la probabilité d'émission d'une séquence d'observations acoustiques. Chaque phonème en contexte, ou triphone, est modélisé par un HMM. En composant ces HMMs, il devient possible de modéliser des mots, voire des séquences de mots (figure 3.1). De cette façon, nous obtenons un modèle capable d'estimer la probabilité $P(X|W)$ d'observer une séquence de vecteurs acoustiques sachant une séquence de mots.

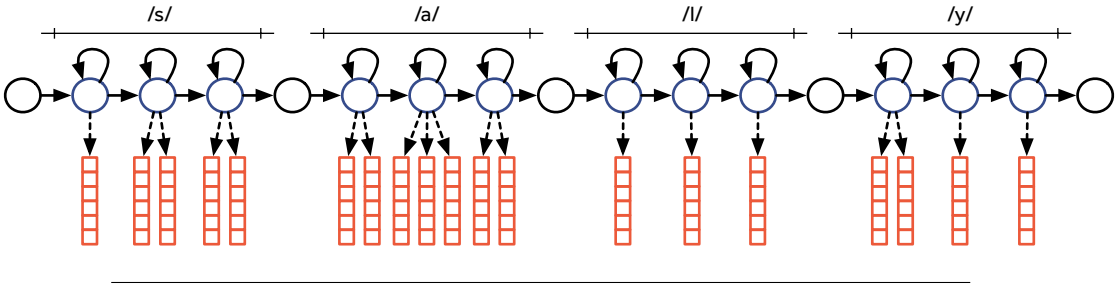

FIGURE 3.1 – Illustration d'un modèle acoustique fondé sur des HMMs avec le mot *salut*.

L'apprentissage consiste à estimer d'une part les probabilités de transition entre les états des HMMs, d'autre part à estimer les fonctions de densité de probabilité d'émission pour chaque état. Ces dernières sont obtenues, sachant une séquence d'observations acoustiques, soit par des GMMs qui associent à chaque état des HMM une densité de probabilité d'émission, soit par un modèle neuronal qui donne une distribution de probabilité sur les états des HMM.

Modèles de mélange de gaussiennes

Les GMM ont été le modèle le plus répandu pendant des années pour estimer les probabilités d'émission des trames acoustiques sur les états d'un HMM pour la reconnaissance de la parole. Le principe est d'associer à chaque état une somme pondérée de densités de probabilité gaussiennes :

$$b_j(x_i) = \sum_{k=1}^{K} w_{j,k} \mathcal{N}(x_i; \mu_{j,k}, \Theta_{j,k}) \tag{3.5}$$

avec $\mathcal{N}(x_i; \mu, \Theta)$ une densité de probabilité gaussienne et $w_{j,k}$ les poids du mélange de gaussiennes. En pratique, K, le nombre de gaussiennes par mélange, est déterminé empiriquement.

Les modèles acoustiques sont entraînés sur des données de parole multi-locuteurs. Afin d'adapter les modèles aux conditions dans lesquelles ils sont exploités, il est important de gérer les variabilités inter-locuteurs. Ceci est réalisé à l'aide de techniques d'adaptation aux locuteurs. L'ensemble des techniques repose sur deux principes : soit modifier les paramètres des mélanges de gaussiennes, à savoir moyenne et variance, pour chaque locuteur, soit modifier les vecteurs acoustiques en prenant en compte les spécificités de chaque locuteur. On retrouve parmi les techniques les plus courantes, souvent cumulées, les transformations Maximum Likelihood Linear Regression (MLLR) (LEGGETTER et WOODLAND, 1995), Constrained Maximum Likelihood Linear Regression (CMLLR) (DIGALAKIS, RTISCHEV et NEUMEYER, 1995), fMLLR (GALES, 1998) et Maximum A Posteriori Estimation (MAP) (GAUVAIN et LEE, 1994). Ces techniques, qui donnent de bonnes améliorations aux approches HMM-GMM, sont difficilement transposables aux approches HMM-DNN utilisées aujourd'hui. Des travaux cherchent à tirer parti de ces transformations en donnant en entrée des DNN des paramètres dérivés des modèles GMM (TOMASHENKO et KHOKHLOV, 2014).

Modèles neuronaux

L'utilisation de modèles neuronaux comme alternative aux GMM pour estimer les probabilités d'émission des états des HMM a été proposée dans les années 1990. De simples MLP (BOURLARD et WELLEKENS, 1987), les modèles ont évolués vers l'utilisation d'architectures plus complexes comme les DNN (HINTON et al., 2012) ou les Time Delay Neural Network (TDNN) (WAIBEL et al., 1989), un type particulier de CNN. Il a été montré récemment que ces derniers permettent d'obtenir des résultats à l'état de l'art (PEDDINTI, POVEY et KHUDANPUR, 2015). Ainsi, les approches neuronales ont progressivement remplacé les GMM pour estimer les probabilités d'émission des états des HMM.

Pour l'apprentissage de ces modèles, un alignement au niveau phonèmes entre la transcription et la source acoustique est requis. Ce dernier est généralement constitué à l'aide d'un SRAP HMM-GMM. Ensuite, le décodage consiste à présenter au modèle une séquence de vecteurs acoustiques pour estimer les distributions de probabilité d'émission des états des HMM pour chaque vecteur acoustique.

3.2.2 Modèles acoustiques neuronaux de bout en bout

Une autre manière de construire des SRAP est d'utiliser simplement un unique réseau de neurone pour convertir un signal acoustique en une séquence de mots. Bien que ces modèles soient communéments considérés comme des SRAP *de bout en bout*, ils ne font généralement pas l'impasse sur des données d'apprentissage segmentées, une extraction de paramètres acoustiques, un dictionnaire de prononciation, et même pour certains un modèle de langage. C'est pourquoi, nous les considérons ici plutôt comme des modèles acoustiques. Cependant, contrairement aux modèles fondés sur des HMM, la modélisation du langage est intrinsèque grâce à l'architecture

encodeur-décodeur. L'usage d'un éventuel modèle de langage externe permet alors de recalculer les poids dans un graphe d'hypothèses de reconnaissance.

Les approches neuronales pour la modélisation acoustiques reposent sur la fonction CTC pour se passer de l'alignement entre signal de parole et séquence d'unités sous-lexicales (GRAVES et JAITLY, 2014). De plus, des couches récurrentes bidirectionnelles permettent de prendre en compte des dépendances à long terme dans la prise de décision. De nombreuses améliorations ont été proposées ensuite, comme le mécanisme d'attention (BAHDANAU, CHO et BENGIO, 2014 ; CHAN et al., 2016), le décodage avec un transducteur fini pondéré (MIAO, GOWAYYED et METZE, 2015) ou l'utilisation de couches convolutionnelles (AMODEI et al., 2016). Le principe est toujours d'avoir en entrée une séquence de vecteurs acoustiques et en sortie l'estimation d'une distribution de probabilité sur les unités sous-lexicales autorisées.

3.3 Modélisation du langage

La modélisation du langage a pour but d'estimer la probabilité $P(W)$ d'observer une certaine séquence de mots W. Autrement dit, il s'agit d'estimer si une séquence de mots donnée vérifie une certaine grammaire propre à la langue considérée. La probabilité d'observer une séquence de mots $W = (w_1, ..., w_T)$ s'exprime comme le produit des probabilités d'observer chaque mot sachant l'historique :

$$P(W) = P(w_1) \prod_{t=2}^{T} P(w_t | w_1, \ldots, w_{t-1}) \tag{3.6}$$

3.3.1 Modèles de langage n-grammes

En reconnaissance de la parole, les modèles de langages sont généralement des modèles n-grammes dont l'ordre est limité entre 2 et 4. La probabilité estimée par un modèle de langage d'ordre n est alors approximée par :

$$P(W) = P(w_1) \prod_{t=2}^{T} P(w_t | w_{t-(n-1)}, \ldots, w_{t-1}) \tag{3.7}$$

Le modèle de langage est estimé en maximisant la vraisemblance d'un corpus textuel d'apprentissage (DEMPSTER, LAIRD et RUBIN, 1977). En notant $C(w_t, ..., w_{t+k})$ le nombre d'occurrences de la séquence $(w_t, ..., w_{t+k})$ dans le corpus d'apprentissage, la probabilité d'observer un mot w_t est estimée par :

$$P(w_t | w_{t-(n-1)}, \ldots, w_{t-1}) = \frac{C(w_{t-(n-1)}, \ldots, w_{t-1}, w_t)}{C(w_{t-(n-1)}, \ldots, w_{t-1})} \tag{3.8}$$

Bien que pour obtenir de bons résultats, les modèles de langages sont estimés sur des textes les plus grands et les plus proches possibles du domaine d'application du SRAP, certains n-grammes peuvent toujours être non observés dans le corpus d'apprentissage et se voir attribuer une probabilité nulle. Pour éviter cela, des techniques de lissage et de repli sont employés pour redistribuer des poids sur les n-grammes

non observés (KATZ, 1987 ; KNESER et NEY, 1995). Ainsi, les modèles de langages obtenus sont suffisamment génériques pour la reconnaissance de la parole.

Pour obtenir des quantités de données suffisamment grandes pour estimer des modèles de langages robustes, il est parfois nécessaire de travailler sur des données relativement éloignées du domaine d'application du SRAP. Ceci induit naturellement un biais : les n-grammes les plus fréquents dans un domaine ne sont pas ceux les plus fréquents dans un autre. Par exemple, un discours politique n'utilisera pas le même vocabulaire qu'un compte-rendu médical. Pour construire des SRAP pour des applications peu dotées en ressources linguistiques, il est possible d'adapter un modèle de langage générique à un domaine particulier. L'adaptation est obtenue en réalisant une interpolation linéaire entre plusieurs modèles de langages. Certains seront appris sur de grandes quantités de données mais d'un domaine éloigné de celui visé, d'autres seront estimés sur des données plus restreintes mais dont le contenu est plus spécifique.

3.3.2 Modèles de langage neuronaux

Nous avons vu dans le chapitre 2 que les réseaux de neurones permettent de modéliser des séquences simples. Les modèles de langages ne font pas exception et peuvent par exemple être constitué d'un MLP (BENGIO et al., 2003 ; SCHWENK, 2007) ou d'un RNN (MIKOLOV et al., 2010). Les modèles de langages neuronaux permettent de prendre en compte un historique plus long dans la séquence de mots. En effet, lorsqu'ils sont construits avec des MLP, le nombre de paramètres augmente linéairement avec l'ordre du modèle. Au contraire, la taille d'un modèle de langage n-grammes augmente exponentiellement à son ordre. De plus, lorsqu'ils sont fondés sur des RNN, la taille de l'historique est indéfinie : les décisions du modèle sont prises en fonction de l'observation courante et des décisions passées. Ainsi, un modèle de langage neuronal permet une gestion de dépendances plus longues. C'est pourquoi ils sont largement utilisés pour réévaluer les hypothèses d'un modèle de langage n-grammes ou d'un modèle acoustique neuronal.

3.4 Évaluation des systèmes de reconnaissance de la parole

Les systèmes de reconnaissance de la parole sont régulièrement évalués au cours de campagnes d'évaluation portant sur différentes thématiques. Les thématiques d'une campagne se veulent proche des problématiques courantes du domaine et vont ainsi en difficulté croissante au cours du temps : les campagnes NIST ont par exemple considéré progressivement de 1988 à 2009 de la parole lue, de la parole conversationnelle, des journaux télévisés et des enregistrements de réunions (PALLETT et al., 2009). Plus récemment, les campagnes MGB Challenge (BELL et al., 2015 ; ALI et al., 2016 ; ALI, VOGEL et RENALS, 2017) sont dédiées à la construction de SRAPs à partir de données d'apprentissage imparfaites, obtenues par sous titrage automatique de journaux télévisés. Les deux dernières éditions ont porté sur la langue arabe et en particulier la gestion des dialectes.

Les campagnes permettent de comparer différents systèmes de manière rigoureuse en proposant des données de test identiques pour tous les candidats. Ces données

sont normalisées, c'est-à-dire qu'un ensemble de traitements est effectué sur les transcriptions afin d'éliminer les difficultés sur lesquelles il n'est pas souhaité que les systèmes soient évalués. Ainsi, les transcriptions sont passées en lettres minuscules, les ponctuations sont supprimées, les nombres sont réécrits en lettres et les abbréviations sont développées. Dans cette thèse, toutes les transcriptions utilisées pour la reconnaissance de la parole font l'objet de ces traitements.

Le critère d'évaluation des SRAPs est le taux de reconnaissance. Il s'agit de minimiser la distance d'édition entre l'hypothèse de reconnaissance et la transcription de référence. Généralement, l'hypothèse de reconnaissance est alignée à la référence à l'aide de l'algorithme de Levenshtein, puis les insertions, omissions et substitutions sont comptabilisées pour obtenir le taux d'erreur mot, noté *WER (Word Error Rate)* :

$$WER = \frac{\#insertions + \#omissions + \#substitutions}{\#mots} \tag{3.9}$$

Le *WER* donne ainsi le rapport entre le nombre de mots erronés et le nombre de mots de la référence.

3.5 Conclusion

Dans ce chapitre, nous avons présenté le principe de la reconnaissance automatique de la parole et des moyens courants pour répondre à cette tâche. Ainsi, nous avons décrit les composants principaux des SRAPs, à savoir le modèle acoustique et le modèle de langage. Ces derniers sont fortement dépendants des conditions d'utilisation du SRAP construit : format des données, langue, locuteurs, vocabulaire, champ lexical. Les modèles acoustiques peuvent être adaptés à de nouveaux locuteurs, voire à de nouvelles langues. Il est également possible d'adapter des modèles de langages à de nouvelles thématiques, en modifiant le vocabulaire et en utilisant des données spécifiques. Il existe également des outils de nettoyage et de normalisation du texte, des outils de phonétisation automatique, qui permettent d'accélérer le développement de nouveaux systèmes et réduire l'expertise linguistique nécessaire. Enfin, les données imparfaites, comme des vidéos accompagnées de leurs sous-titres, sont de plus en plus utilisées pour construire de nouveaux corpus d'apprentissage (Rousseau, Deléglise et Estève, 2012 ; Rousseau, Deléglise et Estève, 2014 ; Hernandez et al., 2018 ; Gretter, 2014 ; Bell et al., 2015 ; Ali et al., 2016 ; Ali, Vogel et Renals, 2017).

Les travaux portant sur la reconnaissance de la parole ont permis de simplifier le développement des modèles pour de nouvelles applications. Nous observons que les ambitions scientifiques actuelles sont de s'affranchir de plus en plus des contraintes liées à la nécessité de posséder de larges quantités de données finement annotées. De telles avancées permettraient de développer rapidemment des SRAPs pour de nouvelles langues sans compromis sur les performances. Cependant, il n'est pas évident de savoir à quel point les annotations linguistiques, comme les transcriptions phonétiques ou les étiquettes grammaticales, sont importantes pour obtenir des SRAPs de grande qualité. De plus, si de telles annotations apportent des améliorations significative, comment les obtenir sans engager d'importants moyens humains ? Nous tenterons de répondre à une partie de ces questions dans cette thèse en travaillant sur la granularité des jeux de phonèmes et la détection automatique des erreurs de

transcriptions phonétiques dans les bases de données d'apprentissage. Ce travail bénéficie également au développement de systèmes de synthèse de la parole, ce qui permet de mutualiser les efforts pour améliorer deux tâches différentes.

Chapitre 4

Synthèse de la parole

Sommaire

La synthèse de la parole a pour objectif de reproduire artificiellement la parole humaine. Autrement dit, elle cherche à doter un objet de la capacité de parler. Au cours du temps, la définition et les motivations de la synthèse de la parole ont beaucoup évolué, de même que les moyens mis en œuvre pour répondre au problème. Des statues parlantes de l'Antiquité prêtant leur voix à des divinités, à la construction de systèmes ludiques imitant voyelles et consonnes, en passant par des outils pour les personnes présentant des pathologies de la parole, la synthèse de la parole est devenue progressivement un moyen de fournir des informations par un système informatique (FLANAGAN, 1972). Aujourd'hui, on parle de synthèse de la parole lorsqu'il s'agit de convertir une entrée textuelle en un signal acoustique. Bien plus que la simple intelligibilité du message, les systèmes actuels cherchent à reproduire les nuances et l'expressivité de la parole humaine, au point de rendre difficile la distinction entre un énoncé produit par une machine et un autre produit par un humain.

Convertir un texte en signal de parole, autrement dit passer d'une représentation symbolique à une représentation acoustique de la parole, nécessite plusieurs traitements successifs. Plus particulièrement, deux étapes principales peuvent être distinguées : l'estimation des paramètres linguistiques à partir du texte, et la génération du signal de parole à partir des descripteurs obtenus à l'étape précédente. Tandis que la première étape est commune à la plupart des méthodes de synthèse de la parole, la deuxième est celle qui distingue les paradigmes existants. La synthèse paramétrique, qui considère des modèles de régression pour générer des paramètres acoustiques, et la synthèse par corpus, qui cherche à sélectionner dans un corpus de parole des unités acoustiques pour les concaténer, sont les deux méthodes dominantes actuellement.

Ce chapitre présente un état de l'art de la synthèse de la parole à partir du texte. Nous allons d'abord présenter la tâche d'estimation des paramètres linguistiques, avant de décrire plus précisément les approches de génération de signal de parole les plus répandues actuellement. Nous mettrons notamment en évidence les éléments prévenant la construction rapide de nouveaux systèmes de synthèse de la parole, et les moyens d'en accélérer le processus de développement. Finalement, nous discuterons de la manière dont les systèmes de synthèse de la parole sont évalués.

4.1 Estimation des paramètres linguistiques

Un texte, soit une simple séquence de caractères, ne contient pas suffisamment d'informations pour décrire toutes les subtilités de la parole. A l'écrit, le sens d'un message est véhiculé d'une part à travers le sens propre des mots choisis, d'autre part grâce à la structure de l'énoncé. Tandis que les mots sont distingués par leur orthographe, la structure du message est portée par la ponctuation ainsi que par le rôle et la fonction de chaque mot. A l'oral, les mots ne sont pas distingués par leur orthographe, mais par leur prononciation. De plus, la structure de l'énoncé est véhiculée par les variations mélodiques et rythmiques de la parole. Or, en terme de données, un texte est uniquement une séquence de caractères où le sens des mots et la signification des symboles de ponctuation est absente. Pour transmettre correctement le sens du message dans sa version vocalisée, il est donc nécessaire de procéder à une analyse du texte pour obtenir un ensemble de descripteurs du texte à synthétiser, qui devront être traduits de manière acoustique.

4.1.1 Normalisation du texte

La première étape de traitement linguistique est la réécriture du texte sous forme oralisée. Les abbréviations et les nombres sont développés de manière à les présenter comme ils sont lus. Par exemple, *Dr* devient *Docteur* et *2019* devient *deux mille dix-neuf*. De plus, les conventions d'écriture sont uniformisées, en particulier la ponctuation et le découpage des mots.

Pour réaliser cette tâche, la méthode la plus simple est d'appliquer des règles de réécriture s'appuyant sur des expressions régulières. Le problème est que les règles ne sont valables que pour une langue donnée. Dans (BIGI, 2011), l'auteur décrit une approche de normalisation multilingue. Il est ainsi possible de mutualiser entre les langues des composants de l'outil de normalisation du texte.

Pour le développement de certains composants, il est cependant toujours nécessaire de faire appel à une personne ayant étudié la langue considérée. Tandis que pour certaines langues, une journée suffit à écrire les règles de normalisation, d'autres nécessitent des semaines de travail. De plus, il est fréquent que les règles de réécriture développées manuellement contiennent des erreurs difficiles à détecter. Afin de s'affranchir de ces limitations, des travaux portant sur des systèmes de normalisation automatique du texte ont été menés ces dernières années (SPROAT et al., 2001 ; SCHWARM et OSTENDORF, 2002 ; HAN et BALDWIN, 2011 ; PENNELL et LIU, 2011 ; LIU et JIANG, 2012 ; YANG et EISENSTEIN, 2013 ; GORMAN et SPROAT, 2016 ; NG, GORMAN et SPROAT, 2017). Les modèles proposés utilisent le plus souvent des automates à états finis ou des réseaux de neurones récurrents.

La normalisation du texte est similaire aux traitements appliqués dans la préparations de données pour la reconnaissance automatique de la parole. Les modèles sont en effet appris sur un corpus de parole transcrite, dont le texte respecte des conventions strictes : les nombres et les abbréviations sont développées, les mots sont écrits en minuscules et la ponctuation est supprimée (GRETTER, 2014). Pour la synthèse de la parole, les majuscules et la ponctuation sont conservées pour les étapes suivantes d'analyse du texte. Ainsi, mutualiser les outils de normalisation de texte entre la reconnaissance et la synthèse de parole est généralement avantageux.

4.1.2 Analyse linguistique

Certains mots, pourtant différents dans leur sens et leur prononciation, peuvent s'écrire de manière identique : ce sont les mots homographes hétérophones. Afin de lever les ambiguïtés liées à l'orthographe des mots, une analyse linguistique est réalisée sur le texte normalisé. Il s'agit principalement de déduire la nature et la fonction de chaque mot en fonction de son contexte et de sa morphologie. Plusieurs niveaux d'analyse sont effectués. Une analyse morpho-lexicale permet de déterminer la classe grammaticale des mots par recherche dans un dictionnaire, par application de règles ou par l'utilisation de modèles entraînés sur des données pré-étiquetées. Ensuite, une analyse syntaxique cherche à structurer l'énoncé en organisant les mots en une structure arborescente selon leur classes grammaticales. LIA_TAGG (BÉCHET, 2001a) pour le français et l'anglais, MADAMIRA (PASHA et al., 2014) pour l'arabe ou TRmorph (CÖLTEKIN, 2010 ; CÖLTEKIN, 2014) pour le turc, en sont des exemples. Il est important de noter que ces outils d'analyse dépendent fortement des études linguistiques existantes et que leur qualité est inégale selon les langues.

4.1.3 Transcription phonétique

L'unité de base généralement admise pour l'analyse de la prononciation d'un texte est le phonème. Pour une langue donnée, c'est une représentation symbolique de la plus petite unité sonore distinctive de la parole, c'est-à-dire permettant de distinguer à l'oral un mot d'un autre. La transcription phonétique d'un texte peut se déterminer mot par mot dans un premier temps, mais la prononciation d'un mot étant influencée par ses voisins, il est nécessaire dans un second temps de prendre en compte le texte dans son ensemble pour déterminer correctement la chaîne phonétique associée.

La méthode la plus simple pour obtenir la séquence de phonèmes associée au texte à synthétiser est la recherche dans un dictionnaire de la prononciation de chaque mot du texte. Dans le dictionnaire est indiqué pour chaque mot une ou plusieurs prononciations possibles et sa classe grammaticale. Lorsque plusieurs prononciations sont indiquées, il est nécessaire d'en choisir une qui guidera la synthèse de la parole. Deux cas se présentent : soit il s'agit de variantes de prononciation du même mot, soit il s'agit de mots différents partageant la même orthographe. Dans le cas de variantes de prononciation du même mot, un paramétrage dépendant du contexte permet de choisir la prononciation adéquate. Dans le cas contraire, la classe grammaticale donnée par l'analyse linguistique précédente départage les variantes de prononciation. Si un mot est absent du dictionnaire, il est alors nécessaire d'appliquer un système de transcription phonétique automatique.

Une approche est d'appliquer des règles de prononciation (Béchet, 2001b). Ces règles indiquent pour chaque groupe de lettres, appelés graphèmes, les séquences de phonèmes possibles. Ensuite, un modèle statistique similaire à un modèle de langage est entraîné à estimer la probabilité d'une séquence de règles donnée. Pour cela, à partir d'un dictionnaire de prononciation, l'ensemble des séquences de règles pouvant produire les prononciations indiquées sont générées. Cette étape est l'occasion de détecter des erreurs humaines lors de la création des règles et du dictionnaire. En effet, s'il n'est pas possible de générer la phonétisation du lexique à partir des règles de la table, soit le lexique comporte une erreur, soit il manque une règle. Le modèle probabiliste est ainsi appris sur les séquences de règles issues du dictionnaire de prononciation. Il permet enfin de générer la prononciation d'un nouveau mot en appliquant l'ensemble des règles possibles pour la séquence de lettres du mot considéré, puis en sélectionnant la séquence de règles la plus probable.

Un autre moyen de déterminer la prononciation d'un mot est d'utiliser des modèles de séquences jointes (Galescu et Allen, 2001 ; Galescu et Allen, 2002 ; Bisani et Ney, 2008). Pour entraîner le modèle, les séquences de lettres et de phonèmes d'un dictionnaire de prononciation sont alignées à l'aide d'un algorithme d'alignement forcé. Un modèle de langage est ensuite appris sur les séquences de couples (graphèmes, phonèmes) obtenus. Pour donner la prononciation d'un nouveau mot, ce modèle déterminera la séquence de couples (graphèmes, phonèmes) la plus probable à partir des couples possibles sachant la séquence de lettres du mot considéré, ce qui revient à donner sa prononciation.

Par ailleurs, des systèmes de transcriptions phonétiques sont inspirés de la traduction automatique. Ils considèrent des séquences de caractères à traduire en séquences de phonèmes (Laurent, Deléglise et Meignier, 2009 ; Rao et al., 2015 ; Yao et Zweig, 2015). Cela permet de prendre en compte le texte globalement au moment de la prise de décision. L'inconvénient de cette dernière méthode est de nécessiter

un corpus de mots transcrits en phonèmes dans leur contexte pour l'apprentissage des modèles, ce qui est plus difficile à obtenir que la prononciation d'une simple liste de mots isolés.

Connaître la chaîne phonétique correspondant à un texte est incomplet : un même phonème a en effet des réalisations acoustiques différentes selon le locuteur, sa position au sein d'un mot et d'une phrase, et ses voisins dans la chaîne phonétique. C'est pourquoi on indique généralement pour chaque phonème les phonèmes voisins gauches et droits et sa position dans le mot. L'ensemble des informations attachées à un phonème peut s'organiser sous la forme d'une représentation vectorielle, où chaque coefficient représente une information.

4.1.4 Prosodie

La prosodie est la représentation formelle des éléments expressifs de la parole. Parmi ceux-ci, on retrouve les tons, le rythme, l'accentuation et l'intonation des sons du langage (DI CRISTO, 2000). Concrètement, chaque phonème est associé à une durée, une fréquence fondamentale (F0) et une intensité. La prosodie est estimée grâce à la nature de chaque phonème et à sa position relative dans le message oralisé. Comme les autres paramètres linguistiques, les paramètres prosodiques sont obtenus à l'aide de règles expertes ou à l'aide de modèles statistiques.

Au cours de la synthèse d'un texte, les informations prosodiques sont utilisées soit pour ajouter des contraintes aux modèles de synthèse paramétrique, soit pour guider la sélection d'unités acoustiques en synthèse par corpus. Cependant, les modèles dédiés à l'estimation de la prosodie à partir de paramètres linguistiques peuvent être assez coûteux en calcul et donc ne sont parfois pas utilisées. En synthèse de la parole par corpus par exemple, la prosodie peut être rendue implicitement en cherchant des unités acoustiques enregistrées dans un contexte proche du texte à synthétiser.

De plus en plus de travaux (ARIK et al., 2017 ; GIBIANSKY et al., 2017 ; PING et al., 2017 ; WANG et al., 2017 ; SHEN et al., 2018) montrent que l'estimation des paramètres linguistiques peut être automatisée à l'aide de réseaux de neurones, tant pour la conversion des lettres en phonèmes que pour l'estimation de la durée et de la fréquence fondamentale de chaque phonème. En outre, la symbolique utilisée tend à se simplifier du fait de l'utilisation de lettres à la place des phonèmes, le passage de l'une à l'autre représentation devenant interne aux modèles utilisés. Nous verrons dans le prochain chapitre comment le principe des modèles neuronaux s'applique à l'estimation des paramètres linguistiques.

4.2 Génération du signal de parole

L'estimation des paramètres linguistiques permet de convertir le texte en une séquence de vecteurs numériques. Chaque vecteur correspond à un phonème enrichi d'informations permettant de guider la génération du signal de parole. De nombreuses méthodes ont été proposées pour réaliser la génération du signal de parole à l'aide de ces informations. Parmi celles-ci, on trouve des approches mettant en œuvre des règles de production, des modèles probabilistes ou encore des algorithmes de recherche dans des bases de données d'unités acoustiques. Nous donnons ici une vue d'ensemble des différents paradigmes de génération du signal en

décrivant pour chacun d'eux leur fonctionnement ainsi que leurs avantages et inconvénients.

4.2.1 Synthèse articulatoire

Historiquement, la synthèse articulatoire fut une des premières méthodes de synthèse de la parole. Elle a pour objectif de simuler par le calcul les éléments physiques en jeu dans la production de parole. Son principe repose sur la modélisation du flux d'air traversant l'appareil vocal ainsi que le déplacement des différents éléments articulatoires comme les lèvres, la langue, la machoire (MERMELSTEIN, 1973 ; STORY, 2009 ; STORY, 2011). Le nombre de paramètres à prendre en compte étant très élevé, de nombreuses approximations sont nécessaires, ce qui produit des voix de synthèse de faible qualité. De plus, il est difficile de concevoir des systèmes fonctionnant en temps réel tant les modèles comportent de calculs complexes. C'est pourquoi cette approche n'est généralement pas utilisée dans les applications où la synthèse de la parole à un rôle purement fonctionnel. Elle forme cependant une activité de recherche toujours active aujourd'hui dans la mesure où elle permet de mieux comprendre les mécanismes en jeu dans la production de parole.

4.2.2 Synthèse par règles

La synthèse de la parole par règles, ou synthèse par formants, traite directement du signal de parole, sans modéliser l'appareil phonatoire. Pour cela, elle s'appuie sur des règles décrivant l'évolution temporelle des principaux formants. Chaque formant correspond à un maximum du spectre de la parole et est décrit à l'aide de trois paramètres : sa fréquence, son amplitude et sa bande passante. Les règles permettent de déterminer, à partir de la séquence de phonèmes, de leurs durées et de la courbe prosodique, les trois ou quatre premiers formants d'un signal et leur évolution pour générer la plupart des sons d'un langage. Ainsi, les systèmes construits reposent sur des modèles simples, conduisant aux premiers systèmes de synthèse de la parole dont l'utilisation en temps réel est possible. La technologie Klattalk et DECTalk en sont des exemples (KLATT, 1982 ; HALLAHAN, 1995). La synthèse obtenue est intelligible mais possède un timbre robotique du fait de la trop grande simplicité des modèles.

4.2.3 Synthèse paramétrique statistique

La synthèse paramétrique statistique repose sur la construction de modèles de régression, dont l'objectif est d'estimer une séquences de paramètres acoustiques à partir d'une séquence de paramètres linguistiques extraite du texte. Ces paramètres acoustiques sont issus de techniques de traitement de signal cherchant à caractériser la parole, comme le sont les coefficients MFCC. Un *voice coder*, communément appelé vocodeur en français, est ensuite alimenté par cette représentation paramétrique de la parole. Son rôle est alors de générer le signal de parole.

Avec l'intérêt grandissant pour la synthèse de la parole, de plus en plus de données adaptées à la construction de nouveaux systèmes sont collectées. Celles-ci sont constituées d'une part des enregistrements de locuteurs prononçant des textes choisis en fonction de leur couverture phonétique, d'autre part des transcriptions de ces

textes en mots et en phonèmes. Les données sont généralement segmentées au niveau des phrases, ou au moins au niveau d'un groupe de souffle, soit un ensemble de mots consécutifs pouvant être prononcés sans respiration. De cette façon sont formés des couples *signal-texte* de taille restreinte relativement au corpus entier, tout en présentant les phonèmes en contexte.

Pour préparer les données en vue de l'apprentissage des modèles statistiques, le texte suit les traitements linguistiques décrits précédemment, tandis que le signal de parole est également l'objet de traitements visant à l'obtention d'une séquence de paramètres acoustiques. Ces traitements acoustiques permettent de représenter le signal dans le domaine spectral à l'aide de coefficients acoustiques calculés selon une échelle de perception de la parole. Ainsi, des coefficients MFCC sont calculés pour chaque trame de signal, d'une durée comprise entre 10 et 25 ms, avec leurs dérivées première et seconde représentant la dynamique du signal. De cette manière, le problème de synthèse de la parole est ramené à celui de convertir une séquence de vecteurs en une autre.

Inspirée de la reconnaissance de la parole, et issue de la synthèse par formants, la synthèse paramétrique statistique utilise des HMM pour modéliser la dynamique spectrale des évènements sonores observés sur les données d'apprentissage. Le système le plus répandu actuellement est HTS (HMM Speech Synthesis System) (ZEN et al., 2007) et fonctionne en deux étapes : l'alignement et la régression. L'étape d'alignement consiste à estimer la durée de chaque phonème pour connaître le nombre de vecteurs acoustiques à produire pour chaque phonème. En effet, un vecteur de paramètres linguistiques est associé à un phonème, et un vecteur de paramètres acoustiques correspond à une trame de signal de durée fixe. Autrement dit, les vecteurs d'informations linguistiques sont dupliqués de telle sorte qu'il y en ait autant que de vecteurs acoustiques à estimer. Ainsi, cette étape permet de lier la représentation symbolique de la parole à sa représentation temporelle. Ensuite, l'étape de régression consiste à estimer un vecteur de paramètres acoustiques pour chaque vecteur de paramètres linguistiques de la séquence. Pour cela, chaque phonème en contexte est modélisé par un HMM, et des arbres de décisions (YOUNG, ODELL et WOODLAND, 1994) ou des DNN permettent d'associer un modèle aux phonèmes en contexte qui n'ont pas été observés dans le corpus d'apprentissage.

Les systèmes construits sur le paradigme de la synthèse de la parole paramétrique statistique présentent de nombreux avantages. En effet, ils sont relativement robustes à la qualité des données de parole disponibles pour l'apprentissage. De plus, ils peuvent être appris avec des locuteurs différents, grâce à des mécanismes d'adaptation aux locuteurs, ce qui leur procure une grande flexibilité. Des contraintes globales sur un segment peuvent être prises en compte, et l'empreinte mémoire des systèmes est assez réduite en pratique. Enfin, l'intelligibilité est reconnue pour être presque garantie, même si les unités phonétiques à synthétiser ne sont pas présentes dans le même contexte au sein du corpus d'apprentissage. Cependant, malgré la rapidité avec laquelle il est possible de construire un système de synthèse paramétrique statistique lorsque des données sont disponibles, les industriels ont longtemps délaissé ce paradigme de synthèse de la parole. En effet, la synthèse paramétrique statistique ne permet pas directement d'obtenir un signal de parole valide, mais est limité par les performances des vocodeurs. De plus, les modèles de prédiction de la prosodie donnent uniquement une tendance globale de l'évolution de celle-ci et ne peuvent pas déterminer des variations subtile de la prosodie. Ceci rend les voix produites peu naturelles, et comme l'illustrent les campagnes d'évaluation

Blizzard Challenge (BLACK et TOKUDA, 2005), les systèmes de synthèse par corpus obtiennent souvent des meilleurs scores.

Des travaux récents ont montré qu'il était possible d'obtenir des vocodeurs générant de la parole perçue comme naturelle en utilisant des architectures complexes de réseaux de neurones (OORD et al., 2016). Ceci donne un regain d'intérêt à la synthèse de parole paramétrique pour des applications industrielles nécessitant des voix de haute qualité. Nous verrons dans un prochain chapitre que des modèles neuronaux de bout en bout succèdent aux systèmes de synthèse par HMM, réduisant d'autant plus l'expertise nécessaire au développement de systèmes de synthèse dans de nouvelles langues.

4.2.4 Synthèse par corpus

La synthèse par sélection et concaténation d'unités acoustiques, ou synthèse par corpus, repose sur l'idée de générer des nouveaux segments de parole à partir de parole préenregistrée. Autrement dit, la parole produite est issue de la sélection d'unités acoustiques dans une base de données qui sont concaténées pour reconstituer le message à synthétiser. La question qui s'est d'abord posée est quelle unité acoustique de base utiliser. Les phonèmes ont naturellement été proposés dès les années 1950 comme candidat pour décrire les unités acoustiques (HARRIS, 1953). Cependant, la variabilité de réalisation acoustique d'un même phonème et les effets de coarticulation sont tels que la parole résultant de cette technique était difficilement intelligible. Finalement, c'est le diphone, composé de la deuxième moitié d'un phonème et de la première moitié du phonème suivant, qui a démontré la meilleure robustesse à la concaténation d'unités acoustiques (PETERSON, WANG et SIVERTSEN, 1958 ; DIXON et MAXEY, 1968) Pour aller plus loin, les séquences de diphones les plus longues sont privilégiées, ce qui revient à concaténer des unités acoustiques de taille variable (SAGISAKA, 1988).

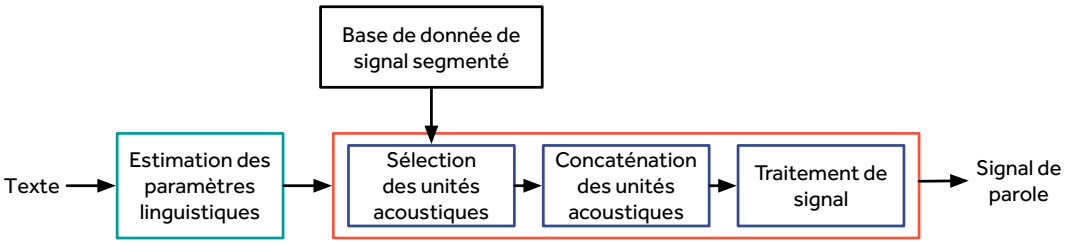

FIGURE 4.1 – Chaîne de traitements d'un système de synthèse de la parole par corpus

Dans (HUNT et BLACK, 1996), les auteurs formalisent la synthèse par corpus comme étant un problème d'optimisation sous contraintes. Lors la génération d'un signal de parole, trois étapes sont nécessaires après l'estimation des paramètres linguistiques à partir du texte (figure 4.1) :

— la sélection des unités acoustiques dans une base de donnée
— la concaténation de ces unités pour former un signal de parole
— le traitement acoustique du signal obtenu

La sélection des unités acoustiques repose sur une fonction de coût définie comme la somme entre un coût cible et un coût de concaténation. Le coût cible évalue la

proximité d'une unité acoustique à la consigne donnée par l'estimation des paramètres linguistiques. Ainsi, il tient compte du contexte phonétique, de la position du phonème dans le mot et dans la phrase, de la présence de frontières prosodiques proches et du style d'énonciation. Le coût de concaténation indique l'importance des effets acoustiques indésirables à la jonction des unités concaténées. C'est ce coût qui va favoriser le choix d'unités consécutives. Il est déterminé à partir du spectre, de la fréquence fondamentale, de la durée et de l'intensité des phonèmes concernés. La recherche de la meilleure séquence d'unités dans le graphe des possibilités se fait ensuite avec un algorithme de décodage *Viterbi* ou *Beam-search*. Une fois que les unités acoustiques sont sélectionnées, il s'agit de les concaténer de la manière la plus lisse possible. La méthode *Pitch Synchronous OverLap-Add (PSOLA)* peut être utilisée (MOULINES et CHARPENTIER, 1990) pour modifier le signal dans le domaine temporel afin d'ajuster la durée et la fréquence fondamentale des trames acoustiques autour des points de concaténation.

Les données utilisées pour la synthèse par corpus sont tout d'abord un ensemble de segments de parole enregistrée, avec les transcriptions en mots associées. Ces transcriptions textuelles sont converties en phonèmes comme nous l'avons vu en décrivant les traitements linguistiques effectués sur le texte. Un algorithme d'alignement forcé permet ensuite de faire correspondre chaque phonème obtenu à une séquence d'échantillons de signal. Nous obtenons alors les frontières des diphones, soit le découpage en unités acoustiques du signal de parole. La transcription phonétique doit donc être la plus précise possible dans la mesure ou c'est elle qui repère les unités acoustiques disponibles. Si une unité acoustique est mal annotée, soit elle ne sera jamais sélectionnée, soit elle sera sélectionnée dans des cas inappropriés. Ceci implique donc une validation manuelle de la transcription phonétique automatique, c'est-à-dire une vérification des phonèmes et une correction lorsque cela est nécessaire.

Lors de la création d'une nouvelle voix, une autre difficulté est de savoir combien de segments de parole doivent être enregistrés. En effet, plusieurs représentants par diphone doivent être pris en compte, afin de couvrir une grande variété de contextes d'élocution. De plus, pour gagner en naturel, les unités acoustiques sélectionnées sont les plus longues possibles, en privilégiant les séquences de diphones consécutifs. Pour accélérer le processus de création d'une voix, il est possible de minimiser les segments à enregistrer en optimisant la couverture phonétique de chaque segment et en se limitant à un nombre réduit de ces derniers. Les limites de cette réduction résident dans le risque d'utiliser des unités acoustiques dans des contextes inadaptés lorsque les unités acoustiques adaptées sont absentes du corpus enregistré. Par exemple, si dans une phrase synthétisée, l'avant-dernier diphone est issu d'un milieu de phrase et qu'il est concaténé à un diphone de fin de phrase, la prosodie est fortement affectée et la parole semblera peu naturelle. La combinatoire devient donc rapidement très importante lorsqu'un corpus de parole doit contenir de multiples représentants des séquences de diphones les plus courantes (SAGISAKA, 1988 ; BLACK, 1994 ; HUNT et BLACK, 1996).

La synthèse par corpus possède l'avantage de produire de la parole naturelle dans la mesure où les éléments composants le signal généré ont réellement été prononcés. De plus, ils offrent des possibilités de contrôle précis par leur dépendance forte aux

paramètres linguistiques. Cependant, les points de concaténation d'unités acoustiques sont parfois le lieu de phénomènes acoustiques indésirables, comme des variations soudaines de la fréquence fondamentale du signal ou du rythme d'élocution. Pour limiter ces problèmes, les locuteurs enregistrant les bases de données de parole doivent maintenir une diction la plus constante possible. Des travaux récents (ESPIC et al., 2018 ; JIANG et al., 2018), cherchent à s'affranchir de ces limites en concevant des systèmes hybrides entre la synthèse par corpus et la synthèse paramétrique. Plutôt que de se référer à des consignes linguistiques strictes, la sélection d'unités à concaténer peut en effet être guidée par une séquence de paramètres acoustiques estimée par un modèle statistique. Il est alors possible d'utiliser un corpus de parole avec une prosodie plus variée, comme des livres audios, pour aller vers des voix plus expressives.

4.3 Évaluation des systèmes de synthèse de la parole

Comme nous l'avons évoqué précédemment, la qualité d'un système de synthèse de la parole réside dans sa capacité à transmettre le sens d'un message textuel, non seulement par la diction de la bonne séquence de mots, mais aussi avec les nuances, le rythme et l'intonation adaptés. Nous comprenons alors que pour qu'elle soit complète, une évaluation est nécessairement subjective. En effet, le sens et l'expression d'un message vocal restent des notions qui ne peuvent pas être interprétées par un outil d'évaluation automatique. Ainsi, des exemples de sorties produites par le système évalué doivent être écoutés par différentes personnes afin d'obtenir un score fiable. Ceci représente un processus potentiellement long et coûteux, qu'il peut être difficile de mettre en place dans le cadre du développement rapide de nouveaux systèmes. Une autre manière de mesurer la qualité d'un système de synthèse de la parole est de procéder à une évaluation objective, qui peut être réalisée systématiquement. Cependant, elle ne peut pas remplacer totalement les tests subjectifs qui restent indispensables dans de nombreux cas d'usage.

Les tests subjectifs sont conduits en faisant écouter plusieurs exemples de sorties des systèmes de synthèse à évaluer à un panel d'évaluateurs. Deux types de tests existent :

— les tests de préférence, où il s'agit de classer les exemples les uns par rapport aux autres (AB, ABX)
— les tests de notation, où il s'agit de donner pour chaque exemple un score pour différents critères (MOS, DMOS, MUSHRA)

Le test AB consiste à présenter aux évaluateurs des paires d'exemples en leur demandant d'indiquer lequel traduit le plus une certaine propriété. Ce peut être "quel exemple préférez-vous ?" ou "lequel semble plus triste ?" par exemple. Le test ABX est une variante du test AB où les évaluateurs sont invités à indiquer lequel des exemples présentés est le plus proche d'un exemple de référence selon certains critères (similitude, émotion, etc.). Concernant les tests de notation, le score MOS est couramment utilisé pour analyser les sorties de systèmes de synthèse de la parole dans de nombreux aspects. La campagne d'évaluation Blizzard Challenge (KING et al., 2018) utilise par exemple le score MOS pour évaluer les systèmes présentés par les candidats. Un score de 1 (mauvais) à 5 (excellent) est attribué à chaque exemple dans différentes dimensions (effort d'écoute, intelligibilité, similarité à une voix naturelle, qualité, rythme, intonation, etc.). Le test DMOS est une variante de MOS où

c'est l'écart à une référence donnée qui est évalué. Le test MUSHRA permet de comparer de multiples exemples simultanément avec une notation sur une échelle de 1 à 100. Comparativement au test MOS, le test MUSHRA permet d'obtenir des résultats statistiquement significatifs avec relativement moins de participants, mais nécessite un temps d'évaluation plus long.

Le problème de l'évaluation subjective est qu'elle est nécessairement conduite sur peu d'exemples, et n'est donc pas toujours représentative des qualités des systèmes évalués. En effet, il est difficile de réunir un grand nombre d'évaluateurs, et chaque participant ne peut pas écouter de nombreux exemples sur une longue durée. Des métriques d'évaluation objectives ont été proposées pour l'évaluation de signaux de parole, mais sont peu utilisées dans le domaine de la synthèse de la parole : PESQ (RIX et al., 2001) ou POLQA (BEERENDS et al., 2013) par exemple. Une autre manière d'évaluer objectivement les systèmes de synthèse de la parole est de se limiter à la consigne phonétique en calculant le PER, soit la distance de Levenstein entre la séquence de phonèmes estimée et la séquence de phonèmes attendues. Dans (CHEVELU et al., 2015), les auteurs proposent un test de similarité donnant un coût d'alignement automatique, puis effectuent un test AB subjectif sur les exemples présentant une forte divergence. Ils montrent ainsi qu'en se concentrant sur des exemples permettant réellement de discriminer les systèmes évalués, les résultats sont plus significatifs qu'une même évaluation sur des exemples aléatoires, tout en réduisant le nombre de tests perceptifs requis.

4.4 Conclusion

Les systèmes de synthèse de la parole permettent aujourd'hui de vocaliser un texte de manière intelligible et de façon de plus en plus naturelle. De plus, leur fonctionnement en temps réel donne la possibilité de construire des systèmes de dialogue homme-machine. Cependant, leur diffusion à grande échelle est limitée car ils reposent d'une part des connaissances spécifiques aux langues développées, et d'autre part nécessitent une collecte de données coûteuse. Les approches récentes mettant en œuvre des réseaux de neurones sur toute la chaîne de traitement proposent de simplifier les paramètres linguistiques nécessaires et de tirer parti de données de qualité variable. Nous verrons dans un prochain chapitre que ces modèles neuronaux permettent d'accéler la construction de systèmes de synthèse de la parole tout en améliorant leurs performances.

Deuxième partie

Contributions

Chapitre 5

Reconnaissance automatique de la parole et données imparfaites : cas de la langue arabe dans les émissions télévisées

Sommaire

LES systèmes de reconnaissance de la parole ont de nombreuses applications. Parmi celles-ci, on trouve le sous-titrage de documents audiovisuels, qui peut être partiellement assisté. Il s'agit soit de transcrire automatiquement la parole et de synchroniser les transcriptions avec la vidéo, soit d'aligner automatiquement une transcription manuelle. Généralement, c'est cette dernière solution qui est retenue dans la mesure où la tâche de sous-titrage comporte des contraintes spécifiques difficiles à prendre en compte pour les SRAP. Par exemple, les textes affichés ne doivent pas prendre trop de place sur l'écran, ce qui implique d'effectuer des approximations par rapport à ce qui est réellement prononcé. De plus, le sens du discours ne doit pas être modifié par des erreurs de transcription, ces dernières répondant notamment à une réglementation précise sur l'accessibilité des documents aux personnes sourdes et malentendantes. Autrement dit, les erreurs des SRAP ne sont tolérées que partiellement : il faut donc reprendre l'annotation manuellement si des transcriptions automatiques sont exploitées. Un sous-titrage revient alors souvent à un alignement automatique d'une transcription approximative manuelle ou d'une transcription automatique corrigée manuellement.

De plus en plus de documents audiovisuels sous-titrés sont rendus disponibles par les chaînes de télévision, notamment sur le Web. C'est une véritable opportunité pour construire des SRAP pour de nouvelles langues, en utilisant les données disponibles pour l'apprentissage des modèles. Cependant, comme les transcriptions sont approximatives d'une part et que l'alignement peut comporter des erreurs d'autre part, les données sont qualifiées d'imparfaites pour la reconnaissance de la parole. Nous présentons dans ce chapitre une méthode pour traiter de telles données imparfaites afin de construire un SRAP performant. Ce travail s'inscrit dans le cadre de notre participation à la deuxième édition de la campagne d'évaluation Multi-Genre Brodcast (MGB) Challenge, pour la tâche de reconnaissance de la parole en arabe.

5.1 Contexte de l'étude

5.1.1 Présentation de la campagne d'évaluation Multi-Genre Broadcast Challenge

Le Multi-Genre Broadcast (MGB) Challenge est une campagne d'évaluation portant sur différentes tâches : la reconnaissance de la parole, la segmentation et le regroupement en locuteurs et l'alignement semi-supervisé. La deuxième édition, organisée par le *Qatar Computing Research Institute (QCRI)*, est centrée sur la langue arabe. Les participants à la campagne d'évaluation sont contraints d'utiliser uniquement les données fournies par les organisateurs pour l'apprentissage des modèles, et les systèmes proposés sont tous évalués sur les mêmes données. Notre participation à la campagne s'est limitée à la tâche de reconnaissance de la parole. L'objectif est de travailler sur une tâche difficile pour la reconnaissance de la parole, à savoir exploiter pour l'apprentissage des données imparfaites, se confronter à une langue que l'on ne maitrise pas et développer un système robuste adapté à des conditions réelles d'utilisation. Plus de détails sur la campagne sont données dans (ALI et al., 2016).

5.1.2 Description des données

Les données de la campagne d'évaluation sont issues d'émissions de la chaîne de télévision Al Jazeera et du site web *Aljazeera.net*. La plupart des émissions sont en arabe standard (70%), mais une part des enregistrements de parole comporte des dialectes régionaux. Les données sont composées d'enregistrements acoustiques accompagnés de transcriptions textuelles, de dictionnaires de prononciations, et d'un grand corpus de textes.

Les données audio correspondent à plus de 2000 émissions de la chaîne de télévision Al Jazeera enregistrées entre 2005 et 2015. Les thématiques abordées sont principalement la politique (76%), la société (9%) et l'économie (8%). Les enregistrements sont divisés en trois parties : une pour l'apprentissage des modèles acoustiques (TRAIN), une pour leur évaluation en cours de développement (DEV), et une autre pour l'évaluation finale des systèmes (TEST). Seules les transcriptions pour TRAIN et DEV sont fournies, le corpus TEST étant réservé à l'évaluation des différentes participations par les organisateurs de la campagne. Le tableau suivant récapitule la répartition des données fournies entre TRAIN, DEV et TEST :

Corpus	Émissions	Durée (h)	Segments	Mots
TRAIN	2 214	1 128,0	376 011	7 815 566
DEV	16	8,5	4 940	57 647
TEST	17	10,1	N/A	N/A

TABLE 5.1 – Répartition des données audio de la campagne d'évaluation MGB Challenge.

Les émissions ont été transcrites manuellement sans respecter fidèlement le contenu acoustique : il y a des paraphrases, des répétitions supprimées, et des résumés lorsque plusieurs locuteurs se sont exprimés. Les transcriptions ont été avant tout conçues pour être consultées indépendemment de la vidéo lors de leur publication sur le web. Pour constituer un corpus d'apprentissage pour la reconnaissance de la parole, elles ont été alignées automatiquement à l'aide d'un SRAP du QCRI. Ensuite, certaines émissions de 2015 ont été mises de côté pour former le corpus DEV et TEST. Ces dernières ont été restranscrites fidèlement pour fournir des données d'évaluation. Lors de cette retranscription manuelle, les organisateurs ont observé un WER de 5% sur les transcriptions manuelles imparfaites.

De plus, des données exclusivement textuelles sont fournies dans l'objectif d'estimer des modèles de langage. Ces dernières sont issues d'articles parus sur le site web *Aljazeera.net* entre 2004 et 2011. Ce corpus est ainsi constitué de plus de 4,7 millions de séquences de mots, principalement des phrases. On compte 122 millions d'occurrences de mots, pour 1,4 millions de mots distincts. Si on filtre les mots n'ayant qu'une occurrence dans le corpus (hapax), on obtient un total de 526 000 mots distincts.

En plus des données audio et textuelles, deux dictionnaires de prononciation sont fournis. L'un est un dictionnaire graphémique, tandis que l'autre est un dictionnaire phonétique. Le dictionnaire graphémique est une liste de plus de 950 000 mots, où chaque lettre des mots est considérée comme un phonème. La plupart des voyelles ne sont donc pas indiquées, conformément à la graphie courante de l'arabe. Le dictionnaire phonétique correspond quant à lui à une phonétisation automatique de la liste des 526 000 mots du corpus de modélisation de la langue après filtrage des

hapax (ALI et al., 2014). Cependant, les participants sont libres d'utiliser un autre lexique de leur choix.

La graphie de la langue arabe étant difficile à manipuler pour des personnes non initiées, les données textuelles sont translittérées grâce à l'alphabet de Buckwalter (BUCKWALTER, 2002), qui utilise des caractères latins et des symboles informatiques. L'évaluation des modèles sera faite selon cette translittération.

5.1.3 Spécificités de la langue arabe

L'arabe est une langue afro-asiatique de la famille des langues sémitiques. Parlée par plus de 300 millions de locuteurs, cela en fait la langue sémitique la plus parlée. Il existe cependant une distinction importante entre l'arabe littéral, constitué de l'arabe classique et de l'arabe standard moderne (MSA) utilisé surtout à l'écrit, et l'arabe dialectal, avec de nombreuses variantes régionales privilégiées à l'oral. Étant reconnu dans tout le monde arabe, l'arabe standard est largement utilisé à l'oral dans les programmes de télévision, bien qu'il ne soit la langue maternelle d'aucun locuteur.

La prononciation de l'arabe est centrée sur les consonnes, qui sont au nombre de 28, tandis qu'on dénombre seulement trois voyelles qui se déclinent en une variante courte et une variante longue. D'autres voyelles peuvent être ajoutées selon les dialectes mais elles restent tout de même minoritaires. Cette importance des consonnes se retrouve à l'écrit, où les voyelles sont représentées principalement par des diacritiques. De plus, ces dernières sont souvent omises, les lecteurs déduisant le sens des mots selon leur contexte et donc leur prononciation. C'est le cas dans les transcriptions du MGB Challenge, où une même graphie peut correspondre à une multitude de mots différents avec une prononciation différente. Cependant, lorsqu'un mot est écrit avec tous ses diacritiques, la prononciation est sans ambiguïté : nous considérerons dans notre étude la voyellation automatique des mots pour en obtenir la prononciation.

Pour finir, l'arabe est une langue riche morphologiquement. C'est en particulier une langue flexionnelle, c'est-à-dire que les mots changent de forme selon leur nature grammaticale. Cette particularité entraîne un nombre important de formes lexicales, ce qui est problématique dans le cadre d'un SRAP où la taille du vocabulaire est limitée.

5.1.4 Stratégie de participation

Notre participation au MGB Challenge s'est faite autour de trois axes : la construction d'un dictionnaire de prononciation prenant en compte les spécificités de l'arabe, l'amélioration du corpus d'apprentissage à partir des données imparfaites fournies, et l'adaptation des modèles au domaine d'application. Ce dernier point regroupe l'adaptation acoustique aux locuteurs et la modélisation du langage. La description du système proposé est détaillée dans (TOMASHENKO et al., 2016). S'agissant d'un travail collaboratif, la description de l'ensemble des parties est nécessaire à la compréhension du système proposé. Cependant, la contribution de ce travail de thèse se limite à la construction d'un nouveau dictionnaire de prononciation.

Des études sur la reconnaissance de la parole en arabe ont montré que des SRAP fondés sur des lexiques graphémiques et phonétiques sont complémentaires (NGUYEN et al., 2009 ; SOLTAU et al., 2009). Cependant le lexique phonétique fourni par les organisateurs ne renseigne pas la forme voyellée de la graphie des mots, ce qui conduit à un grand nombre d'hypothèses de prononciation par mot. La figure 5.1 donne la distribution des mots du lexique fourni pour la campagne en fonction du nombre de prononciations par mot.

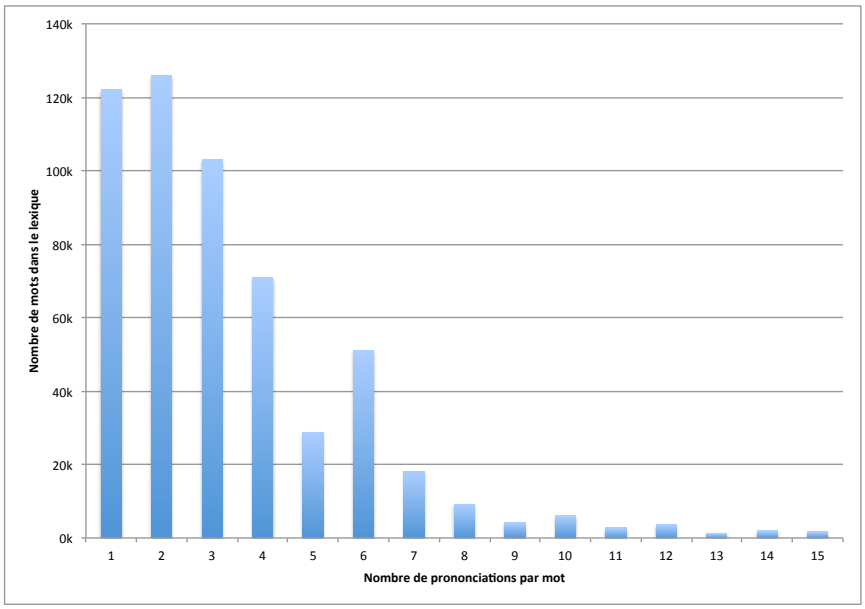

FIGURE 5.1 – Distribution des mots du lexique fourni pour la campagne en fonction du nombre de prononciations par mot

Nous observons que près de 50% des mots du lexique ont au moins trois variantes de prononciation, ce qui est pénalisant pour l'apprentissage des modèles acoustiques. De plus, le lexique a été construit indépendamment des données pour la modélisation acoustique. Il y a donc un nombre important de mots présents dans les transcriptions de parole dont on ne connait pas la prononciation. On compte ainsi seulement 168 000 mots des 526 000 mots du lexique qui sont présents dans les transcriptions du corpus d'apprentissage, et 35 000 mots du corpus d'apprentissage ne sont pas dans le lexique. Nous nous proposons alors de construire un lexique pour l'apprentissage des modèles acoustiques qui soit voyellé et où tous les mots seront présents. Les transcriptions seront modifiées en conséquence pour inclure les différents diacritiques. Pour le décodage, comme le corpus DEV et le corpus TEST ont des transcriptions non voyellées, nous considérons un lexique non voyellé mais dont la prononciation aura été déduite de la forme voyellée des mots.

Comme nous l'avons vu, les données fournies pour l'apprentissage des modèles acoustiques sont imparfaites dans la mesure où les transcriptions ne reflètent pas fidèlement ce qui est réellement prononcé. Pour aligner ces transcriptions manuelles avec le signal, les organisateurs du MGB Challenge ont utilisé un SRAP développé par le QCRI. Cela leur a permis de calculer entre autres le Matching Error Rate (MER) pour chaque enregistrement. Cette métrique, calculée sur le même principe que le Word Error Rate (WER), indique le taux de différence entre l'hypothèse de reconnaissance et la transcription alignée. La figure 5.2 donne le MER au niveau mot et

au niveau caractères en fonction de la quantité en heures de parole du corpus d'apprentissage.

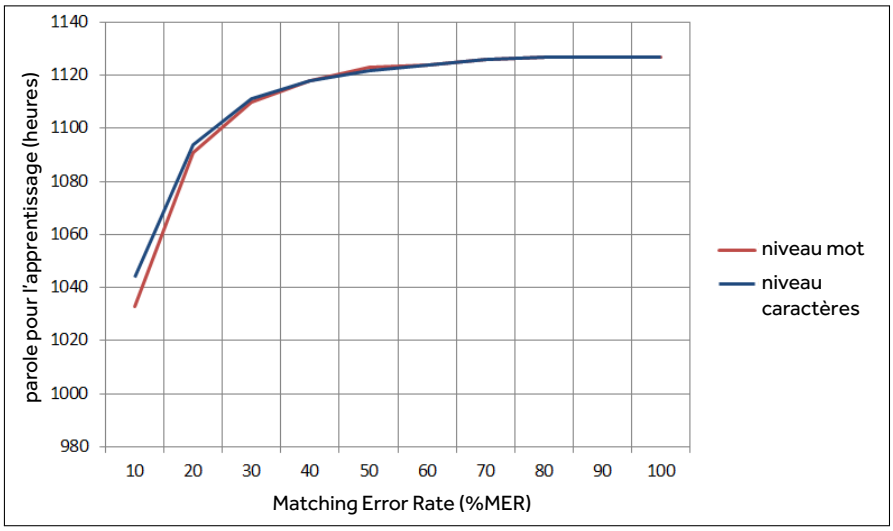

FIGURE 5.2 – *Matching Error Rate* au niveau mot et au niveau caractères des données fournies pour le MGB Challenge en fonction du nombre d'heures de parole

Nous observons que la différence entre transcription de référence et reconnaissance automatique de la parole devient assez importante lorsque l'ensemble du corpus d'apprentissage est considéré. Une telle divergence s'explique soit par des erreurs de reconnaissance, soit pas des différences entre la transcription manuelle et ce qui est prononcé. Cela peut pénaliser l'apprentissage des modèles acoustiques, c'est pourquoi nous nous intéressons à un mécanisme de sélection des données. L'idée est de réduire itérativement la taille du corpus d'apprentissage en sélectionnant les données les plus en accord avec un SRAP initial. Un tel mécanisme a déjà été utilisé pour construire le corpus TEDLIUM en langue anglaise (ROUSSEAU, DELÉGLISE et ESTÈVE, 2012).

Afin d'obtenir un SRAP adapté au domaine d'utilisation, nous emploieront des techniques d'adaptation aux locuteurs pour tenir compte de leurs différences acoustiques. S'agissant d'émissions télévisées d'une période continue, nous pouvons supposer que certains locuteurs seront présents aussi bien dans les données d'apprentissage que dans les données d'évaluation. Concernant la modélisation du langage, nous utiliserons les données uniquement textuelles pour construire un modèle généraliste, puis nous le spécialiserons grâce aux transcriptions des émissions. Le vocabulaire sera choisi selon un compromis entre le nombre de mots différents et la couverture des transcriptions fournies. Ce même vocabulaire sera celui du dictionnaire de prononciation utilisé pour le décodage.

5.2 Voyellation du texte et phonétisation en contexte

5.2.1 Principe

L'alphabet arabe comporte deux sortes de représentations : les lettres, qui sont toujours écrites, et les diacritiques, représentant les voyelles courtes, souvent omis. Il en résulte que l'orthographe d'un mot sorti de son contexte ne permet pas de distinguer deux prononciations différentes (et potentiellement deux sens différents). La construction d'un dictionnaire de prononciation, aussi appelé lexique, à partir de la représentation graphémique des mots n'est donc pas un exercice trivial. Cependant, lorsque les mots sont voyellés, c'est-à-dire que tous les diacritiques sont écrits, des règles simples permettent d'en déduire la séquence de phonèmes.

Plusieurs méthodes permettraient d'obtenir un dictionnaire phonétique :

1. voyeller les mots dans leur contexte et conserver pour le lexique les formes voyellées les plus fréquentes
2. générer un lexique à l'aide d'un modèle statistique, puis sélectionner les entrées les plus vraisemblables à l'aide d'un modèle acoustique
3. voyeller le texte en différenciant les mots selon leur voyellation (e.g. deux occurrences de *ktb* peuvent devenir *ktb_kataba* et *ktb_kutubN* dans le texte)

Pour générer le dictionnaire phonétique proposé, les organisateurs ont retenu la méthode (1) en se servant de l'outil MADAMIRA (PASHA et al., 2014). MADAMIRA est un outil capable de générer des hypothèses de voyellation pour chaque mot d'après son contexte lexical. Le procédé de génération des hypothèses repose sur un modèle stochastique, et un score de confiance est donné pour chaque voyellation proposée. Les organisateurs du MGB Challenge ont ainsi gardé les trois meilleures hypothèses de voyellation de MADAMIRA lorsque le score de confiance était au moins à 80% du meilleur score, puis ont déduit la phonétisation en appliquant les règles décrites dans (BIADSY, HABASH et HIRSCHBERG, 2009). La méthode (2) a été appliquée avec succès dans (ZHANG et al., 2017) pour l'anglais. Les auteurs ont en effet montré qu'il était possible d'obtenir des performances proches d'un lexique de grande qualité construit manuellement lorsque des données de parole transcrites et un petit lexique initial étaient disponibles. Un modèle acoustique est utilisé pour filtrer les entrées d'un lexique qui ne sont pas ou peu utilisées lors d'un alignement forcé. Pour la campagne, nous choisissons la méthode (3) en expérimentant la conservation de la première ou des deux premières hypothèses de MADAMIRA. Cette voyellation est suivie d'une étape de phonétisation où des règles de translitteration sont appliquées.

Une fois la voyellation terminée, le processus de phonétisation se fait aisément avec des règles simples de translitteration. Pour leur lexique, les organisateurs du MGB Challenge se fondent sur des règles décrites dans (BIADSY, HABASH et HIRSCHBERG, 2009), en ajoutant une règle : les mots se terminant par une voyelle ont une variante de prononciation supplémentaire où la dernière voyelle n'est pas prononcée (ALI et al., 2014). En effet, ils ont remarqué que les locuteurs réalisent souvent cette élision. Le lexique final est construit en faisant correspondre les phonétisations obtenues avec les mots sous forme graphémique. Dans notre étude, nous suivons cette méthode de phonétisation en comparant les règles et les jeux de phonèmes décrits dans (ALI et al., 2009) et (BIADSY, HABASH et HIRSCHBERG, 2009), respectivement notés dans la suite *règles (phonèmes) A* et *règles (phonèmes) B*.

5.2.2 Voyellation des données textuelles

Les données voyellées correspondent aux transcriptions du corpus TRAIN et du corpus DEV, ainsi que le texte fourni pour l'apprentissage du modèle de langage. La voyellation consiste à analyser les textes à l'aide de MADAMIRA et à extraire le texte ainsi voyellé. Parmi les informations fournies par MADAMIRA lors de l'analyse d'un texte, nous trouvons pour chaque mot des voyellations possibles, ordonnées selon un score de confiance. Nous avons choisi de conserver la ou les deux meilleures hypothèses, et d'écrire un mot voyellé dans le format `<écriture sans voyelle>_<écriture voyellée 1>+<écriture voyellée 2>`. Par exemple, le mot *ktb* pourra s'écrire `ktb_kataba` ou encore `ktb_kutuba+kutubN`. Etant donné la nature contextuelle de la voyellation, nous espérons récupérer l'ensemble des voyellations possibles d'une même orthographe par recoupement de plusieurs occurrences d'un même mot.

Une particularité des textes du MGB Challenge est qu'ils présentent des nombres écrits en chiffres. Ceci est problématique dans la mesure où la phonétisation d'un mot est déduite de son écriture en lettres. Une difficulté de la transformation des nombre de l'écriture en chiffres à l'écriture en lettres réside dans le fait que les nombres en arabe s'accordent en genre, masculin et féminin. Or, il est d'une part difficile de déterminer le genre des nombres d'après leur contexte d'utilisation, et d'autre part le choix d'un accord influence la voyellation des mots du même contexte lexical. En effet, certains noms masculins et féminins ne se différencient que par leurs diacritiques. Pour traiter cette difficulté, nous avons choisi de voyeller tous les textes trois fois : avec les nombres en chiffres, avec les nombres au masculin et avec les nombres au féminin. Tandis que la voyellation des mots qui ne sont pas des nombres est conservée de la version où les nombres sont en chiffres, la voyellation des nombres résulte de la concaténation des voyellations des nombres au masculin et au féminin. Cela permet de couvrir tous les cas de voyellation des nombres sans influencer négativement la voyellation du contexte lexical de ces mots.

5.2.3 Phonétisation

Une fois la voyellation des textes terminée, il s'agit de construire un lexique phonétique. Plus particulièrement, nous distinguons lexique d'apprentissage et lexique de décodage.

Le lexique d'apprentissage est utilisé lors de l'apprentissage des modèles acoustiques. Son objectif est de proposer pour chaque mot du corpus TRAIN la phonétisation correspondant au signal de parole. Dans les faits, nous nous contentons d'indiquer un ensemble de phonétisations possibles, dérivées des hypothèses de voyellation. Il s'agit donc d'un compromis entre une phonétisation figée du corpus d'apprentissage et une grande variabilité dans les hypothèses de prononciation. Par exemple pour *ktb*, nous aurons dans le lexique :

```
ktb_kataba  k a t a b
ktb_kataba  k a t a b a
ktb_kutuba  k u t u b
ktb_kutuba  k u t u b a
ktb_kutubi  k u t u b
ktb_kutubi  k u t u b i
```

```
ktb_kutubK  k u t u b i n
ktb_kutub   k u t u b
ktb_kutubN  k u t u b u n
```

Ainsi, chaque entrée est identifiée par un mot voyellé en contexte, accompagné des phonétisations possibles. Ceci implique que les mots soient transcrits selon le même format dans le texte du corpus d'apprentissage. Cela est possible pour le corpus TRAIN, mais pas pour DEV ou TEST où le SRAP proposé pour la campagne se doit de respecter le formalisme des organisateurs. C'est pourquoi nous développons un second lexique, spécifique au décodage.

Le lexique de décodage est utilisé lors de la transcription de nouvelles données audio. Il donne alors l'orthographe des mots tels qu'ils doivent être obtenus en sortie du système, avec les différentes phonétisations associées. Ainsi, par rapport au lexique d'apprentissage les entrées ne sont pas voyellées. Par exemple pour *ktb* :

```
ktb  k a t a b
ktb  k a t a b a
ktb  k u t u b
ktb  k u t u b a
ktb  k u t u b i
ktb  k u t u b i n
ktb  k u t u b u n
```

Il s'agit simplement du lexique d'apprentissage auquel nous avons retiré les marques de voyellation. De plus, la liste des mots présents est choisie selon le modèle de langage et non en fonction du vocabulaire du corpus d'apprentissage.

Afin de passer d'un mot voyellé à sa phonétisation, nous expérimentons deux jeux de règles, notés A et B, proposés respectivement dans (ALI et al., 2009) et (BIADSY, HABASH et HIRSCHBERG, 2009). Les principales différences résident dans le jeu de phonèmes et la gestion des prononciations dialectales. En particulier, le jeu de règles A génère davantage de phonétisations par mot dans la mesure où les voyelles courtes sont associées à des phonèmes plus diverses (tableau 5.2). De plus, des phonèmes spécifiques à certains dialectes sont pris en compte.

Notation du jeu A	**Description**	**Équivalence dans le jeu B**
/AE/	voyelle courte *Fatha*	
/AA/	version pharyngeale de /AE/	/a/
/AH/	version emphatique de /AE/	
/AE :/	version longue de /AE/	
/AA :/	version longue de /AA/	/A/
/AH :/	version longue de /AH/	

TABLE 5.2 – Comparaison des jeux de phonèmes A et B pour la voyelle *Fatha*

Afin de limiter les variantes de prononciation par mot, dues à la fois aux hypothèses de voyellation et aux variantes phonétiques pour un même mot voyellé, nous avons choisi de conserver uniquement la ou les deux premières hypothèses de voyellation de MADAMIRA. Dans le premier cas, appelée statégie *1-best*, nous conservons pour chaque mot en contexte la première hypothèse de voyellation de MADAMIRA selon sa mesure de confiance. Cela génère en réalité plusieurs voyellations différentes pour une même écriture graphémique en raison des différents contextes observés dans les données textuelles. Dans le deuxième cas, où stratégie *2-best*, ce sont les

deux premières hypothèses de voyellation de chaque mot en contexte qui seront phonétisées. Nous obtenons donc quatre lexiques d'apprentissage et quatre lexiques de décodage, avec respectivement deux lexiques construits selon les règles A et deux selon les règles B. La figure 5.3 donne pour chacun des lexiques générés le nombre de prononciations par mot en fonction du nombre de mots voyellés.

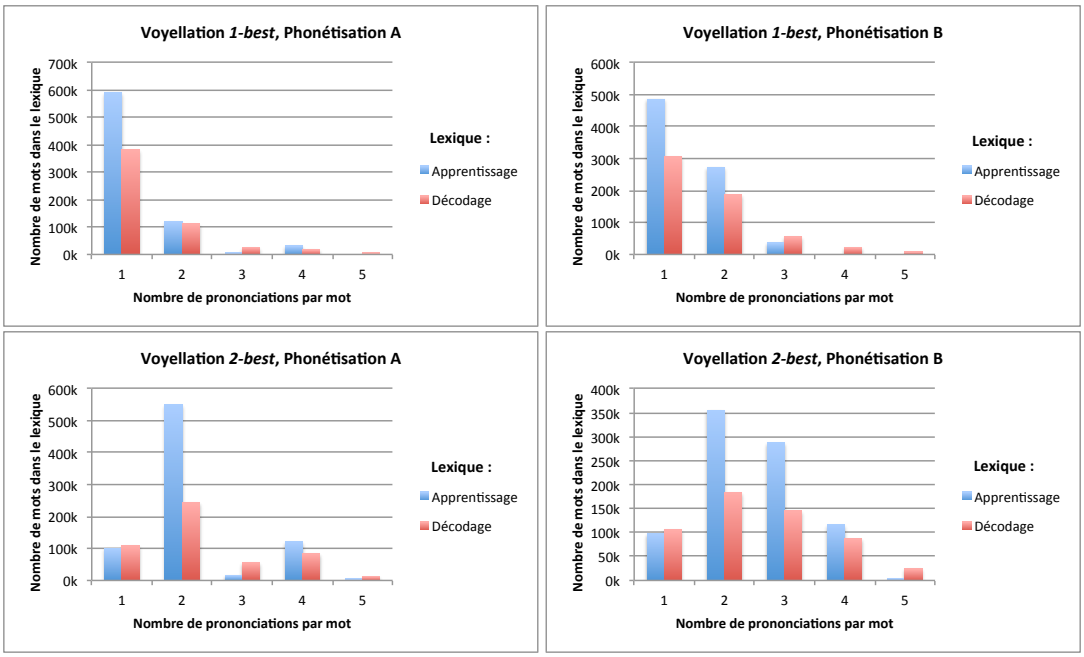

FIGURE 5.3 – Nombre de prononciations par mot en fonction du nombre de mots pour les lexiques générés à partir de stratégies de voyellation et de règles de phonétisation différentes

Nous observons que quelle que soit la stratégie utilisée, le nombre de phonétisations par mot est moins important qu'avec le lexique phonétique d'origine. De plus, le corpus d'apprentissage est mieux couvert car l'ensemble des mots présents est ici phonétisé. Au niveau des lexiques de décodage, ils comportent un nombre de mots similaire au lexique fourni par les organisateur du MGB Challenge. Il est maintenant nécessaire de sélectionner le lexique donnant les meilleures performances lors de son utilisation en reconnaissance de la parole.

5.2.4 Evaluation des lexiques dans le cadre de la reconnaissance de la parole

Pour comparer les performances des différents lexiques, nous construisons des SRAP fondés sur chacun d'eux. Les modèles acoustiques, de type HMM-GMM, sont entraînés sur l'ensemble du corpus TRAIN à l'aide de paramètres acoustiques PLP et d'une adaptation aux locuteurs fMLLR. Le modèle de langage utilisé est un modèle bigramme appris sur l'ensemble des données textuelles fournies, excepté les transcriptions du corpus DEV qui sert à l'évaluation. Les systèmes sont évalués en terme de WER et celui qui donnera les meilleurs résultats servira de base à l'apprentissage d'un modèle acoustique HMM-DNN. Le tableau 5.3 donne le taux d'erreur mot pour chaque SRAP construit, dont l'un sert de référence en étant construit avec le lexique phonétique fourni.

#	Lexique	WER
(L0)	Lexique fourni par les organisateurs du MGB Challenge (baseline)	40,8
(L1)	Lexique construit selon les règles A avec la stratégie 1-best	47,5
(L2)	Lexique construit selon les règles A avec la stratégie 2-best	44,8
(L3)	Lexique construit selon les règles B avec la stratégie 1-best	39,7
(L4)	Lexique construit selon les règles B avec la stratégie 2-best	**39,2**

TABLE 5.3 – Taux d'erreur de mot (WER), en %, des systèmes de reconnaissance de la parole selon le lexique utilisé

Nous observons que les lexiques construits selon les règles de phonétisation B permettent d'obtenir des meilleures résultats qu'avec le lexique de référence. Ce n'est pas le cas des lexiques construits selon les règles de phonétisation A. Cela confirme le choix des auteurs de (ALI et al., 2014) quant aux règles de phonétisation les plus appropriées. Un jeu de phonèmes plus simple permet en effet une meilleure capacité de généralisation des modèles acoustiques. Par ailleurs, la stratégie de voyellation *2-best* donne des meilleures résultats que la stratégie *1-best*, quelles que soient les règles de phonétisation utilisées. La combinaison d'un lexique construit selon les règles B avec la stratégie de voyellation *2-best* sera donc retenue pour le système final.

5.3 Sélection des données et alignement

Pour constituer un corpus d'apprentissage de qualité suffisante pour la modélisation acoustique, nous avons procédé à une sélection des données acoustiques et à un réalignement des transcriptions textuelles. Notre méthode s'appuie sur une approche itérative qui a notamment été utilisée pour développer le corpus en langue anglaise TEDLIUM (ROUSSEAU, DELÉGLISE et ESTÈVE, 2012). Le principe est de construire successivement des SRAP pour décoder le corpus d'apprentissage et sélectionner les segments de la référence en accord avec le résultat du décodage. La figure 5.4 montre une vue d'ensemble des étapes successives du processus de sélection des données. Le tableau 5.4 donne le taux d'erreur mot obtenu sur le corpus DEV pour chaque SRAP construit en fonction des données d'apprentissage utilisées. Pour chaque SRAP, nous utilisons le lexique graphémique afin d'obtenir un système complémentaire à celui fondé sur le lexique phonétique.

La première étape a été de construire un premier SRAP à partir de la totalité du corpus TRAIN. Un modèle acoustique HMM-GMM est entraîné à l'aide de paramètres acoustiques PLP (Perceptual Linear Prediction) et d'une adaptation aux locuteurs fMLLR (feature space maximum likelihood linear regression). Les alignements phonémiques obtenus par ce modèle permettent ensuite d'effectuer l'apprentissage d'un modèle HMM-DNN. La typologie du modèle est la suivante : une entrée de taille 40×11 où les vecteurs PLP sont pris avec leurs 10 vecteurs voisins, six couches cachées de taille 2048, et une couche de sortie de taille 8827. Un modèle de langage bigramme, appris sur l'ensemble des données textuelles fournies, est utilisé pour le décodage. Finalement, les hypothèses de reconnaissance sont réévaluées par un second modèle de langage 4-grammes estimé sur les mêmes données.

Le SRAP obtenu a permis de faire un premier décodage du corpus TRAIN. En comparant les hypothèses de reconnaissance et les transcriptions de référence, nous avons sélectionné les segments qui partageaient à la fois la même séquence de mots et le même alignement. Plus précisément, nous avons retenu les segments dont les

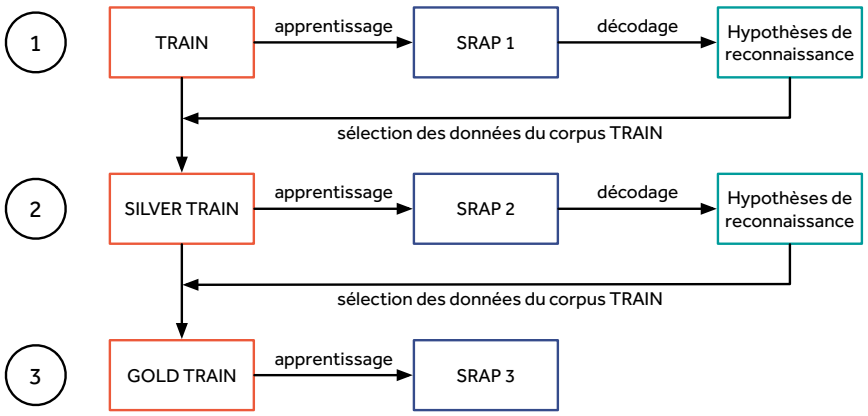

FIGURE 5.4 – Méthode de sélection des données acoustiques pour constituer un corpus d'apprentissage des modèles acoustiques à partir de données imparfaites

bornes temporelles déterminées par le SRAP étaient incluent dans les bornes temporelles de la référence. Cette première sélection nous a permis de constituer un corpus SILVER TRAIN de 326 heures de parole.

A partir du corpus SILVER TRAIN, nous construisons un deuxième SRAP avec la même topologie que le premier. Nous décodons alors une nouvelle fois les données acoustiques du corpus TRAIN qui n'ont pas été sélectionnées à la première étape. En plus des segments qui correspondent parfaitement à la transcription de référence, soit environ 16h de parole, nous ajoutons des sous-parties de certains segments qui sont partiellement alignés correctement. Pour obtenir un corpus de taille suffisante pour estimer des modèles acoustiques robustes à la tâche de reconnaissance de la campagne d'évaluation, nous ajoutons également les segments comptabilisant jusqu'à deux mots de différence. Nous obtenons finalement un corpus GOLDEN TRAIN de 648 heures de parole, à partir duquel nous construisons un troisième SRAP afin d'observer le gain apporté par la sélection de données.

Corpus	Durée (h)	Segments	WER (%)
TRAIN	1128	376 011	27,9
SILVER TRAIN	326	150 201	27,8
GOLD TRAIN	648	398 438	**26,2**

TABLE 5.4 – Taux d'erreur mot (WER) obtenu sur le corpus DEV pour chaque SRAP selon les données d'apprentissage utilisées

D'une part, nous observons que le SRAP appris sur le corpus SILVER TRAIN, bien que n'exploitant qu'à peine 30% de la totalité des données acoustiques, obtient un taux d'erreur identique au SRAP construit à partir du corpus TRAIN. Cela montre que la première étape de sélection permet bien de déterminer les segments de bonne qualité au sein de l'ensemble du corpus. D'autre part, nous observons un gain lorsqu'un SRAP de même topologie est appris sur le corpus GOLD TRAIN. C'est pourquoi nous utiliserons uniquement ce corpus pour entraîner les modèles acoustiques du système final.

5.4 Description du système proposé

5.4.1 Modèles acoustiques

La modélisation acoustique a tout d'abord reposé sur celle décrite à la section 5.3, à savoir un modèle HMM-DNN appris sur le corpus GOLD TRAIN et utilisant un lexique graphémique. Nous avons ensuite expérimenté un nouveau type de modèle acoustique, appelé *Chain Model*, fondé sur une architecture neuronale de type TDNN (PEDDINTI, POVEY et KHUDANPUR, 2015 ; POVEY et al., 2016). Ce modèle, noté $TDNN_1$, est entraîné avec des vecteurs PLP de taille 40 concaténés avec des i-vecteurs de taille 100 pour l'adaptation aux locuteurs. Un modèle à la topologie identique, noté $TDNN_2$, a également été estimé en utilisant le lexique phonétique décrit à la section 5.2 plutôt qu'un lexique graphémique.

Par ailleurs, nous avons expérimenté l'adaptation aux locuteurs proposée dans (TOMASHENKO et KHOKHLOV, 2014 ; TOMASHENKO, KHOKHLOV et ESTÈVE, 2016). Elle repose sur l'utilisation de vecteurs de paramètres dérivées des modèles GMM à la place des i-vecteurs. L'idée est de bénéficier des techniques d'adaptation aux locuteurs développées pour les GMMs, en particulier de l'adaptation *maximum a posterior (MAP)*, tout en exploitant les capacités de généralisation des modèles neuronaux. Nous avons ainsi développé deux modèles acoustiques, notés respectivement DNN_{GMMD} et $TDNN_{GMMD}$, qui reprennent les deux topologies précédemment décrites.

5.4.2 Modèles de langage

La modélisation du langage repose sur des modèles n-grammes. En particulier, nous utilisons un modèle bigramme pour le décodage et un modèle 4-grammes pour réestimer les poids associés aux hypothèses de reconnaissance. La modélisation comporte trois étapes : la préparation des données d'apprentissage, le choix du vocabulaire, et l'estimation des modèles à proprement parler.

Les données d'apprentissage sont constituées d'une part d'un corpus générique de textes issus du site *Aljazeera.net*, d'autre part des transcriptions du corpus GOLDEN TRAIN. Nous préférons en effet ne pas utiliser les transcriptions du reste du corpus TRAIN, qui peuvent éventuellement s'éloigner de réelles transcriptions de parole. De plus, nous filtrons tous les mots composés de caractères n'appartenant pas à l'alphabet de Buckwalter. Les nombres écrits en chiffres auront également été convertis en lettres.

Pour déterminer le vocabulaire adapté, nous estimons deux modèles unigrammes, chacun sur un corpus textuel. Les deux modèles sont ensuite interpolés linéairement pour obtenir un unique modèle unigramme. Les poids d'interpolation sont calculés de telle sorte que la perplexité du modèle obtenu soit minimisée sur les transcriptions du corpus DEV. Finalement, le vocabulaire que nous retenons est l'ensemble des 300 000 mots les plus fréquents d'après ce modèle de langage.

Enfin, nous estimons deux modèles bigrammes et 4-grammes sur chaque corpus textuel en limitant l'estimation au vocabulaire fixé. Les modèles sont interpolés deux à deux afin d'obtenir un modèle bigramme et un modèle 4-grammes. Cette interpolation permet de bénéficier de modèles suffisamment robustes, car estimés sur de

grandes quantités de données, tout en étant adaptés au contexte d'utilisation. Le tableau 5.5 donne la perplexité sur le corpus DEV des différents modèles de langages estimés selon le corpus d'apprentissage.

Ordre du modèle	Corpus d'apprentissage	Perplexité
2	Aljazeera.net	2377
2	GOLD TRAIN	3055
2	Interpolation linéaire avec $\lambda(Aljazeera.net) = 0,52$	**1005**
4	Aljazeera.net	2002
4	GOLD TRAIN	2754
4	Interpolation linéaire avec $\lambda(Aljazeera.net) = 0,53$	**832**

TABLE 5.5 – Perplexité des modèles de langages sur le corpus DEV en fonction des données d'apprentissage utilisées

Les résultats montrent que l'interpolation linéaire de deux modèles de langages permet d'obtenir une perplexité moindre qu'un modèle généraliste ou un modèle spécifique estimé sur peu de données. Le décodage n'est pas effectué directement avec le modèle de langage 4-grammes dans la mesure où cela prendrait trop d'espace mémoire vis-à-vis de la taille du vocabulaire.

5.4.3 Évaluation

Nous procédons à l'évaluation des différents SRAP construits. Ils se distinguent par leurs modèles acoustiques ou le lexique utilisé, mais partagent tous les mêmes modèles de langage. Le tableau 5.6 donne le WER pour chaque SRAP sur un décodage du corpus DEV.

Système	WER (%)
DNN_1	26,2
$TDNN_1$	21,4
$TDNN_2$	23,3
DNN_{GMMD}	21,9
$TDNN_{GMMD}$	22,1

TABLE 5.6 – Taux d'erreur mot (WER) obtenu sur le corpus DEV pour chaque SRAP selon le modèle acoustique utilisé

Lorsque l'adaptation aux locuteurs est issue des i-vecteurs, c'est le modèle chain TDNN qui obtient un meilleur résultat que le modèle DNN équivalent. En revanche, c'est l'inverse lorsque ce sont des vecteurs GMMD qui sont considérés. Le modèle $TDNN_2$, le seul utilisant un lexique phonétique, se place quant à lui entre les modèles DNN_1 et $TDNN_1$. Pour étudier la complémentarité des modèles acoustiques, nous évaluons les différentes combinaisons possibles (tableau 5.7). La combinaison se fait au niveau de réseaux de confusion, où chaque liaison est pondérée selon le résultat d'une interpolation linéaire sur le corpus DEV. Cela permet de trouver la combinaison optimale de modèles acoustiques.

Nous observons que bien que le modèle DNN_{GMMD} permettait d'obtenir un WER inférieur à celui obtenu avec le modèle $TDNN_{GMMD}$, ce sont bien les combinaisons de modèles TDNN qui donnent les meilleurs résultats. Il y a donc bien une complémentarité entre les lexiques graphémiques et phonétiques, ainsi qu'entre les vecteurs

Combinaison de modèles acoustiques				WER (%)
$TDNN_1$	$TDNN_2$	DNN_{GMMD}	$TDNN_{GMMD}$	
•		•		20,5
	•	•		20,5
		•	•	20,5
	•		•	20,4
•	•			20,4
•			•	20,0
•	•	•		19,9
	•	•	•	19,8
•		•	•	19,8
•	•		•	19,7
•	•	•	•	19,5

TABLE 5.7 – Taux d'erreur mot (WER) obtenu sur le corpus DEV pour chaque SRAP selon les modèles acoustiques combinés

d'adaptation aux locuteurs. Nous obtenons finalement le meilleur résultat en combinant l'ensemble des modèles acoustiques, ce qui montre que les différents types de modèles acoustiques contribuent de manière positive.

5.5 Conclusion

Dans le cadre de la campagne d'évaluation MGB Challenge, nous avons développé des systèmes de reconnaissance de la parole pour la langue arabe. La particularité de cette campagne est de proposer des données d'apprentissage imparfaites pour une langue relativement peu étudiée par les participants. Nous avons expérimenté différents types de modèles acoustiques, différentes représentations de l'identité des locuteurs, et nous avons comparé un lexique graphémique et phonétique. Finalement, nous avons montré que les différents systèmes développés étaient complémentaires et que leur combinaison contribuait à améliorer les performances de reconnaissance de la parole. De plus, nous avons montré qu'en suivant une méthode de sélection et d'alignement automatique des données, il était possible d'améliorer significativement la qualité du corpus d'apprentissage. Cette approche peut être transposée facilement à d'autres langues. Ce travail donne ainsi un cadre méthodologique pour le développement rapide de systèmes de reconnaissance de parole pour de nouvelles langues, lorsque beaucoup de données sont disponibles mais de qualité inégale. Enfin, les résultats obtenus au regard du classement de la campagne (tableau 5.8) montrent qu'il est possible de générer des systèmes performants avec l'approche proposée (ALI et al., 2016).

Participants	WER (%)
QCRI	14,7
LIUM	**16,7**
MIT	17,3
NDSC	18,2
NHK	29,5
Cairo University	38,8
Seville University	51,1
Eqra	54,7

TABLE 5.8 – Résultats officiels du MGB-2 Challenge (ALI et al., 2016)

En examinant les caractéristiques des SRAP des autres participants, nous observons des stratégies assez différentes pour obtenir de bons résultats. Le système du QCRI (KHURANA et ALI, 2016) utilise l'ensemble des 1200 heures de parole fournies pour l'apprentissage des modèles acoustiques, avec une technique d'augmentation de données consistant à modifier la vitesse et le volume du signal. Ainsi, ils obtiennent trois fois plus de données acoustiques. La modélisation acoustique résulte de la combinaison de trois modèles TDNN, LSTM et BLSTM. La modélisation du langage est réalisée par trois modèles : un modèle de langage 3-grammes pour le décodage et une réévaluation effectuée successivement par un modèle de langage 4-grammes et un RNN. Le système du MIT (ALHANAI, HSU et GLASS, 2016) utilise l'ensemble des 1200 heures de parole fournies pour l'apprentissage des modèles acoustiques. Après avoir comparé des modèles CNN, TDNN, LSTM, H-LSTM et G-LSTM, les auteurs concluent que ce sont les modèles récurrents (LSTMs) qui permettent d'obtenir les meilleurs résultats. Le système proposé combine les deux meilleures hypothèses du modèle acoustique G-LSTM, et la modélisation du langage est similaire à celle du système du QCRI : un modèle de langage 3-grammes pour le décodage et une réévaluation effectuée successivement par un modèle de langage 4-grammes et un RNN. Le NDSC (YANG et al., 2016) a effectué une sélection des données acoustiques selon le score MER fourni, aboutissant à un corpus de 680 heures de parole pour l'apprentissage des modèles acoustiques. Leur système combine des modèles acoustiques DNN, LSTM et TDNN, et la modélisation du langage est réalisée par un modèle 3-grammes lors du décodage, suivie d'une réévaluation des hypothèses à l'aide d'un modèle RNN. L'ensemble de ces participants a utilisé le lexique graphémique fourni plutôt que de travailler sur un lexique phonétique. En somme, nous remarquons que les meilleurs résultats sont obtenus avec des modèles acoustiques neuronaux récurrents et une modélisation du langage avec un réseau de neurones récurrent également. Notre participation est la seule a faire l'usage d'un lexique phonétique, qui pourtant apporte une amélioration des performances du système après combinaison. Quant aux données utilisées pour l'apprentissage des modèles acoustiques, deux stratégies sont concurrentes : améliorer le corpus d'apprentissage par des techniques d'augmentation de données, ou au contraire réaliser une sélection des données pour améliorer la précision des transcriptions.

Chapitre 6

Transcription phonétique multimodale et détection des erreurs de phonétisation

Sommaire

La synthèse de la parole s'appuie sur des transcriptions phonétiques d'enregistrements de parole. Le signal de parole est segmenté en unités acoustiques, elles-mêmes annotées par des phonèmes. En particulier, la synthèse par concaténation utilise cette annotation pour sélectionner les unités recherchées, tandis que la synthèse paramétrique repose sur ces données lors de l'apprentissage des modèles acoustiques. Dans les deux cas, la qualité d'un système de synthèse de la parole dépend de l'adéquation entre ces transcriptions et ce qui a été prononcé. Dans (Brognaux, Picart et Drugman, 2014 ; Dall et al., 2016), les auteurs montrent que la qualité de la synthèse par concaténation pour le français est améliorée lorsque les transcriptions phonétiques sont plus précises et corrigées manuellement. De plus, les auteurs de (Arik et al., 2017 ; Gibiansky et al., 2017) montrent qu'un système de synthèse de la parole neuronal converge plus rapidemment et fait moins d'erreurs de prononciation lorsqu'il est appris sur des transcriptions phonétiques plutôt que sur du texte brut. Or, les transcriptions phonétiques sont produites par une phonétisation automatique du texte. L'hypothèse de phonétisation dépend donc d'une analyse d'une séquence de caractères dans leur contexte lexical plutôt que du signal de parole. Comme il n'existe pas une manière unique de prononcer la même séquence de texte, et que la phonétisation automatique est source d'erreurs de prédiction, la phonétisation obtenue peut être erronnée. C'est pourquoi la transcription phonétique automatique est suivie d'une validation manuelle de la transcription. Cette étape de correction est potentiellement longue alors que peu d'erreurs de phonétisation sont présentes. Dans ce chapitre, nous commençons par mettre en évidence les différences entre une phonétisation automatique et le contenu phonétique prononcé. Ensuite, nous proposons une méthode pour améliorer la phonétisation automatique en utilisant des informations issues du texte et du signal de parole. Enfin, nous décrivons une méthode de détection des erreurs de phonétisation, fondée sur l'utilisation du signal de parole, qui permet de faciliter la tâche de validation manuelle des transcriptions. Les expérimentations sont menées sur une base de données en français, avant d'être étendues pour d'autres langues.

6.1 Phonétisation automatique et contenu phonétique d'un énoncé

6.1.1 Le système de phonétisation automatique de Voxygen

Le système de phonétisation automatique de Voxygen est composé d'un ensemble de modules permettant de transformer progressivement une séquence de mots en séquence de phonèmes. Quatre modules principaux peuvent être distingués : un lexique, un module de translittération, un analyseur morpho-syntaxique et un module de réécriture (figure 6.1).

Le lexique est une liste de mots où chacun d'eux est associé à une prononciation et une classe grammaticale. Ainsi, chaque mot peut avoir plusieurs prononciations différentes, décrites par une séquence de phonèmes, mais chaque prononciation est associée à une unique étiquette grammaticale. Cette dernière permet donc de distinguer des mots possédant la même orthographe, ayant des prononciations différentes. Le lexique regroupe à la fois les mots les plus courants et les mots qui ne suivent pas les règles habituelles de prononciation de la langue : on parle alors d'exceptions. Les autres mots sont couverts par un ensemble de règles de translitération.

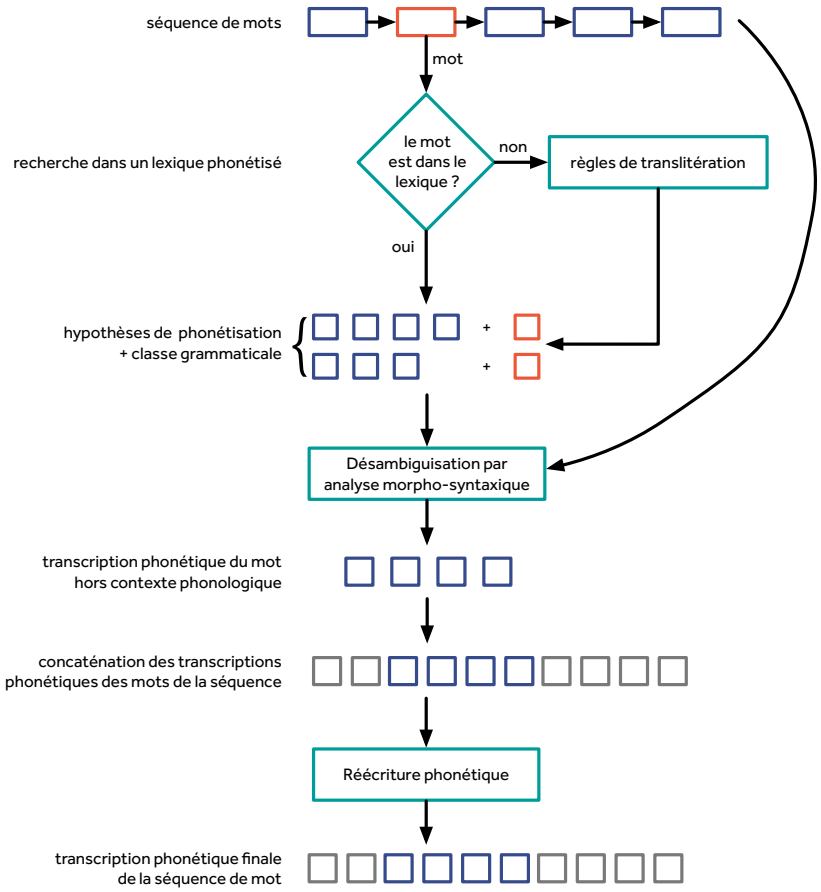

FIGURE 6.1 – Système de phonétisation automatique à base de règles

Le module de translittération se fonde sur un ensemble règles pour associer à chaque mot un ou plusieurs couples *(phonèmes, étiquette grammaticale)*. Ces règles s'appuient sur la séquence de caractères composant le mot, mais aussi sa morphologie. Par exemple, une règle peut indiquer que la séquence de caractères `[c h]` peut se prononcer `/ʃ/` comme dans *chat*, ou `/k/` comme dans *écho*. Une autre règle permettra de distinguer les prononciations de *couvent* en tant que nom commun ou en tant que conjugaison du verbe *couver*. En somme, les mêmes sorties sont attendues pour le module de translittération que pour le module lexique.

Pour chaque mot d'une séquence donnée, les hypothèses de transcription phonétique, ainsi que les classes grammaticales correspondantes, sont recherchées dans le lexique ou déterminées par des règles de translitération. Cela donne pour la séquence de mots un ensemble de séquences de phonèmes possibles et les étiquettes grammaticales correspondantes. Ensuite, une analyse morpho-syntaxique de la séquence de mots est effectuée afin de faire une hypothèse sur la classe grammaticale de chaque mot selon son contexte. Finalement, une séquence de phonèmes correspondant à la séquence de mots est déterminée d'après le résultat de cette analyse.

La séquence de phonèmes obtenue après l'analyse morpho-syntaxique ne prend pas en compte les contraintes phonologiques de la langue. C'est pourquoi un module de réécriture est appliqué sur la séquence de phonèmes dans sa totalité pour tenir compte des dépendances intra- et inter-mots. Ce module est fondé sur un ensemble de règles propres à chaque langue.

6.1.2 Erreurs de phonétisation

La transcription phonétique cherche à décrire les unités utilisées par un locuteur dans un enregistrement sonore. Le contenu phonétique de référence est ainsi la séquence de phonèmes associée à un signal de parole. Celle-ci peut être déterminée manuellement par un annotateur en écoutant le signal et en le transcrivant en phonèmes. En pratique, il part d'une phonétisation automatique et y effectue des modifications. La phonétisation automatique étant issue du texte, elle peut être différente de la séquence de phonèmes recherchée. En effet, il y a souvent plusieurs choix phonétiques possibles pour un même texte, qu'un phonétiseur automatique ne peut pas différencier. Nous qualifions ainsi d'*erreur de phonétisation* les différences entre le contenu phonétique de référence et la phonétisation automatique.

Pour mesurer la différence entre deux séquences de phonèmes, en particulier entre le contenu phonétique de référence et la phonétisation automatique, nous calculons le taux d'erreur de phonétisation, où Phone Error Rate (PER). Cette métrique correspond au taux de divergence en moyenne entre les séquences de phonèmes référence et hypothèse au sens de la distance de Levenshtein. En pratique, il s'agit de compter le nombre de substitutions, d'omissions et d'insertions de phonèmes dans la séquence hypothèse, et de diviser par le nombre de phonèmes de la séquence de référence :

$$PER = \frac{\#\text{substitutions} + \#\text{omissions} + \#\text{insertions}}{\#\text{phonèmes de la référence}} \tag{6.1}$$

Une information complémentaire pour analyser les erreurs de phonétisation provient du taux de substitutions, d'omissions et d'insertions parmi les erreurs identifiées. Ces métriques sont particulièrement intéressantes dans le cadre de la phonétisation automatique car elles permettent de donner une idée du type d'erreur commis par les systèmes de transcription phonétique. Il est important de noter que ce que nous qualifions comme *erreur de phonétisation* peut concerner des hypothèses de phonétisation appropriées pour le texte considéré. En effet, il s'agit souvent d'un choix du locuteur qui diverge de la cible phonétique, ce choix ne pouvant pas être déduit à partir du texte seul.

6.2 Transcription phonétique multimodale

6.2.1 Données expérimentales

Nous cherchons dans cette partie à obtenir des hypothèses de transcription phonétique issues de sources différentes, c'est-à-dire acoustiques et textuelles. L'objectif est d'obtenir des séquences phonétiques plus proches de la réalité acoustique que ce que permet une phonétisation automatique à partir du texte seul. Pour cela, nous considérons une base de données de synthèse de la parole de plusieurs voix en français, consituée d'enregistrements de parole et de leurs transcriptions en mots et en phonèmes. Le texte correspondant aux séquences de mots du corpus a été transcrit en phonèmes à l'aide du système de phonétisation automatique de Voxygen, puis la transcription phonétique a été vérifiée et corrigée manuellement. Le corpus a été construit pour la synthèse par concaténation, c'est-à-dire que pour chaque voix, le

contenu phonétique possède une large couverture des séquences de phonèmes possibles dans la langue. Les séquences de mots sont représentées sous la forme de séquences de caractères, où un symbole permet d'identifier les frontières de mots. Les séquences de phonèmes associées sont également délimitées entre les mots (tableau 6.1).

Graphemes	l e s | é c r a n s | s o n t | a l l u m é s
Phonemes	L EI | Z EI K R AN | S ON | T A L U M EI

TABLE 6.1 – Représentation d'un exemple du corpus phonétisé.

Nous divisons la base de données de synthèse en deux parties pour distinguer les données utiles pour l'apprentissage des modèles et les données nécessaires à leur évaluation. La première constitue le corpus d'apprentissage et a une durée de 50 heures de parole, comportant 9 locuteurs pour 90 135 séquences de mots. La deuxième, le corpus de test, comporte 1 heure de parole, pour 3 locuteurs et 950 séquences de mots. L'ensemble des séquences phonétiques associées aux enregistrements ont été vérifiées manuellement après l'écoute de ces derniers, conformément au protocole de création de voix de synthèse de Voxygen. Celles-ci constituent la référence pour l'apprentissage et l'évaluation des systèmes de phonétisation automatique. Les locuteurs et les séquences de mots du corpus de test sont distincts de ceux du corpus d'apprentissage. Chaque séquence de mots correspond à une phrase ou un groupe de souffle.

6.2.2 Phonétisation automatique à partir du texte

Nous avons présenté précédemment le système de phonétisation automatique de Voxygen (S0). Nous cherchons maintenant à le comparer à d'autres méthodes de phonétisation automatique à l'état de l'art. L'objectif est d'obtenir des hypothèses de phonétisation plus précises ou complémentaires. Pour cela, nous construisons deux systèmes de phonétisation automatique : l'un est fondé sur un modèle de n-grammes joints (S1), l'autre est construit autour d'une architecture neuronale encodeur-décodeur avec un mécanisme d'attention (S2).

Pour le modèle n-grammes joints, nous utilisons le toolkit *Phonetisaurus* (NOVAK et al., 2012 ; NOVAK, MINEMATSU et HIROSE, 2013), dont l'apprentissage s'effectue en deux étapes. Premièrement, un alignement est réalisé pour chaque mot entre les séquences de caractères et les séquences de phonèmes correspondantes. Cet alignement repose sur le principe de l'algorithme *Expectation-Maximization*, qui s'effectue itérativement, et permet d'aboutir à un corpus de séquences de couples *(graphèmes, phonèmes)*. La deuxième étape consiste à entraîner un modèle de langage sur le corpus obtenu afin de générer les hypothèses de phonétisation. Nous avons alors construit un modèle de langage intra-mots 6-grammes reposant sur *SRILM* (STOLCKE, 2002) Le système de phonétisation obtenu permet d'obtenir une ou plusieurs hypothèses de phonétisation pour des mots isolés. Cependant, il n'est pas capable de prendre en compte le contexte lexical dans sa décision.

Le modèle de phonétisation neuronal repose sur une architecture encodeur-décodeur avec un mécanisme d'attention (figure 6.2). Ce dernier permet de pondérer chaque lettre d'une séquence de mots en entrée dans le choix des phonèmes en sortie, en tenant compte de dépendances intra- et inter-mots. Ainsi, les hypothèses formulées

sont dépendantes du contexte lexical et peuvent donc prendre en compte les liaisons ou la position des mots dans la séquence par exemple.

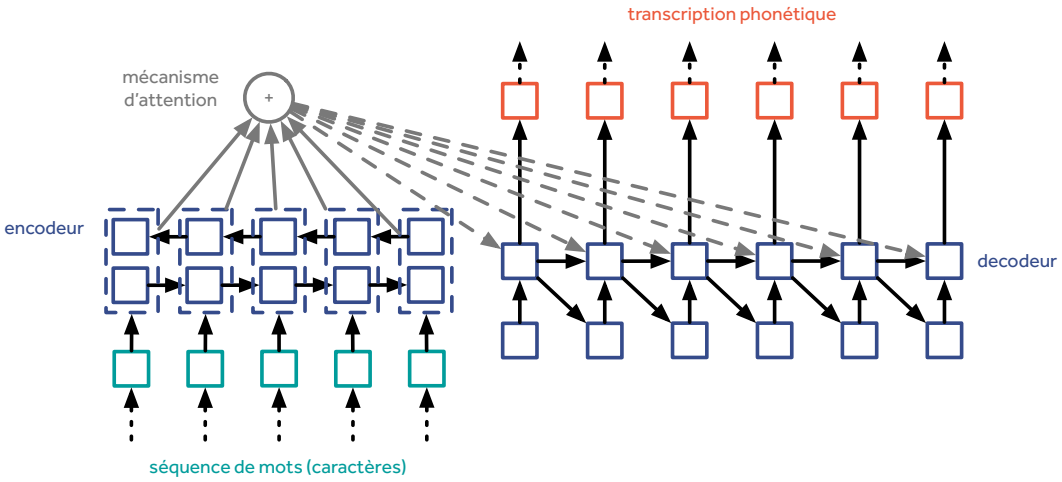

FIGURE 6.2 – Système de phonétisation automatique neuronal de bout en bout.

Nous avons choisi de suivre l'architecture proposée par (BAHDANAU, CHO et BENGIO, 2014) en nous appuyant sur le toolkit *NMTPy* (CAGLAYAN et al., 2017) pour implémenter notre modèle. L'encodeur est composé d'une couche récurrente bidirectionnelle tandis que le décodeur consiste en deux couches de GRU additionnées d'un mécanisme d'attention. Les caractères sont représentées par des *embeddings* de 64 dimensions, puis ces vecteurs sont projetés sur des couches de 128 dimensions. L'apprentissage du modèle utilise l'algorithme d'optimisation *Adam* avec un *learning rate* égal à 10^{-4} pour des *batch* de taille 32. Finalement nous appliquons la technique du *dropout* pour chaque couche récurrente avec une probabilité de 0, 4.

La tableau 6.2 donne le PER des différents systèmes de phonétisation automatique.

#	Système	PER	S	O	I
(S0)	phonétisation automatique par règles	2,9	70,9	17,8	11,3
(S1)	phonétisation par modèle de n-grammes joints	6,4	41,4	50,3	8,3
(S2)	phonétisation par modèle neuronal	2,8	62,8	22,9	14,3

TABLE 6.2 – Taux d'erreur de phonétisation (PER) des systèmes de phonétisation automatique, dont la proportion de substitutions (S), d'omissions (O) et d'insertions (I), en %.

Pour le système à base de règles, la plupart des erreurs de phonétisation sont des substitutions (70,9%). Celles-ci résultent d'une divergence du locuteur par rapport à la cible phonétique. Le modèle de n-grammes joints est davantage pénalisé par les omissions (50,3%). Quant au modèle neuronal, il a un taux d'erreur proche de celui à base de règles.

Nous procédons ensuite à une analyse qualitative plus fine des erreurs de phonétisation. Nous remarquons que les erreurs principales concernent des confusions entre voyelles ouvertes et fermées, des confusions entre consonnes voisées et non voisées, des contraintes phonologiques qui n'ont pas été anticipées correctement (e.g. 72 : `/swasãtəduz/` vs. `/swasãnduz/`, 36 : `/trãnsis/` vs. `/trãtsis/`), ou des erreurs

au niveau des liaisons. Le diagramme 6.3 détaille le type des erreurs des différents systèmes :

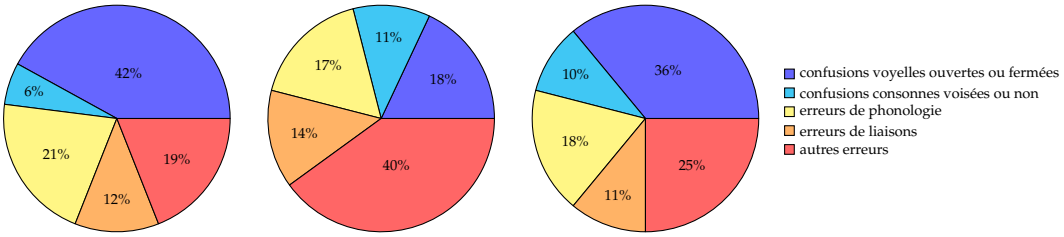

FIGURE 6.3 – Répartition des erreurs des systèmes de phonétisation automatique – respectivement (S0), (S1) et (S2) – selon leur type.

Nous observons que le phonétiseur à base de règles et le phonétiseur neuronal, en plus d'avoir un taux d'erreur proche, ont une répartition similaire des types d'erreurs. En effet, les confusions entre voyelles ouvertes et fermées sont prédominantes, ainsi que des problèmes au niveau de la gestion des contraintes phonologiques dans une moindre mesure. D'autres erreurs spécifiques sont relativement fréquentes, comme les confusions entre consonnes voisées et non-voisées et les problèmes de liaisons. Ces erreurs ne peuvent être détectées qu'à l'écoute des enregistrements et sont très pénalisantes pour la synthèse de la parole. Les deux systèmes se comportent donc globalement de la même manière, ils ne se différentient ni par le taux d'erreur, ni par leur complémentarité. Le phonétiseur fondé sur un modèle de n-grammes fait quant à lui des erreurs moins typées, et en plus grand nombre. On remarque tout de même qu'il fait relativement plus d'erreurs sur les erreurs de liaison, ce qui se comprend par le fait qu'il ne prend pas en compte le contexte lexical. Pour obtenir une phonétisation plus juste vis-à-vis du signal de parole, un phonétiseur capable de prendre une décision en contexte et dépendante des enregistrements acoustiques semble plus appropriée.

6.2.3 Alignement forcé

Pour obtenir une phonétisation du texte dépendante du signal acoustique, nous nous proposons de réaliser un alignement forcé du texte de référence. L'alignement forcé utilise un modèle acoustique et un dictionnaire de prononciations, construit à partir de la liste des mots du corpus d'évaluation, afin de faire correspondre dans le domaine temporel le texte et les enregistrements acoustiques. Ceci nous permet d'obtenir une phonétisation dépendante du signal acoustique, que nous comparerons à la phonétisation de référence.

Le modèle acoustique est construit à l'aide du toolkit de reconnaissance de la parole Kaldi (POVEY et al., 2011). Nous avons tout d'abord construit un modèle acoustique HMM/GMM traitant le signal comme des vecteurs de coefficients MFCC adaptés aux locuteurs par fMLLR. Ensuite, nous avons appris un modèle TDNN-LSTM/HMM avec des couches cachées de 300 dimensions grâce aux alignements phonémiques produits par le modèle HMM/GMM. Celui-ci prend en entrée des vecteurs de 40 coefficients MFCC haute-résolution et des i-vecteurs de 100 dimensions pour l'adaptation aux locuteurs.

Le dictionnaire de prononciations utilisé pour l'alignement forcé est celui inclus dans le phonétiseur à base de règles. Cependant, étant tout d'abord conçu pour la synthèse de la parole, certaines variantes de prononciation peuvent en être absentes. En effet, il s'agit en synthèse de la parole de choisir une variante de prononciation appropriée parmi toutes celles possibles. De plus, l'alignement forcé pourrait être limité par une dépendance trop importante à une phonétisation à partir du texte. Si la phonétisation attendue n'est pas proposée dans le dictionnaire de prononciations, il n'y a aucune possibilité pour que l'alignement forcé donne la bonne solution. Ainsi, la complémentarité recherchée entre phonétisation automatique du texte et alignement forcé ne peut pas être parfaite. Nous décidons alors d'enrichir le lexique à l'aide d'hypothèses de phonétisation automatique : nous ajoutons au lexique les 1, 2 ou 3 meilleures hypothèses de prononciation générées automatiquement pour chaque mot. Ces hypothèses sont générées à l'aide de notre modèle de n-grammes joints, particulièment adapté au traitement de mots isolés.

Le tableau suivant donne le taux d'erreur de phonétisation de l'alignement forcé en fonction de l'enrichissement du lexique de prononciation :

#	Enrichissement (en nombre de variantes par mot)	PER	S	O	I
(S3)	0	4.6	49,6	5,0	45,4
(S3+1)	1	**4.3**	49,7	5,9	44,5
(S3+2)	2	4.9	52,4	6,5	41,1
(S3+3)	3	5.3	52,5	6,7	40,8

TABLE 6.3 – Taux d'erreur de phonétisation (PER) des systèmes d'alignement forcé en fonction de l'enrichissement du lexique de prononciation, en %.

Nous observons dans le tableau 6.3 que l'ajout au lexique de la meilleure hypothèse du phonétiseur à modèles de n-grammes joints permet de réduire le taux d'erreur du système. Cependant, ajouter davantage de variantes de prononciation impacte négativement l'alignement forcé. Un compromis est donc à trouver entre la liberté laissée au modèle acoustique et la précision du lexique phonétique.

Nous remarquons également que la répartition des erreurs entre substitutions, omissions et insertions est différente de celle obtenue avec les systèmes de phonétisation en contexte à partir du texte. En effet, il y a beaucoup moins d'omissions, mais plus d'insertions. Nous supposons que l'alignement forcé réalise plus d'erreurs sur les liaisons ou les variantes phonologiques, qui sont plus difficiles à représenter dans un lexique de mots isolés. Pour s'en assurer, nous analysons qualitativement les erreurs produites par le système (S3+1), celui obtenant les meilleurs résultats.

Sur la figure 6.4, nous observons que les erreurs de phonétisation suivent une répartition similaire à celles du phonétiseur à base de règles et du phonétiseur neuronal. La prise en compte des données acoustiques avec l'alignement forcé ne suffit donc pas à obtenir des hypothèses complémentaires dans le type d'erreurs. Nous supposons que le lexique de mots isolés ne permet pas suffisamment de liberté dans le choix de la phonétisation par le modèle acoustique. Pour vérifier notre hypothèse, nous expérimentons la reconnaissance de phonèmes, qui consiste à transcrire directement le signal de parole en chaînes phonétiques.

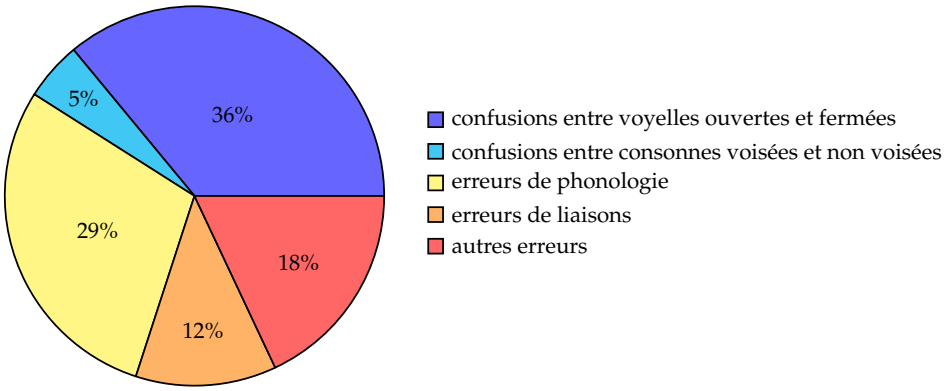

FIGURE 6.4 – Répartition des erreurs du système d'alignement forcé (S3+1) selon leur type.

6.2.4 Reconnaissance de phonèmes

La reconnaissance de phonèmes consiste à transcrire un signal de parole en une séquence de phonèmes. Autrement dit, il s'agit d'une tâche de reconnaissance automatique de la parole où le vocabulaire est l'ensemble des phonèmes possibles. Nous avons expérimenté un système de reconnaissance de la parole de bout-en-bout en nous appuyant sur l'architecture de Deepspeech 2 (AMODEI et al., 2016). Il s'agit d'un réseau de neurones utilisé comme un modèle de séquences jointes selon la terminologie utilisée dans le chapitre 2. Il est constitué de deux couches de convolution, de cinq couches récurrentes, et d'une couche de neurones complètement connectés. En entrée, le signal est donné sous la forme de spectrogramme, et les phonèmes sont générés en sortie avec une fonction *softmax*. L'alignement préalable des séquences de phonèmes avec le signal dans le corpus d'apprentissage est évité grâce à l'utilisation de la fonction CTC.

Un des avantages d'une architecture neuronale de bout-en-bout pour la reconnaissance de phonèmes est de permettre l'inclusion d'informations additionnelles externes, aussi bien à l'apprentissage qu'au décodage. Nous expérimentons ainsi l'ajout d'informations sur le texte en plus du signal en entrée. Une connaissance de la transcription en mots pourrait en effet bénéficier à la transcription en phonèmes. Dans la mesure où nous avons observé de bons résultats de conversion du texte en phonèmes à l'aide du phonétiseur neuronal décrit à la section 6.2.2, nous ajoutons à l'entrée du système la sortie de l'encodeur du phonétiseur à partir du texte. Nous comparons deux approches pour inclure cette information : concaténer à la sortie des couches de convolution le dernier état de l'encodeur de texte, ou concaténer la somme des états de l'encodeur de texte. Les deux approches ont pour but d'obtenir un unique vecteur représentatif de la séquence de mots, ceci afin d'éviter un alignement entre la séquence de lettres et la séquence de phonèmes. La figure 6.5 montre l'architecture du système de transcription phonétique multimodal obtenu.

Le tableau 6.4 donne le taux d'erreur de phonétisation pour les systèmes de reconnaissance de phonèmes appris sur les données décrites à la section 6.4.1. Le système (S4) correspond au modèle Deepspeech 2 uniquement, tandis que les systèmes (S4+*last*) et (S4+*sum*) ont une entrée multimodale, en ajoutant respectivement au spectrogramme du signal le dernier état et la somme des états de l'encodeur de texte.

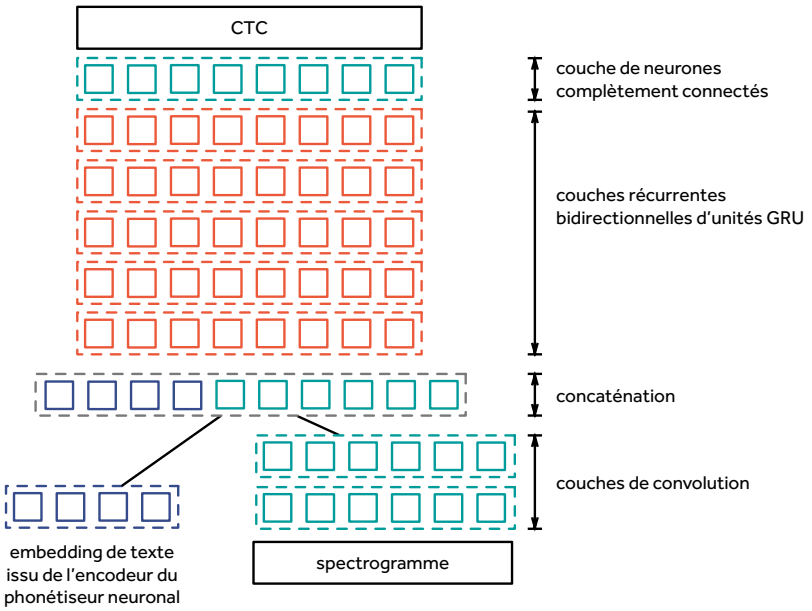

FIGURE 6.5 – Architecture du système de reconnaissance de phonèmes multimodal, prenant en entrée un vecteur encodant la séquence de mots et le spectrograme du signal

#	Système	PER	S	O	I
(S4)	reconnaissance de phonèmes de bout en bout	9,9	78,8	14,6	6,6
(S4+*last*)	(S4) + dernier état de l'encodeur de (S2)	10,3	69,7	24,5	5,8
(S4+*sum*)	(S4) + somme des états de l'encodeur de (S2)	9,6	77,2	15,9	6,9

TABLE 6.4 – Taux d'erreur de phonétisation (PER) des systèmes de reconnaissance de phonèmes de bout en bout, dont la proportion de substitutions (S), d'omissions (O) et d'insertions (I), en %.

Nous observons que le taux d'erreur de phonétisation des systèmes de reconnaissance de phonèmes est plus élevé que celui des systèmes de phonétisation à partir du texte ou fondés sur un alignement forcé. Cependant, la reconnaissance de phonèmes est directement complémentaire à la phonétisation à partir du texte, prenant ses décisions d'après une modalité différente. Au contraire de l'alignement forcé, cette approche ne dépend pas d'un vocabulaire limité ou de variantes de prononciation finies. C'est cette caractéristique que nous recherchons dans le cadre de la détection des erreurs de phonétisation. De plus, l'ajout d'informations sur le texte en le représentant par la somme des états de l'encodeur de (S2) permet d'améliorer la justesse de ses hypothèses de reconnaissance. Cependant, dans le cadre du développement rapide de systèmes, la nécessité d'estimer deux modèles est discutable au regard du gain obtenu.

La figure 6.6 montre la répartition des erreurs des systèmes de phonétisation automatique selon leur type. Nous remarquons qu'un grand nombre d'erreurs n'est pas typé, c'est-à-dire qu'ils ne correspondent pas aux erreurs principalement observées auparavant. De plus, les confusions entre consonnes voisées ou non, ainsi que les confusions entre consonnes voisées ou non, sont toujours relativement nombreuses. La reconnaissance de phonèmes est cependant moins sujette aux erreurs au niveau

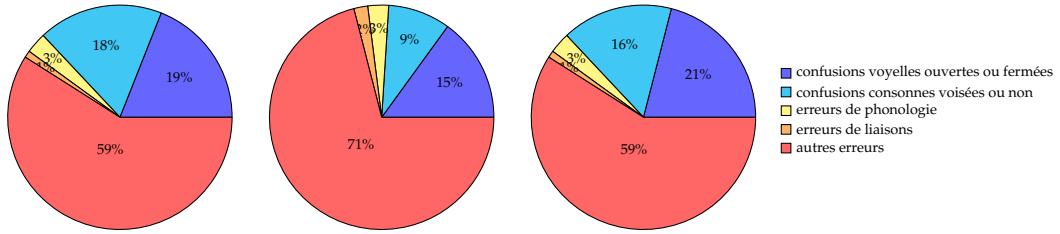

FIGURE 6.6 – Répartition des erreurs des systèmes de phonétisation automatique – respectivement (S4), (S4+*last*) et (S4+*sum*) – selon leur type.

des liaisons ou des contraintes phonologiques. Ces types d'erreurs sont donc facilement identifiables à l'aide du modèle acoustique. Les résulats obtenus avec le système (S4+*last*) sont globalement moins bons qu'avec les deux autres systèmes. Ces derniers ont quant à eux des résultats identiques.

6.3 Détection des erreurs de phonétisation

Nous avons vu que la phonétisation automatique repose sur un traitement du texte. Celle-ci est source d'erreurs dans la mesure où un locuteur peut ne pas suivre cette cible phonétique lors d'un enregistrement. Les erreurs sont de plusieurs types et peuvent affecter négativement la qualité des systèmes de synthèse de la parole. Nous nous intéressons ici à une méthode permettant de limiter les erreurs de phonétisation dans les bases de données de synthèse de parole. Le principe est de détecter des zones susceptibles d'être erronées et de demander la correction manuelle des phonèmes concernés à un opérateur humain. L'avantage de cette méthode est d'éviter la vérification manuelle de l'ensemble des bases de données concernées. Ainsi, nous cherchons à maximiser le nombre d'erreurs détectées tout en minimisant le travail de validation manuelle.

6.3.1 Tâche de détection d'erreurs

Nous avons précédemment expérimenté différents systèmes de transcription phonétique, à partir du texte, à partir du signal ou multimodal. La tâche de détection d'erreurs consiste à comparer les transcriptions automatiques entre elles afin de repérer les zones de consensus et les zones de désaccord. Lorsque deux transcriptions générées par deux systèmes différents sont identiques, nous considérons que les transcriptions sont correctes. Au contraire, lorsque les transcriptions divergent, nous supposons qu'il y a un doute sur la séquence phonétique et nous considérons erronés les phonèmes concernés. Plus précisément, étant donné que c'est le système de phonétisation automatique de Voxygen qui est employé pour transcrire les bases de données de synthèse, ce sont les erreurs des transcriptions issues de ce dernier que nous cherchons à détecter. Ainsi, nous comparons systématiquement les hypothèses de transcription phonétique avec la phonétisation à base de règles. La figure 6.7 montre une vue d'ensemble de l'architecture du système de détection des erreurs de phonétisation.

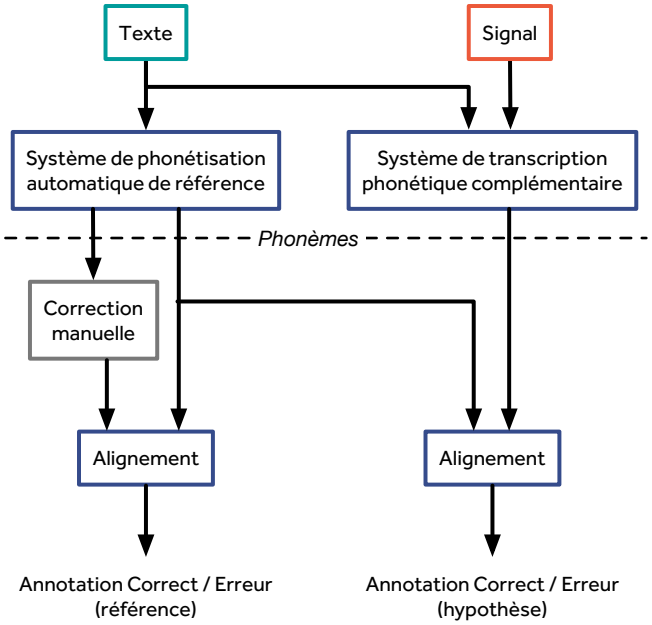

FIGURE 6.7 – Méthode de détection d'erreurs de transcription phonétique.

Nous pouvons évaluer la tâche de détection d'erreur à part entière dans la mesure où les transcriptions phonétiques correctes, obtenues après validation manuelles, sont disponibles. La phonétisation à base de règles est alignée à la phonétisation de référence, c'est-à-dire la phonétisation automatique de référence suivie d'une correction manuelle, pour identifier les zones d'erreurs. Les deux phonétisations automatiques sont également alignées, ce qui constitue les hypothèses d'erreurs. L'évaluation de la détection des erreurs de phonétisation consiste alors à comparer les hypothèses d'erreurs à celles identifiées par la référence.

Nous évaluons la détection d'erreurs avec deux métriques standard, à savoir la *Précision* et le *Rappel*. Appliquées à la détection d'erreurs, la Précision donne la proportion d'erreurs indiquées par le système qui sont réellement des erreurs, et le Rappel mesure la proportion d'erreurs retrouvées automatiquement. De plus, pour mesurer le gain de temps pour la validation manuelle, nous introduisons une nouvelle métrique, appelée Manual Checking Rate (MCR). Cette métrique consiste à donner la proportion de données phonétiques à vérifier pour couvrir l'ensemble des zones détectées comme erronées :

$$MCR = \frac{\text{nombre de phonèmes détectés comme erronés}}{\text{nombre de phonèmes de la référence}}$$

Nous évaluons la détection d'erreurs sur les mêmes données que la tâche de transcription phonétique, avec les systèmes précédemment décrits.

6.3.2 Résultats

Nous commençons par évaluer la détection d'erreurs avec les systèmes de phonétisation à partir du texte (tableau 6.5) sur le même corpus de test que pour la tâche de

transcription phonétique.

Système	Précision	Rappel	MCR
(S1)	28,0	**61,4**	6,2
(S2)	**58,7**	53,0	**2,6**

TABLE 6.5 – Évaluation de la détection d'erreurs avec les systèmes de phonétisation à partir du texte.

Nous observons qu'avec le système (S1), fondé sur un modèle n-grammes joints, le Rappel est plus important que pour le système (S2), fondé sur un modèle neuronal. C'est l'objectif recherché dans la mesure où plus le Rappel est important, plus il y a d'erreurs détectées. Cependant, la Précision est relativement faible. Autrement dit, il y a beaucoup de fausses alarmes, ce qui conduit à devoir vérifier 6,2% des données manuellement, plutôt que 2,6% avec le système (S2). En effet, le taux d'erreur de phonétisation de (S1) était plus important que celui des autres systèmes (tableau 6.2). Le Rappel observé avec le système (S2) est cependant relativement faible. Ceci peut être expliqué par les résultats de phonétisation proches entre (S0) et (S2). En effet, nous n'avions pas observé de complémentarité entre la phonétisation par règles et le système neuronal. Pour confirmer cette hypothèse, nous évaluons la détection des erreurs de phonétisation obtenue avec les systèmes d'alignement forcé (tableau 6.6).

Système	Précision	Rappel	MCR
(S3)	33,9	52,8	4,3
(S3+1)	**38,0**	65,7	**4,8**
(S3+2)	35,4	75,0	5,9
(S3+3)	33,8	**78,9**	6,5

TABLE 6.6 – Évaluation de la détection d'erreurs avec les systèmes fondés sur l'alignement forcé.

Nous observons que davantage d'erreurs sont détectées à l'aide des systèmes fondés sur l'alignement forcé, grâce à leur complémentarité avec la phonétisation à partir du texte. Plus les contraintes sur le lexique sont relâchées, c'est-à-dire plus il y a de variantes de prononciation autorisées par mot, plus le Rappel augmente. Cependant, la meilleure Précision est obtenue avec une variante additionnelle par mot, avant de diminuer à nouveau. Cela est à mettre en lien avec les résultats de phonétisation (tableau 6.3), qui présentaient la même distribution. Nous comprenons qu'un trop grand nombre de variantes dans le lexique, potentiellement fausses, diminuent les performances de l'alignement forcé. Il s'agit donc de trouver un compromis entre la quantité de données à vérifier et la quantité d'erreurs détectées. Nous évaluons ensuite la détection d'erreurs avec les systèmes de reconnaissance de phonèmes, qui dépendent uniquement du signal (tableau 6.7).

Système	Précision	Rappel	MCR
(S4)	18,7	81,2	12,2
(S4+*last*)	16,2	78,6	13,7
(S4+*sum*)	18,5	80,0	12,2

TABLE 6.7 – Évaluation de la détection d'erreurs avec les systèmes de reconnaissance de phonèmes

La détection d'erreurs avec les systèmes de reconnaissance de phonèmes permet d'obtenir le Rappel le plus important. Ainsi, plus de 80% des erreurs peuvent être

détectées. Ceci se fait au prix d'une Précision relativement faible, même si la quantité de données à vérifier reste contenue : il est possible de se limiter à 12,2% de données à vérifier manuellement. Autrement dit, cela permet de diviser le temps de vérification manuelle d'un facteur de 8. Même si nous observions un taux d'erreur de phonétisation plus faible avec le système (S4+*sum*), cette information supplémentaire ne permet d'obtenir ni un meilleur Rappel, ni une meilleure Précision. Il semble donc plus judicieux de ne considérer qu'une approche simple de la reconnaissance de phonèmes.

6.4 Extension à d'autres langues

Nous avons présenté sur des données en français une méthode de détection des erreurs de phonétisation. Nous expérimentons maintenant la même méthode sur d'autres langues, où moins de données sont disponibles pour estimer les modèles. L'objectif est de réduire le coût de validation manuelle dans un processus de création de nouvelles voix de synthèse de la parole. Deux cas se présentent : lorsque qu'une nouvelle voix est créée pour une langue qui a déjà été traitée et lorsque qu'une voix est créée dans une nouvelle langue. Dans le premier cas, nous illustrons la méthode par la création de voix en italien et en arabe, et dans le deuxième cas, nous illustrons la méthode par la création d'une voix en turc. Ainsi, un corpus de phrases est enregistré pour chaque langue et une transcription phonétique est générée automatiquement à partir du texte. Nous essayons alors de détecter automatiquement les erreurs de phonétisation, qui seront ensuite vérifiées manuellement pour obtenir une base de données de synthèse corrigée.

6.4.1 Création d'une voix pour une langue déjà traitée

Dans le cas d'une langue déjà traitée, nous disposons d'un phonétiseur de référence ainsi que des données acoustiques transcrites en mots et en phonèmes. Ces dernières correspondent aux bases de données de signal de parole utilisées pour la synthèse des voix existantes. Des modèles peuvent donc être estimés sur ces données pour la création de la nouvelle voix.

Données expérimentales

Les données expérimentales sont composées de bases de données pour la synthèse de parole en italien et en arabe. Contrairement au français, peu de données transcrites manuellement sont déjà disponibles. En effet, nous disposons des données de deux voix italiennes et une voix en arabe dont les transcriptions ont été vérifiées manuellement. De plus, nous avons les enregistrements d'une nouvelle voix italienne et d'une voix arabe, mais dont la transcription phonétique a été déterminée automatiquement à partir du texte. Le tableau 6.8 donne la répartition des données utilisées.

Les données italiennes sont composées de bases de données pour la synthèse de la parole de troix voix féminines It_1, It_2 et It_3. Les transcriptions phonétiques des voix It_1 et It_2 ont été validées manuellement, alors que les transcriptions de la voix It_3 proviennent d'une phonétisation automatique du texte. Les voix It_1 et It_2 forment un corpus $TRAIN$ pour l'apprentissage des modèles. Un corpus $TEST$ de 200 phrases

Langue	*TRAIN*		*TRAIN*$^+$		*TEST*	
	Segments	Parole	Segments	Parole	Segments	Parole
Italien	6 805	10,0 h	1 971	4,6 h	200	0,6 h
Arabe	4 765	11,3 h	2 709	7,6 h	200	0,5 h

TABLE 6.8 – Répartition des données en italien et arabe pour la transcription phonétique et la détection des erreurs de phonétisation

est extrait des données de la voix It_3 et est validé manuellement pour l'évaluation des systèmes. Le reste des données de It_3 constitue le corpus $TRAIN^+$.

En arabe, seulement une voix masculine et une voix féminine ont été enregistrées, Ar_1 et Ar_2, mais avec un plus grand volume de parole. Ar_1 a été validée manuellement, tandis que Ar_2 ne bénéficie que d'une transcription phonétique automatique. Selon le même principe que pour l'italien, Ar_1 constitue un corpus $TRAIN$, 200 phrases de Ar_2 vérifiées manuellement constituent le corpus $TEST$, et le reste de Ar_2 constitue le corpus $TRAIN^+$.

Transcription phonétique

Les enregistrements des voix It_3 et Ar_2 ont été collectés lors de la lecture de textes offrant un compromis entre la couverture des unités acoustiques des langues considérées et la quantité de parole à enregistrer. La transcription phonétique de ces enregistrements a ensuite été obtenue à l'aide d'un phonétiseur à partir du texte. Ce phonétiseur utilise des règles de translittération et un dictionnaire de prononciation, comme décrit à la section 6.1.1. Le tableau 6.9 donne le taux d'erreur de phonétisation à partir du texte pour chacune des langues.

Langue	**PER**	**S**	**O**	**I**
Italien	1,3	14,8	9,0	76,2
Arabe	3,8	20,6	4,8	74,6

TABLE 6.9 – Taux d'erreur de phonétisation (PER) sur le corpus *TEST* et proportions S/O/I du système de phonétisation à partir du texte, en %.

Nous remarquons que le taux d'erreurs de phonétisation en italien est similaire à celui observé en français. Cependant, en arabe, il est relativement plus élevé. Dans la mesure où la plupart des voyelles ne sont pas écrites en arabe et que c'est une langue plus riche morphologiquement, cela rend la tâche de phonétisation à partir du texte plus difficile que pour l'italien. Pour les deux langues, nous observons en majorité des erreurs d'insertions. Celles-ci sont principalement dues pour l'italien au phénomène de gémination, et pour l'arabe à la terminaison des mots dont la prononciation peut varier selon le contexte lexical. Notre objectif est de corriger ces erreurs de phonétisation, pour une langue comme pour l'autre, en minimisant la vérification manuelle des données.

Nos expérimentations en français ont montré qu'un maximum d'erreurs de phonétisation étaient détectées en comparant la phonétisation à partir du texte et les transcriptions d'un système de reconnaissance de parole de bout-en-bout (section 6.2.4). Le Rappel était en effet le plus important avec le système de transcription phonétique à partir du signal seul. Ainsi, nous concentrons nos efforts sur la construction de modèles neuronaux de reconnaissance de phonèmes pour l'italien et l'arabe.

Chacun des systèmes est appris sur le corpus *TRAIN* propre à la langue traitée. Le tableau 6.10 donne le taux d'erreur de phonétisation sur le corpus *TEST* pour les systèmes de reconnaissance de phonèmes.

Langue	PER	S	O	I
Italien	30,9	61,6	22,2	16,2
Arabe	65,1	75,8	15,0	9,2

TABLE 6.10 – Taux d'erreur de phonétisation (PER) sur le corpus *TEST* et proportions S/O/I des systèmes de reconnaissance de phonèmes de bout-en-bout appris sur le corpus *TRAIN*, en %.

Nous remarquons que le taux d'erreurs de phonétisation de l'arabe et largement plus important que celui de l'italien, lui-même nettement plus élevé que celui observé en français. Nous supposons que comme les modèles sont estimés sur peu de données, et notamment avec peu de locuteurs différents, ils ne sont pas robustes à un nouveau locuteur. En particulier, nous pensons que le modèle de l'arabe est moins robuste au changement de locuteur que le modèle de l'italien car appris sur une seule voix. Nous avons donc cherché à adapter les modèles au nouveau locuteur du corpus *TEST*. Pour cela, et comme le but de notre approche est de vérifier la totalité de l'annotation de $TRAIN^+$, nous divisons pour chacune des langues ce dernier corpus en deux parties, notées $TRAIN_1^+$ et $TRAIN_2^+$. En partant des modèles entraînés sur *TRAIN*, nous estimons deux nouveaux modèles pour chaque langue sur $TRAIN_1^+$ et sur $TRAIN_2^+$. Autrement dit, nous réalisons une étape de *fine tuning*. Les modèles appris sur *TRAIN* et $TRAIN_1^+$ (respectivement $TRAIN_2^+$) nous servirons à décoder $TRAIN_2^+$ (respectivement $TRAIN_1^+$). Pour évaluer les performances des modèles, nous calculons leur taux d'erreur de phonétisation sur le corpus *TEST* en moyennant les résultats des deux modèles pour chaque langue (tableau 6.11).

Langue	PER	S	O	I
Italien	10,3	46,1	33,8	20,1
Arabe	5,1	72,7	17,0	10,3

TABLE 6.11 – Taux d'erreur de phonétisation (PER) sur le corpus *TEST* et proportions S/O/I des systèmes de reconnaissance de phonèmes de bout-en-bout appris sur le corpus *TRAIN* et la moitié du corpus $TRAIN^+$, en %.

Après une spécialisation des modèles sur les nouveaux locuteurs, nous observons un taux d'erreur significativement plus faible. Le modèle de l'italien passe ainsi de $30,9\%$ à $10,3\%$ d'erreurs de reconnaissance de phonèmes, alors que le modèle de l'arabe passe de $65,1\%$ à $5,1\%$ d'erreurs. Lorsque peu de données d'apprentissage sont disponibles, il est donc possible d'obtenir de bons résultats de reconnaissance à l'aide d'une méthode de validation croisée.

Détection des erreurs de phonétisation

Comme pour le français, la détection des erreurs de phonétisation consiste à comparer les hypothèses de transcription phonétique à partir du texte et à partir du signal. Nous utilisons pour cela les modèles de reconnaissance adaptés aux locuteurs à transcrire. Après un alignement des transcriptions selon l'algorithme de Levenshtein, les phonèmes obtenus à partir du texte qui ne correspondent pas à ceux transcrits à partir du signal sont annotés comme erronés. Le tableau 6.12 donne les

résultats de détection des erreurs de phonétisation à partir du texte en terme de Précision, Rappel et MCR, obtenus en moyenne par les modèles adaptés sur $TRAIN_1^+$ et $TRAIN_2^+$.

Langue	Précision	Rappel	MCR
Italien	7,8	63,0	10,4
Arabe	42,9	98,1	8,3

TABLE 6.12 – Évaluation de la détection d'erreurs en italien et en arabe avec les systèmes de reconnaissance de phonèmes

Nous observons que pour l'italien, la Précision est assez faible et le Rappel est seulement de $63,0\%$ pour $10,4\%$ de phonèmes à vérifier. Ce score, plus faible que celui observé en français, peut être expliquée par le manque de données pour estimer les modèles acoustiques. Cependant, la phonétisation à partir du texte faisant moins d'erreur qu'en français, la méthode reste valable pour éliminer la plupart des erreurs après validation manuelle.

En ce qui concerne l'arabe, nous observons de très bons résultats de détection d'erreurs avec $98,1\%$ de Rappel pour seulement $8,3\%$ de données à valider. La problématique liée aux voyelles est particulièrement bien gérée par la modélisation acoustique et confirme la complémentarité des deux systèmes de phonétisation. La méthode de détection d'erreurs à partir de l'analyse du signal donne donc des résultats inégaux pour les différentes langues selon les problématiques rencontrées, bien que les résultats soient dans tous les cas exploitables pour réduire le coût de validation manuelle.

6.4.2 Création d'une voix pour une nouvelle langue

Dans le cas de la création d'une voix pour une nouvelle langue, il est nécessaire de construire un phonétiseur pour la nouvelle langue avant de procéder à la création d'un script de lecture puis de l'enregistrement d'un locuteur. Ce n'est qu'après ces étapes que nous pourrons nous intéresser à la détection et à la correction des erreurs de phonétisation. Le turc a été choisi car il s'agit d'une langue relativement bien documentée mais que nous n'avions pas encore traitée. Ainsi, tous les aspects de la construction d'un système de synthèse de la parole pour une nouvelle langue pourront être abordés.

Ce travail s'inscrit dans le projet MAGMAT (Méthodologie et Architecture Générique de développement Multilingue Accéléré pour la Traduction parole-parole) qui vise à définir et mettre en œuvre une méthodologie agile et incrémentale de développement en temps contraint d'un système de traduction de la parole vers la parole. Ce projet collaboratif, financé par la DGA (Direction Générale de l'Armement) et la DGE (Direction Générale des Entreprises), a réuni le LIUM (Laboratoire d'Informatique de l'Université du Mans), Voxygen et Airbus. Dans ce projet, nous cherchions à maximiser les interactions entre les différents systèmes dans leur conception et à minimiser le coût humain pour l'annotation des données.

Caractéristiques de la langue turque

Le turc est une langue de la famille des langues turciques, du groupe des langues altaïques. Elle est parlée principalement en Turquie et à Chypre. On compte près de 80 millions de locuteurs dans le monde dont c'est la langue maternelle.

Typologiquement, c'est une langue agglutinante, principalement au moyen de suffixes. Cela signifie que les unités sémantiques sont construites en apposant des affixes les uns à la suite des autres. Cette grande richesse morphologique implique que le locuteur a une grande liberté pour former des noms composés. Ainsi, la taille du vocabulaire pour former des lexiques couvrant largement la langue est nécessairement importante. L'analyse morpho-syntaxique est d'une part rendue plus difficile par ce point, mais est d'autre part facilitée par la structure grammaticale. En effet, l'ordre des mots est régulier et suit la séquence *sujet-objet-verbe*. De plus, il n'y a ni classes nominales ni genre grammatical. Il n'existe donc pas par exemple de distinction entre masculin et féminin pour les noms, les adjectifs, les participes, les pronoms et les verbes. Seul le genre des personnes peut parfois être précisé explicitement par un mot lorsque le contexte le demande.

Par ailleurs, le turc est considéré comme ayant majoritairement une écriture phonétique, c'est-à-dire que chaque graphème correspond à un phonème. Il est donc possible de définir des règles simples de phonétisation à partir du texte. Cependant, les mots empruntés à d'autres langues, comme les mots arabes qui sont particulièrement courants, suivent des règles de prononciation différentes. Il en est de même pour l'accent lexical, placé sur la dernière syllabe pour les mots d'origine turque, mais suivant des règles différentes pour les emprunts lexicaux. Comme il est difficile de distinguer l'origine des mots, il est difficile de déterminer la phonétique des mots dont la prononciation n'est pas standard.

De plus, le turc comporte un système d'harmonie vocalique : les voyelles des affixes changent en fonction des voyelles du mot racine. Il n'est donc pas possible de donner une prononciation canonique pour les suffixes sans connaître les mots auquels ils se rattachent. Cependant, cela ne pose pas de problème pour un phonétiseur dans la mesure où l'orthographe des mots est également modifiée et suit la prononciation.

En somme, les caractéristiques du turc ont des avantages et des inconvénients pour la synthèse de la parole. La bonne correspondance entre graphèmes et phonèmes réduit les contraintes dans le développement d'un phonétiseur, mais la grande richesse morphologique et l'irrégularité des emprunts lexicaux réduit l'intérêt des approches d'analyse morpho-syntaxique. Considérant ces caractéristiques, deux approches nous paraissent pertinentes pour la création d'un phonétiseur : construire un phonétiseur à base de règles couplé à un lexique d'exceptions, et développer un phonétiseur par apprentissage à partir données conçues pour la reconnaissance de la parole. Pour la première approche, nous nous appuierons sur des études sur la phonétisation du turc décrites dans la littérature. Pour la seconde approche, nous construirons dans un premier temps un système de reconnaissance de la parole pour ensuite procéder à un alignement forcé des données d'apprentissage. Ceci nous permettra d'obtenir un corpus d'apprentissage pour estimer un modèle de phonétisation en contexte.

Ressources existantes

Le turc est une langue bien documentée et de nombreuses ressources sont disponibles, aussi bien au niveau des données acoustiques et textuelles que des dictionnaires de prononciation ou des analyseurs morphologiques. Parmi celles-ci, quelques-unes sont particulièrement adaptées à la construction de systèmes de reconnaissance et de synthèse de parole.

Une ressource importante dans notre étude est le corpus de données du projet Babel (ANDRESEN et al., 2016) Il s'agit d'enregistrements de conversations téléphoniques avec les transcriptions correspondantes qui ont été collectées pour la construction de systèmes de reconnaissance de la parole. Les données sont divisées en un corpus TRAIN pour l'apprentissage de modèles acoustiques et un corpus TEST pour leur évaluation. Le tableau 6.13 récapitule la répartition des données disponibles. Les quantités de données sont reportées en temps de parole strict, c'est-à-dire sans compter les temps de silence éventuels.

Corpus	Durée (h)	Segments	Mots
TRAIN	109,1	98 545	642 381
TEST	9,8	10 203	71 332

TABLE 6.13 – Répartition des données acoustiques du projet Babel

Un lexique est également fourni, donnant la prononciation associée aux mots des transcriptions. Sur les 47 470 mots du corpus TRAIN, 45 512 sont dans le lexique, soit une couverture de 95,9%. Quant au corpus TEST, 9 932 sur 10 306 mots sont dans le lexique, soit une couverture de 96,4%. Le lexique comporte 48 165 mots différents, avec en moyenne 1,13 variantes de prononciation par mot. Ainsi, nous observons que le lexique couvre très bien les données d'apprentissage et d'évaluation.

La reconnaissance de la parole en turc a également été traitée dans (ARISOY et al., 2009). Les auteurs présentent un système de reconnaissance de la parole pour la transcription de données journalistiques. Pour cela, ils s'appuient sur un corpus d'émissions de radio (SARAÇLAR, 2012). Une particularité de leur système est de considérer des unités sous-lexicales plutôt que des mots. De cette façon, ils limitent la taille du vocabulaire tout en bénéficiant d'une bonne couverture lexicale. Suivant le même objectif, les auteurs de (CAN et ARTUNER, 2013) proposent un système de reconnaissance du turc fondé sur des syllabes. Nous conservons cependant l'utilisation des mots comme unité linguistique de base pour la reconnaissance de parole. En effet, sous sommes dans une situation ou le vocabulaire reste limité.

Le corpus d'émissions de radio cité précedemment (SARAÇLAR, 2012) est composé de 68 434 segments de parole, totalisant 130h d'enregistrements et 538 462 occurrences de mots. En outre, on retrouve 54 269 mots uniques. Cependant, aucun lexique de prononciation n'est fourni, et seulement 11 815 mots sont présents dans le lexique du projet Babel, soit un couverture de 21,8%. Pour exploiter ces données pour la reconnaissance de la parole, il est donc nécessaire d'enrichir le lexique avec la prononciation de nouveaux mots.

En ce qui concerne la phonétisation elle-même, plusieurs études ont proposé d'utiliser des règles de phonétisation (OFLAZER et INKELAS, 2003 ; OFLAZER et INKELAS, 2006 ; ALTINOK, 2016). Ces dernières s'appuient en partie sur une analyse morphologique du texte pour désambiguiser les mots. On trouve dans la littérature de nombreux travaux sur l'analyse morphologique (OFLAZER, 1994 ; GÜNGÖR, 1995 ; SAK,

GÜNGÖR et SARAÇLAR, 2008 ; CÖLTEKIN, 2010 ; CÖLTEKIN, 2014). Elle est surtout utile pour la phonétisation des noms propres ou des mots empruntés à des langues étrangères.

Jeu de phonèmes

Un des points clés dans la mutualisation des ressources et des outils pour la reconnaissance et la synthèse de parole est de définir un unique jeu de phonèmes. Ainsi, nous pourrons bénéficier des modèles acoustiques construits pour préparer des données utiles à la création de voix de synthèses. Dans la suite, nous utilisons la notation SAMPA [1] pour écrire les phonèmes.

Le turc possède 8 voyelles, notées `{a, e, ı, i, o, ö, u, ü}` qui ont respectivement pour prononciation `/a, e, 1, i, o, 2, u, y/`. Chacune des voyelles peut être courte ou longue, mais `/o:/` et `/2:` ne se retouvent que dans les abbréviations. Les consonnes sont quant à elles au nombre de 26 : /p, t, tS, k, c, b, d, dZ, g, gj, f, s, S, v, w, z, Z, m, n, N, l, 5, r, j, h, G/ Parmi ces consonnes, on trouve trois paires d'allophones, respectivement palatal et non-palatal : `/c, k/`, `/l, 5/` et `/gj, g/`. Nous retenons un unique symbole pour chaque paire d'allophones, la distinction pouvant se faire au niveau des unités acoustiques. Nous obtenons ainsi un jeu de 39 phonèmes.

Phonétiseur à base règles

Le phonétiseur à base de règles s'appuie sur l'architecture du système de Voxygen en reprenant les composants du phonétiseur décrit dans (ALTINOK, 2016). Le système se fonde sur un dictionnaire de prononciation d'une part, et un ensemble de règles de conversion de graphèmes à phonèmes d'autre part.

Lorsqu'un mot de la séquence à phonétiser est absent du dictionnaire, les règles sont appliquées pour obtenir la séquence phonétique. Pour simplifier le développement nous ne procédons pas à une analyse morphologique. Le lexique suffit en effet à couvrir la plupart des exceptions.

Phonétiseur neuronal

Nous souhaitons comparer le phonétiseur à base de règles à un phonétiseur appris sur des données phonétisées. Nous avons choisi de développer un système de phonétisation automatique neuronal de bout en bout comme décrit à la section 6.2.2. Pour développer ce phonétiseur traitant les mots dans leur contexte lexical, il est nécessaire de constituer un corpus d'apprentissage parallèle avec des séquences de mots transcrites en phonèmes. Parmi les données disponibles, un tel corpus n'existe pas. Cependant, nous disposons des données du projet Babel, conçues pour l'apprentissage de modèles acoustiques de reconnaissance de la parole.

Nous developpons en premier lieu un système de reconnaissance de la parole construit à l'aide de *Kaldi* (POVEY et al., 2011). Nous avons tout d'abord construit un modèle acoustique HMM/GMM traitant le signal comme des vecteurs de coefficients

1. https ://www.phon.ucl.ac.uk/home/sampa/turkish.htm

MFCC adaptés aux locuteurs par fMLLR. Ce dernier est estimé sur la subdivision TRAIN du corpus Babel. Ensuite, nous avons estimé, à partir des alignements phonémiques produits par le modèle HMM/GMM, un modèle TDNN-LSTM/HMM avec des couches cachées de dimension 300. Celui-ci prend en entrée des vecteurs de 40 coefficients MFCC haute-résolution et des i-vecteurs de dimension 100 pour l'adaptation aux locuteurs. Associé à un modèle de langage, le modèle acoustique obtenu forme un système de reconnaissance de la parole qui peut être utilisé à part entière.

Une fois les modèles acoustiques estimés, nous avons procédé à un alignement forcé du corpus au niveau phonèmes en se fondant sur le lexique accompagnant les transcriptions de parole. Ainsi, nous avons obtenu la transcription phonétique de tous les segments du corpus du projet Babel. Autrement dit, nous nous appuyons sur le système de reconnaissance de la parole et les données ayant permis sont apprentissage pour construire le corpus d'apprentissage d'un phonétiseur.

Finalement, nous estimons les paramètres du phonétiseur neuronal sur le corpus de transcriptions alignées. Nous obtenons alors un phonétiseur capable de traiter des séquences de mots dans leur contexte grâce aux couches récurrentes et au mécanisme d'attention du modèle neuronal.

Comparaison des phonétiseurs

Afin de comparer et d'évaluer les phonétiseurs, 1000 phrases ont été sélectionnées et transcrites en phonèmes manuellement. Cela correspond à un corpus parallèle de 10 562 mots. De plus, les mots portant des subtilités de prononciation ont été annotés et caractérisés. Les phrases ont été choisies pour les difficultés phonétiques qu'elles comportent. En effet, nous avons vu que même si l'écriture de la langue turque est réputée phonétique, il existe de nombreuses exceptions problématiques pour la tâche de phonétisation. Le tableau 6.14 donne la liste des marqueurs qui ont été utilisés et leur fréquence d'apparition dans le corpus.

Marqueur	Description	Occurrences	Exemple
SW	Allongement mot	1090	`şayan => /S AA j A n/`
LVS	Allongement dernière voyelle	331	`hayatında => /h A j AA t W n d A/`
FW	Mot étranger	87	`Paris => /p AA r i s/`
PNA	Nom propre avec suffixe	236	`Filiz'e => /f i l i z e/`
HOM	Homographe hétérophone	185	`aşık olan => /AA S W k | o l a n/` `aşık atan => /A S W k | A t A n/`
OW	Mot ancien	73	`misak-i milli` `=> /m ii s AA k | i | m i l l ii/`
ACR	Acronyme	57	`CHP'yi => /dZ ee h ee p ee j i/`
VI	Insertion voyelle	43	`program => /p W r o g r A m/`

TABLE 6.14 – Marqueurs d'annotation des mots présentant une phonétique irrégulière et leur nombre d'occurrences dans le corpus

Nous remarquons que 20% des mots sont annotés comme ayant une phonétique irrégulière.

Nous évaluons ensuite les deux phonétiseurs sur la tâche de phonétisation de ce corpus. En moyenne, le taux d'erreur de phonétisation (PER) du phonétiseur à base de règles de réécriture est de 3,4%. Ce taux d'erreur est calculé au niveau phonèmes,

c'est-à-dire que chaque phonème erroné est comptabilisé. Pour le phonétiseur neuronal, ce taux d'erreur est de 7,7%.

Nous nous intéressons ensuite au taux d'erreur de phonétisation en fonction de la classe des mots. Le tableau 6.15 donne ce taux d'erreur de phonétisation au niveau mot en fonction du type d'erreur. Autrement dit, chaque mot comprenant au moins un phonème erroné est considéré comme erroné.

Phonétiseur	SW	LVS	FW	PNA	HOM	OW	ACR	VI	NULL
à base de règles	16,7	39,9	66,1	16,9	43,2	71,2	61,4	2,3	2,2
neuronal	74,0	65,3	65,9	19,1	61,1	80,8	75,4	16,3	15,5

TABLE 6.15 – Taux d'erreur de phonétisation au niveau mot en fonction du type d'erreur. L'étiquette NULL correspond aux mots non classés.

Bien que le phonétiseur à base de règles soit plus performant en moyenne que le phonétiseur neuronal, ce dernier obtient des performances similaires pour les mots étrangers et les noms propres. De plus, nous remarquons que c'est la phonétisation des mots anciens qui pose le plus de problèmes. Les mots étrangers, les homographes hétérophones et les acronymes font également partie des mots les plus difficiles à phonétiser à partir du texte. Nous comprenons que le taux d'erreur de phonétisation varie de manière importante selon le vocabulaire utilisé. Un dictionnaire de prononciation des exceptions les plus fréquentes peut donc aider à réduire la quantité d'erreurs de phonétisation.

Application à la détection des erreurs de phonétisation

Une base de données d'unités acoustiques provenant d'un même locuteur est un composant de base pour la synthèse de la parole. Nous utilisons le phonétiseur à base de règles pour sélectionner des phrases optimisant la couverture de phonèmes en contexte parmi un corpus de textes journalistiques. Nous obtenons un corpus de 2730 phrases comportant en moyenne 8,2 mots par phrase. Un locuteur turc natif est ensuite enregistré sur la lecture de ces phrases, ce qui nous permet d'obtenir 7 heures de parole.

Finalement, nous procédons à une détection automatique des erreurs de phonétisation en comparant les hypothèses du phonétiseur neuronal à celles du phonétiseur à base de règles. Pour estimer le nombre de phonèmes qui peuvent être corrigés grâce à la détection automatique des erreurs, nous procédons à une correction manuelle de 1000 phrases du corpus obtenu. Le tableau 6.16 donne les résultats de la détection des erreurs.

Système	Précision	Rappel	MCR
Phonétiseur neuronal	8,9	49,0	5,8

TABLE 6.16 – Évaluation de la détection d'erreurs avec le système de phonétisation neuronal

Nous observons qu'il est possible de corriger 49,0% des erreurs de phonétisation en vérifiant 5,8% des phonèmes. En corrigeant uniquement les erreurs détectées, cela permet d'améliorer la qualité de l'annotation de la base de données d'unités acoustiques sans compromettre la rapidité de sa création. Il reste cependant la moitié des

erreurs de phonétisation qui ne sont pas corrigées, ce qui peut causer des erreurs de concaténation lors de la synthèse de la parole. Une analyse plus fine des transcriptions phonétiques, notamment en utilisant un autre modèle acoustique, permettrait de détecter davantage d'erreurs.

6.5 Conclusion

Dans ce chapitre, nous avons comparé plusieurs approches permettant d'obtenir la séquence de phonèmes correspondant à une séquence de texte ou un signal de parole. En effet, nous avons mis en œuvre un phonétiseur fondé sur un lexique et des règles de translittération, un phonétiseur statistique, un phonétiseur neuronal, une méthode d'alignement forcé utilisant un modèle acoustique de reconnaissance de la parole, et un système de reconnaissance de la parole neuronal de bout-en-bout. Ce dernier a été modifié de façon à tenir compte d'informations extraites du texte afin de prendre une décision sur les deux modalités. Nous avons montré que quelle que soit la méthode utilisée, des erreurs importantes de transcription subsistaient. Lorsqu'elles sont utilisées pour annoter des bases de données de synthèse de parole, ces dernières peuvent affecter négativement la qualité de tels systèmes. Cela est illustré par le type des erreurs observées, qui correspondent aux problèmes rencontrés lors du développement de système de synthèse de la parole.

Nous avons ensuite proposé une méthode de détection des erreurs de phonétisation en comparant les hypothèses de transcriptions phonétiques des différents systèmes proposés. Nous avons montré que la plupart des erreurs réalisées pouvaient être détectées, tout en limitant la quantité de données à vérifier. Lorsqu'une correction de l'annotation est nécessaire, cela permet de minimiser le temps de validation manuelle.

Nous avons d'abord illustré notre approche sur des données en français, que nous avions en nombre important relativement aux autres langues. Ensuite, nous avons répliqué notre méthode en italien et en arabe, langues pour lesquelles nous avions moins de données. Nous avons alors montré que nous pouvions également détecter les erreurs de phonétisation dans un contexte de langues peu dotées. Nous obtenons de bons résultats de détection d'erreurs en suivant une méthode de validation croisée pour adapter les modèles aux locuteurs des données à vérifier. Finalement, nous avons évalué la méthode de détection d'erreurs de phonétisation lors de la création d'une voix de synthèse pour le turc, une langue qui n'avait pas été traitée auparavant. Nous avons montré qu'en exploitant des données conçues pour la reconnaissance de la parole, nous pouvions construire un corpus d'apprentissage pour un phonétiseur neuronal de mots en contexte. Ceci nous a permis de comparer les hypothèses d'un tel phonétiseur à celles d'un phonétiseur à base de règles, et donc de détecter un nombre important d'erreurs de phonétisation lors de la création d'un corpus de parole transcrit pour la synthèse.

En français le taux d'erreur de phonétisation pour le phonétiseur à base de règles est de 2,9%. En appliquant la méthode de détection des erreurs avec le système de reconnaissance de phonèmes, nous pouvons repérer jusqu'à 81,2% de ces erreurs. Si ces erreurs sont corrigées manuellement, cela nécessite la vérification de 12,2% des phonèmes. Ensuite, le taux d'erreurs résiduelles, qui correspond au PER après correction, serait de 0,54%. De même, on passe en italien d'un taux d'erreur de phonétisation de 1,3% à un taux d'erreurs résiduelles de 0,48%. En arabe, où on observait

un taux d'erreur de 3,8%, le taux d'erreurs résiduelles est égal à 0,07%. Pour le turc, qui ne bénéficie pas de la même méthode de détection des erreurs de phonétisation, le taux d'erreur de la base de données passe de 3,4% à 1,73% après corrections. Dans tous les cas, l'amélioration de la qualité d'annotation des bases de données de parole est significative, en ne vérifiant qu'une fraction des données. L'impact de ces erreurs résiduelles sur la qualité de la synthèse de la parole est difficile à quantifier. Cela revient à estimer l'impact d'une mauvaise prononciation lors de la génération. En synthèse par corpus, cela peut conduire à sélectionner des unités acoustiques inappropriées et donc produire un texte inintelligible. En pratique, la rareté de ces unités mal annotées, grâce à la correction de la plupart des erreurs, conjuguée à la possibilité d'exclure manuellement certaines unités acoustiques du graphe de recherche lors de la génération, permet d'éviter la plupart des défauts. Pour la synthèse paramétrique, les modèles acoustiques sont d'autant mieux estimés que la base de données d'apprentissage contient un nombre réduit d'erreurs d'annotations.

Conclusion

Nous avons présenté dans ce document notre étude sur la construction rapide, performante et mutualisée de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues. Ce travail nous a permis d'aborder différents aspects de notre objet d'étude, la parole, aussi bien sur le plan acoustique que linguistique. Ainsi, nous avons vu que le contenu de la parole pouvait être traduit de manière symbolique et traité comme tel. En partant du texte, des outils permettent d'analyser la structure grammaticale, de normaliser l'orthographe des mots et d'obtenir leur prononciation. En partant du signal, nous pouvons calculer une représentation numérique de l'information véhiculée par la parole. La tâche consistant à passer du texte au signal et réciproquement devient alors un problème de modélisation de séquences. La reconnaissance et la synthèse de la parole peuvent donc utiliser des algorithmes similaires pour résoudre leurs problèmes respectifs.

Un premier axe de travail a été de trouver et d'exploiter des leviers d'accélération dans la construction de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues. Pour la reconnaissance de la parole, une des étapes considérées comme les plus coûteuses est la collecte de données de parole annotées. Cette tâche peut être facilitée en considérant des données annotées automatiquement en partie. Notre participation au *Multi-Genre Broadcast Challenge* a montré que le temps nécessaire pour cette collecte pouvait être réduit en utilisant des données imparfaites issues du sous-titrage de journaux télévisés. L'avantage de cette approche est d'exploiter des données disponibles en très grande quantité, ce qui permet de procéder à une sélection des données selon un critère de fiabilité. Ainsi, nous avons utilisé un système de reconnaissance de la parole préalablement construit à partir de l'ensemble des données imparfaites pour transcrire l'ensemble du signal de parole disponible. En comparant les transcriptions automatiques à celles de référence, nous avons pu isoler les segments de parole transcrite minorant un certain taux d'erreur mot. Pour la synthèse de la parole, c'est également la création de bases de données de signal transcrit en phonèmes qui impacte le plus le temps de développement d'une nouvelle langue. Celle-ci passe par la phonétisation automatique de texte, l'enregistrement d'un locuteur sur la lecture de ce texte, puis la vérification manuelle des transcriptions phonétiques vis-à-vis du signal de parole enregistré. D'une part, nous avons montré qu'un phonétiseur à base de règles et un phonétiseur neuronal fondé sur une architecture encodeur-décodeur obtenaient des résultats similaires sur la tâche de phonétisation des mots en contexte. Ainsi, il est possible de choisir la technologie la plus rapide à développer selon les ressources disponibles pour une nouvelle langue. D'autre part, nous avons proposé une méthode de détection automatique des erreurs de phonétisation qui réduit le temps de validation des bases de données de signal. Notre approche consiste à tenir compte du signal de parole pour obtenir une transcription phonétique complémentaire de celle obtenue à partir du texte.

Un deuxième axe de travail a été d'obtenir de bonnes performances en reconnaissance et en synthèse de la parole quelle que soit la langue. A cet effet, nous avons retenu l'architecture hybride HMM/TDNN pour la modélisation acoustique en reconnaissance de la parole. Cette dernière a permis à notre système pour l'arabe d'être bien classé au *Multi-Genre Broadcast Challenge* et a donné de bons résultats pour d'autres langues également. De plus, nous avons combiné des systèmes utilisant soit des graphèmes, soit des phonèmes pour représenter symboliquement les unités acoustiques. Nous avons observé que les deux représentations sont complémentaires pour la langue arabe et qu'elles permettent entre autres de mieux traiter les voyelles. La reconnaissance de la parole de bout-en-bout, avec l'architecture *Deepspeech 2*, donne également de bonnes performances de reconnaissance, même avec relativement peu de données d'apprentissage. De plus, c'est une approche permettant de développer rapidemment de nouveaux systèmes. En effet, nous avons observé que pour la reconnaissance de phonèmes, nous obtenions un taux d'erreur de 9,9% sur des bases de données en français pour la synthèse de la parole. Ce taux d'erreur a été obtenu en réalisant un apprentissage sur 50 heures de parole. Par ailleurs, nous observons avec des données similaires en italien et en arabe un taux d'erreur de reconnaissance de phonèmes de respectivement 10,3% et 5,1%, les systèmes de reconnaissance ayant été appris sur respectivement 12,3 et 15,1 heures de parole. En synthèse de la parole, la qualité des systèmes obtenus dépend en grande partie des performances du phonétiseur et de la qualité de l'annotation des données. En améliorant ces deux composantes, notre travail permet d'améliorer la qualité des voix de synthèse créées pour de nouvelles langues.

Un troisième axe de travail est celui de la mutualité des systèmes. Dans cette thèse, nous nous sommes efforcés de partager les ressources entre la reconnaissance et la synthèse de la parole, ainsi que les outils. Ainsi, un système de reconnaissance de la parole a permis d'annoter des données pour l'apprentissage d'un phonétiseur. L'utilisation de ce phonétiseur peut à la fois enrichir un lexique de prononciation pour la reconnaissance de la parole, mais aussi transcrire du texte dans un système de synthèse. La reconnaissance de la parole a également permis d'améliorer la qualité des bases de données de signal pour la synthèse grâce à une détection automatique des erreurs de phonétisation. Tout ceci est possible dans la mesure où les systèmes partagent le même jeu de phonèmes, ce qui n'affecte pas les performances pour toutes les langues que nous avons traitées. Enfin, la flexibilité des approches neuronales permet des combinaisons de systèmes qui ne sont pas possibles avec d'autres algorithmes. Les représentations intermédiaires des couches des réseaux de neurones peuvent ainsi être mutualisées entre des systèmes et profiter pour l'apprentissage de ressources destinées à la reconnaissance et à la synthèse de parole.

Comme perspectives, nous pouvons aller plus loin dans la transcription phonétique de bases de données de signal de parole pour la synthèse. En effet, il est envisagé d'expérimenter d'autres architectures neuronales multimodales pour transcrire la parole et le texte en chaînes phonétiques. Par exemple, nous pouvons imaginer une architecture composée de deux encodeurs, un pour le signal et un pour le texte, et d'un unique décodeur pour obtenir les phonèmes. De plus, nous pouvons expérimenter les systèmes de synthèse de la parole entièrement neuronaux, qui pourraient exploiter des données utilisées aujourd'hui pour la reconnaissance de parole exclusivement. De nombreux travaux restent à effectuer pour améliorer la prosodie en synthèse de la parole, une composante difficile modéliser. Les grandes bases de données acoustiques, de plus en plus nombreuses, pourraient profiter en ce sens aux systèmes de synthèse. Par ailleurs, nous pouvons envisager des techniques d'augmentation de

données pour l'apprentissage de modèles acoustique de reconnaissance de la parole en utilisant les sorties d'un système de synthèse. De telles techniques permettraient de mieux adresser les langues pour lesquelles nous disposons de peu de données mais de bonnes connaissances linguistiques.

Bibliographie personnelle

Kévin Vythelingum, Yannick Estève, Olivier Rosec. "Acoustic-dependent Phonemic Transcription for Text-to-speech Synthesis". *Interspeech 2018*, Septembre 2018, Hyderabad, India. <hal-01870866>

Kévin Vythelingum, Yannick Estève, Olivier Rosec. "Transcription phonétique automatique pour la synthèse de la parole". *XXXIIe Journées d'Etudes sur la Parole (JEP 2018)*, Juin 2018, Aix-en-Provence, France. <hal-01761937>

Kévin Vythelingum, Yannick Estève, Olivier Rosec. "Error detection of grapheme-to-phoneme conversion in text-to-speech synthesis using speech signal and lexical context". *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Décembre 2017, Okinawa, Japan. <hal-01585770>

Kévin Vythelingum. "Détection des erreurs de phonétisation pour la synthèse de parole". *17e Journée des Doctorants de l'ED STIM*, Mai 2017, Nantes, France. <hal-01762424>

Natalia Tomashenko, Kévin Vythelingum, Anthony Rousseau, Yannick Estève. "LIUM ASR systems for the 2016 Multi-Genre Broadcast Arabic Challenge". *IEEE Workshop on Spoken Language Technology (SLT)*, Décembre 2016, San Diego, CA, USA. <hal-01433188>

Bibliographie

AlHanai, T., W.-N. Hsu et J. Glass (2016). « Development of the MIT ASR system for the 2016 Arabic multi-genre broadcast challenge ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 299–304.

Ali, A., S. Vogel et S. Renals (2017). « Speech recognition challenge in the wild : Arabic MGB-3 ». In : *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, p. 316–322.

Ali, A., Y. Zhang, P. Cardinal, N. Dahak, S. Vogel et J. Glass (2014). « A complete kaldi recipe for building arabic speech recognition systems ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 525–529.

Ali, A., P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals et Y. Zhang (2016). « The MGB-2 challenge : Arabic multi-dialect broadcast media recognition ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 279–284.

Ali, Mohamed, Moustafa Elshafei, Mansour Al-Ghamdi et Husni Al-Muhtaseb (2009). « Arabic phonetic dictionaries for speech recognition ». In : *Journal of Information Technology Research (JITR)* 2.4, p. 67–80.

Altinok, D. (2016). « Towards Turkish ASR : Anatomy of a rule-based Turkish g2p ». In : *arXiv preprint arXiv :1601.03783*.

Amodei, D., S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan et Z. Zhu (2016). « Deep speech 2 : End-to-end speech recognition in english and mandarin ». In : *International Conference on Machine Learning (ICML)*, p. 173–182.

Andresen, J., A. Bills, E. Dubinski, J. G. Fiscus, B. Gillies, M. Harper, T.J. Hazen, A. Jarrett, B. Roomi, J. Ray, A. Rytting, W. Shen et E. Tzoukermann (2016). « IARPA Babel Turkish Language Pack ». In : *LDC2016S10 web download*. Philadelphia : Linguistic Data Consortium.

Arik, S., M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta et M. Shoeybi (2017). « Deep voice : Real-time neural text-to-speech ». In : *Proceedings of the 34th International Conference on Machine Learning (ICML)*, p. 195–204.

ARISOY, E., D. CAN, S. PARLAK, H. SAK et M. SARAÇLAR (2009). « Turkish broadcast news transcription and retrieval ». In : *Transactions on Audio, Speech, and Language Processing* 17.5, p. 874–883.

BAHDANAU, D., K. CHO et Y. BENGIO (2014). « Neural machine translation by jointly learning to align and translate ». In : *arXiv preprint arXiv :1409.0473*.

BÉCHET, F. (2001a). *LIA TAGG*.

– (2001b). « LIA—PHON : Un système complet de phonétisation de textes ». In : *Traitement automatique des langues (TAL)* 42.1, p. 47–67.

BEERENDS, J. G., C. SCHMIDMER, J. BERGER, M. OBERMANN, R. ULLMANN, J. POMY et M. KEYHL (2013). « Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I—Temporal alignment ». In : *Journal of the Audio Engineering Society* 61.6, p. 366–384.

BELL, A. M. (1867). *Visible Speech : The science of Universal alphabetics*. London : Simpkin, Marshall & Co.

BELL, P., M. J. GALES, T. HAIN, J. KILGOUR, P. LANCHANTIN, X. LIU et P. C. WOODLAND (2015). « The MGB challenge : Evaluating multi-genre broadcast media recognition ». In : *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, p. 687–693.

BENGIO, S. et G. HEIGOLD (2014). « Word embeddings for speech recognition ». In : *Interspeech*.

BENGIO, Y., R. DUCHARME, P. VINCENT et C. JAUVIN (2003). « A neural probabilistic language model ». In : *Journal of machine learning research* 3.2, p. 1137–1155.

BIADSY, Fadi, Nizar HABASH et Julia HIRSCHBERG (2009). « Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules ». In : *Proceedings of human language technologies : The 2009 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, p. 397–405.

BIGI, B. (2011). « A multilingual text normalization approach ». In : *Language and Technology Conference*. Springer, p. 515–526.

BISANI, M. et H. NEY (2008). « Joint-sequence models for grapheme-to-phoneme conversion ». In : *Speech communication* 50.5, p. 434–451.

BLACK, A. W. et K. TOKUDA (2005). « The Blizzard Challenge-2005 : Evaluating corpus-based speech synthesis on common datasets ». In : *Ninth European Conference on Speech Communication and Technology*.

BLACK A. W.and Taylor, P. (1994). « CHATR : a generic speech synthesis system ». In : *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, p. 983–986.

BOURLARD, H. et C. J. WELLEKENS (1987). « Multilayer perceptrons and automatic speech recognition ». In : *Proceedings of the First International Conference on Neural Networks*. T. 4, p. 407–416.

BROGNAUX, S., B. PICART et T. DRUGMAN (2014). « Speech synthesis in various communicative situations : Impact of pronunciation variations ». In : *Interspeech*.

BUCKWALTER, T. (2002). « Arabic transliteration ». In : URL : http://www.qamus.org/transliteration.htm.

CAGLAYAN, O., M. GARCIA-MARTINEZ, A. BARDET, W. ARANSA, F. BOUGARES et L. BARRAULT (2017). « Nmtpy : A flexible toolkit for advanced neural machine translation systems ». In : *The Prague Bulletin of Mathematical Linguistics* 109.1, p. 15–28.

CALLIOPE, L. (1989). *La parole et son traitement automatique*. Masson Paris.

CAN, B. et H. ARTUNER (2013). « A syllable-based Turkish speech recognition system by using time delay neural networks (TDNNs) ». In : *International Conference on Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, p. 219–224.

CHAN, W., N. JAITLY, Q. LE et O. VINYALS (2016). « Listen, attend and spell : A neural network for large vocabulary conversational speech recognition ». In : *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, p. 4960–4964.

CHEN, S. F. et J. GOODMAN (1999). « An empirical study of smoothing techniques for language modeling ». In : *Computer Speech & Language* 13.4, p. 359–394.

CHEVELU, J., D. LOLIVE, S. LE MAGUER et D. GUENNEC (2015). « How to compare tts systems : A new subjective evaluation methodology focused on differences ». In : *Interspeech*.

CHO, K., Van Merriënboer B., C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK et Y. BENGIO (2014). « Learning phrase representations using RNN encoder-decoder for statistical machine translation ». In : *arXiv preprint arXiv :1406.1078*.

CÖLTEKIN, C. (2010). « A Freely Available Morphological Analyzer for Turkish. » In : *LREC*. T. 2, p. 19–28.

– (2014). « A set of open source tools for Turkish natural language processing. » In : *LREC*, p. 1079–1086.

CYBENKO, G. (1989). « Approximations by superpositions of sigmoidal functions ». In : *Mathematics of Control, Signals, and Systems*, p. 303–314.

DALL, R., S. BROGNAUX, K. RICHMOND, C. VALENTINI-BOTINHAO, G. E. HENTER, J. HIRSCHBERG, J. YAMAGISHI et S. KING (2016). « Testing the consistency assumption : Pronunciation variant forced alignment in read and spontaneous speech synthesis ». In : *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, p. 5155–5159.

DAVIS, S. et P. MERMELSTEIN (1980). « Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences ». In : *IEEE transactions on acoustics, speech, and signal processing* 28.4, p. 357–366.

DE BROSSES, C. (1765). *Traité de la formation mécanique des langues et des principes physiques de l'étymologie*. T. 1. Paris : Saillant-Vincent-Desaint.

DELATTRE, Pierre (1966). « Les Dix Intonations de base du français ». In : *The French Review* 40.1, p. 1–14.

DEMPSTER, A. P., N. M. LAIRD et D. B. RUBIN (1977). « Maximum likelihood from incomplete data via the EM algorithm ». In : *Journal of the Royal Statistical Society* 39.1, p. 1–22.

DI CRISTO, A. (2000). « Interpréter la prosodie ». In : *XXIIe Journées d'Etudes sur la Parole (JEP)*.

DIGALAKIS, V. V., D. RTISCHEV et L. G. NEUMEYER (1995). « Speaker adaptation using constrained estimation of Gaussian mixtures ». In : *IEEE Transactions on speech and Audio Processing* 3.5, p. 357–366.

DIXON, N. et H. MAXEY (1968). « Terminal analog synthesis of continuous speech using the diphone method of segment assembly ». In : *IEEE transactions on Audio and Electroacoustics* 16.1, p. 40–50.

DUCHI, J., E. HAZAN et Y. SINGER (2011). « Adaptive subgradient methods for online learning and stochastic optimization ». In : *Journal of Machine Learning Research* 12.Jul, p. 2121–2159.

ELMAN, Jeffrey L (1990). « Finding structure in time ». In : *Cognitive science* 14.2, p. 179–211.

ESPIC, F., A. GOVENDER, S. RIBEIRO, C. VALENTINI-BOTINHAO et O. WATTS (2018). « The CSTR entry to the 2018 Blizzard Challenge ». In : *Blizzard Challenge Workshop*.

FANT, G. (1970). *Acoustic theory of speech production*. 2. Walter de Gruyter.

FLANAGAN, J. L. (1972). « Voices of men and machines ». In : *The Journal of the Acoustical Society of America* 51.5A, p. 1375–1387.

FÓNAGY, Iván (2003). « Des fonctions de l'intonation : Essai de synthèse ». In : *Flambeau* 29, p. 1–20.

GALES, M. J. F. (1998). « Maximum likelihood linear transformations for HMM-based speech recognition ». In : *Computer speech & language* 12.2, p. 75–98.

GALESCU, L. et J. F. ALLEN (2001). « Bi-directional conversion between graphemes and phonemes using a joint n-gram model ». In : *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

– (2002). « Pronunciation of proper names with a joint n-gram model for bi-directional grapheme-to-phoneme conversion ». In : *Seventh International Conference on Spoken Language Processing*.

GAUVAIN, J.-L. et C.-H. LEE (1994). « Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains ». In : *IEEE transactions on speech and audio processing* 2.2, p. 291–298.

GHANNAY, Sahar (2017). « Etude sur les représentations continues de mots appliquées à la détection automatique des erreurs de reconnaissance de la parole ». Thèse de doct. Le Mans.

GIBIANSKY, A., S. ARIK, G. DIAMOS, J. MILLER, K. PENG, W. PING, J. RAIMAN et Y. ZHOU (2017). « Deep voice 2 : Multi-speaker neural text-to-speech ». In : *Advances in neural information processing systems*, p. 2962–2970.

GOODFELLOW, I. et A. BENGIO Y. Courville (2016). *Deep Learning*. mit press.

GORMAN, K. et R. SPROAT (2016). « Minimally supervised number normalization ». In : *Transactions of the Association for Computational Linguistics* 4, p. 507–519.

GRAVES, A. et N. JAITLY (2014). « Towards end-to-end speech recognition with recurrent neural networks ». In : *International Conference on Machine Learning (ICML)*, p. 1764–1772.

GRAVES, A., S. FERNÁNDEZ, F. GOMEZ et J. SCHMIDHUBER (2006). « Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks ». In : *Proceedings of the 23rd international conference on Machine learning*. ACM, p. 369–376.

GRETTER, R. (2014). « Euronews : a multilingual speech corpus for ASR. » In : *LREC*, p. 2635–2638.

GÜNGÖR, T. (1995). « Computer processing of Turkish : Morphological and lexical investigation ». In : *Ph.D. Thesis*.

HALLAHAN, W. I. (1995). « DECtalk software : Text-to-speech technology and implementation ». In : *Digital Technical Journal* 7.4, p. 5–19.

HAN, B. et T. BALDWIN (2011). « Lexical normalisation of short text messages : Makn sens a# twitter ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies-Volume 1*. Association for Computational Linguistics, p. 368–378.

HARRIS, C. M. (1953). « A study of the building blocks in speech ». In : *The Journal of the Acoustical Society of America* 25.5, p. 962–969.

HERMANSKY, H. (1990). « Perceptual linear predictive (PLP) analysis of speech ». In : *the Journal of the Acoustical Society of America* 87.4, p. 1738–1752.

HERNANDEZ, F., V. NGUYEN, S. GHANNAY, N. TOMASHENKO et Y. ESTÈVE (2018). « TED-LIUM 3 : twice as much data and corpus repartition for experiments on speaker adaptation ». In : *International Conference on Speech and Computer*. Springer, p. 198–208.

HINTON, G., L. DENG, D. YU, G. DAHL, A. MOHAMED, N. JAITLY, A. SENIOR, V. VANHOUCKE, P. NGUYEN, B. KINGSBURY et T. SAINATH (2012). « Deep neural networks for acoustic modeling in speech recognition ». In : *IEEE Signal processing magazine* 29.

HOCHREITER, S. et J. SCHMIDHUBER (1997). « Long short-term memory ». In : *Neural computation* 9.8, p. 1735–1780.

HORNIK, K. (1991). « Approximation Capabilities of Multilayer Feedforward Networks ». In : *Neural Networks*, p. 251–257.

HUBEL, D. H. et T. N. WIESEL (1962). « Receptive fields, binocular interaction and functional architecture in the cat's visual cortex ». In : *The Journal of physiology* 160.1, p. 106–154.

HUNT, A. J. et A. W. BLACK (1996). « Unit selection in a concatenative speech synthesis system using a large speech database ». In : *International Conference on Acoustics, Speech, and Signal Processing Conference (ICASSP)*. T. 1. IEEE, p. 373–376.

JELINEK, F. (1976). « Continuous speech recognition by statistical methods ». In : *Proceedings of the IEEE* 64.4, p. 532–556.

JIANG, Y., Z.-H. LING, M. LEI, C.-C. WANG, L. HENG, Y. HU, L.-R. DAI et R.-H. WANG (2018). « The ustc system for Blizzard Challenge 2018 ». In : *Blizzard Challenge Workshop*.

JORDAN, M. (1986). « Attractor dynamics and parallelism in a connectionist sequential machine ». In : *Proc. of the Eighth Annual Conference of the Cognitive Science Society*.

KALCHBRENNER, N., E. ELSEN, K. SIMONYAN, S. NOURY, N. CASAGRANDE, E. LOCKHART, F. STIMBERG, A. Van den OORD, S. DIELEMAN et K. KAVUKCUOGLU (2018). « Efficient neural audio synthesis ». In : *arXiv preprint arXiv :1802.08435*.

KAPLAN, R. M. et M. KAY (1994). « Regular models of phonological rule systems ». In : *Computational linguistics* 20.3, p. 331–378.

KATZ, S. (1987). « Estimation of probabilities from sparse data for the language model component of a speech recognizer ». In : *IEEE transactions on acoustics, speech, and signal processing* 35.3, p. 400–401.

KHURANA, S. et A. ALI (2016). « QCRI advanced transcription system (QATS) for the Arabic multi-dialect broadcast media recognition : MGB-2 challenge ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 292–298.

KING, S., J. CRUMLISH, A. MARTIN et L. WIHLBORG (2018). « The Blizzard Challenge 2018 ». In : *Blizzard Challenge Workshop*.

KINGMA, D. P. et J. BA (2014). « Adam : A method for stochastic optimization ». In : *arXiv preprint arXiv :1412.6980*.

KLATT, D. (1982). « The Klattalk text-to-speech conversion system ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 7. IEEE, p. 1589–1592.

KNESER, R. et H. NEY (1995). « Improved backing-off for m-gram language modeling ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 1. IEEE, p. 181–184.

KOEHN, P., H. HOANG, A. BIRCH, C. CALLISON-BURCH, M. FEDERICO, N. BERTOLDI, B. COWAN, W. SHEN, C. MORAN, R. ZENS, C. DYER, O. BOJAR, A. CONSTANTIN et E. HERBST (2007). « Moses : Open source toolkit for statistical machine translation ». In : *Proceedings of the 45th annual meeting of the association for computational linguistics*, p. 177–180.

LAURENT, A., P. DELÉGLISE et S. MEIGNIER (2009). « Grapheme to phoneme conversion using an SMT system ». In : *Interspeech*.

LECUN, Y., K. KAVUKCUOGLU et C. FARABET (2010). « Convolutional networks and applications in vision ». In : *Proceedings of IEEE International Symposium on Circuits and Systems*. IEEE, p. 253–256.

LEGGETTER, Christopher J et Philip C WOODLAND (1995). « Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models ». In : *Computer speech & language* 9.2, p. 171–185.

LÉON, Pierre R. (1993). *Précis de phonostylistique : parole et expressivité*. Nathan.

LIU F.and Weng, F. et X. JIANG (2012). « A broad-coverage normalization system for social media language ». In : *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*. Association for Computational Linguistics, p. 1035–1044.

LU, L., X. ZHANG, K. CHO et S. RENALS (2015). « A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition ». In : *Interspeech*.

MASUKO, T., K. TOKUDA, T. KOBAYASHI et S. IMAI (1996). « Speech synthesis using HMMs with dynamic features ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 1. IEEE, p. 389–392.

MERMELSTEIN, P. (1973). « Articulatory model for the study of speech production ». In : *The Journal of the Acoustical Society of America* 53.4, p. 1070–1082.

MIAO, Y., M. GOWAYYED et F. METZE (2015). « EESEN : End-to-end speech recognition using deep RNN models and WFST-based decoding ». In : *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, p. 167–174.

MIKOLOV, T., M. KARAFIÁT, L. BURGET, J. ČERNOCKỲ et S. KHUDANPUR (2010). « Recurrent neural network based language model ». In : *Interspeech*.

MIKOLOV, T., K. CHEN, G. CORRADO et J. DEAN (2013). « Efficient estimation of word representations in vector space ». In : *arXiv preprint arXiv :1301.3781*.

MONTMIGNON, J.-B. (1785). *Système de prononciation figurée applicable à toutes les langues*. Paris : Royez.

MOULINES, E. et F. CHARPENTIER (1990). « Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones ». In : *Speech communication* 9.5-6, p. 453–467.

NG, A. H., K. GORMAN et R. SPROAT (2017). « Minimally supervised written-to-spoken text normalization ». In : *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, p. 665–670.

NGUYEN, L., T. NG, K. NGUYEN, R. ZBIB et J. MAKHOUL (2009). « Lexical and phonetic modeling for Arabic automatic speech recognition ». In : *Interspeech*.

NOVAK, J. R., N. MINEMATSU et K. HIROSE (2013). « Failure transitions for joint n-gram models and G2P conversion ». In : *Interspeech*, p. 1821–1825.

NOVAK, J. R., P. R. DIXON, N. MINEMATSU, K. HIROSE, C. HORI et H. KASHIOKA (2012). « Improving WFST-based G2P conversion with alignment constraints and RNNLM N-best rescoring ». In : *Interspeech*.

OFLAZER, K. (1994). « Two-level description of Turkish morphology ». In : *Literary and Linguistic computing* 9.2, p. 137–148.

OFLAZER, K. et S. INKELAS (2003). « A finite state pronunciation lexicon for Turkish ». In : *Proceedings of the EACL Workshop on Finite State Methods in NLP*. T. 82, p. 900–918.

– (2006). « The architecture and the implementation of a finite state pronunciation lexicon for Turkish ». In : *Computer Speech & Language* 20.1, p. 80–106.

OORD, A. Van den, S. DIELEMAN, H. ZEN, K. SIMONYAN, O. VINYALS, A. GRAVES, N. KALCHBRENNER, A. SENIOR et K. KAVUKCUOGLU (2016). « Wavenet : A generative model for raw audio ». In : *arXiv preprint arXiv :1609.03499*.

PALLETT, D., J. FISCUS, J. GAROFOLO, A. MARTIN, M. PRZYBOCKI, J. AJOT et B. TJADEN (2009). « The history of automatic speech recognition evaluations at NIST ». In : URL : http://www.itl.nist.gov/iad/mig/publications/ASRhistory.

PASHA, A., M. AL-BADRASHINY, M. T. DIAB, A. EL KHOLY, R. ESKANDER, N. HABASH, Pooleery M., Rambow O. et R. M. ROTH (2014). « Madamira : A fast, comprehensive tool for morphological analysis and disambiguation of arabic. » In : *LREC*. T. 14, p. 1094–1101.

PEDDINTI, V., D. POVEY et S. KHUDANPUR (2015). « A time delay neural network architecture for efficient modeling of long temporal contexts ». In : *Interspeech*.

PENNELL, Deana et Yang LIU (2011). « A character-level machine translation approach for normalization of sms abbreviations ». In : *Proceedings of 5th International Joint Conference on Natural Language Processing*, p. 974–982.

PENNINGTON, J., R. SOCHER et C. MANNING (2014). « Glove : Global vectors for word representation ». In : *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

PETERSON, G. E., W. S.-Y. WANG et E. SIVERTSEN (1958). « Segmentation techniques in speech synthesis ». In : *The Journal of the Acoustical Society of America* 30.8, p. 739–742.

PING, W., K. PENG, A. GIBIANSKY, S. O. ARIK, A. KANNAN, S. NARANG, J. RAIMAN et J. MILLER (2017). « Deep voice 3 : Scaling text-to-speech with convolutional sequence learning ». In : *arXiv preprint arXiv :1710.07654*.

POVEY, D., B. KINGSBURY, L. MANGU, G. SAON, H. SOLTAU et G. ZWEIG (2005). « fMPE : Discriminatively trained features for speech recognition ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 1. IEEE, p. I–961.

POVEY, D., A. GHOSHAL, G. BOULIANNE, L. BURGET, O. GLEMBEK, N. GOEL, M. HANNEMANN, P. MOTLICEK, Y. QIAN, P. SCHWARZ, J. SILOVSKY, G. STEMMER et K. VESELY (2011). « The Kaldi speech recognition toolkit ». In : *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Signal Processing Society.

POVEY, D., V. PEDDINTI, D. GALVEZ, P. GHAHREMANI, V. MANOHAR, X. NA, Y. WANG et S. KHUDANPUR (2016). « Purely sequence-trained neural networks for ASR based on lattice-free MMI. » In : *Interspeech*, p. 2751–2755.

RABINER, L. R. (1989). « A tutorial on hidden Markov models and selected applications in speech recognition ». In : *Proceedings of the IEEE* 77.2, p. 257–286.

RAO, K., F. PENG, H. SAK et F. BEAUFAYS (2015). « Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks ». In : *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, p. 4225–4229.

RIX, A. W., J. G. BEERENDS, M. P. HOLLIER et A. P. HEKSTRA (2001). « Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 2. IEEE, p. 749–752.

ROUSSEAU, A., P. DELÉGLISE et Y. ESTÈVE (2012). « TED-LIUM : an Automatic Speech Recognition dedicated corpus. » In : *LREC*, p. 125–129.

– (2014). « Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. » In : *LREC*, p. 3935–3939.

SAGISAKA, Y. (1988). « Speech synthesis by rule using an optimal selection of non-uniform synthesis units ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, p. 679–682.

SAK, H., T. GÜNGÖR et M. SARAÇLAR (2008). « Turkish language resources : Morphological parser, morphological disambiguator and web corpus ». In : *International Conference on Natural Language Processing*. Springer, p. 417–427.

SARAÇLAR, M. (2012). « Turkish Broadcast News Speech and Transcripts ». In : *Web download*. Philadelphia : Linguistic Data Consortium.

SCHWARM, S. et M. OSTENDORF (2002). « Text normalization with varied data sources for conversational speech language modeling ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 1. IEEE, p. I–789.

SCHWEITZER, C., C. DODANE et J. LAZAR (2018). « L'histoire des alphabets phonétiques du XVIIIe siècle jusqu'à l'API ». In : *XXXIIe Journées d'Etudes sur la Parole (JEP)*.

SCHWENK, H. (2007). « Continuous space language models ». In : *Computer Speech & Language* 21.3, p. 492–518.

SHEN, J., R. PANG, R. J. WEISS, M. SCHUSTER, N. JAITLY, Z. YANG, Z. CHEN, Y. ZHANG, Y. WANG, R. J. SKERRY-RYAN, R. A. SAUROUS, Y. AGIOMYRGIANNAKIS et W. YONGHUI (2018). « Natural tts synthesis by conditioning wavenet on mel spectrogram predictions ». In : *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, p. 4779–4783.

SHIBATA, Y., T. KIDA, S. FUKAMACHI, M. TAKEDA, A. SHINOHARA, T. SHINOHARA et S. ARIKAWA (1999). *Byte Pair encoding : A text compression scheme that accelerates pattern matching*. Rapp. tech. Technical Report DOI-TR-161, Department of Informatics, Kyushu University.

SOLTAU, H., G. SAON, B. KINGSBURY, H.-K. J. KUO, L. MANGU, D. POVEY et A. EMAMI (2009). « Advances in Arabic speech transcription at IBM under the DARPA GALE program ». In : *IEEE Transactions on Audio, Speech, and Language processing* 17.5, p. 884–894.

SPROAT, R., A. W. BLACK, S. CHEN, S. KUMAR, M. OSTENDORF et C. RICHARDS (2001). « Normalization of non-standard words ». In : *Computer speech & language* 15.3, p. 287–333.

STEVENS, S. S., J. VOLKMANN et E. B. NEWMAN (1937). « A scale for the measurement of the psychological magnitude pitch ». In : *The Journal of the Acoustical Society of America* 8.3, p. 185–190.

STOLCKE, A. (2002). « SRILM - an extensible language modeling toolkit ». In : *Seventh International Conference on Spoken Language Processing*.

STORY, B. H. (2009). « Advances in simulation of sentence-level speech production with kinematic models of the vocal tract and vocal folds. » In : *The Journal of the Acoustical Society of America* 126.4, p. 2205–2205.

– (2011). « TubeTalker : An airway modulation model of human sound production ». In : *Proceedings of the First International Workshop on Performative Speech and Singing Synthesis*. P3S 2011, Vancouver, Canada, p. 1–8.

TOKUDA, K., T. KOBAYASHI et S. IMAI (1995). « Speech parameter generation from HMM using dynamic features ». In : *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. T. 1. IEEE, p. 660–663.

TOMASHENKO, N. et Y. KHOKHLOV (2014). « Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing ». In : *Interspeech*.

TOMASHENKO, N., Y. KHOKHLOV et Y. ESTÈVE (2016). « On the Use of Gaussian Mixture Model Framework to Improve Speaker Adaptation of Deep Neural Network Acoustic Models. » In : *Interspeech*, p. 3788–3792.

TOMASHENKO, N., K. VYTHELINGUM, A. ROUSSEAU et Y. ESTÈVE (2016). « LIUM ASR systems for the 2016 Multi-Genre Broadcast Arabic challenge ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 285–291.

VIRTANEN, T., R. SINGH et B. RAJ (2012). *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons.

WAIBEL, A., T. HANAZAWA, G. HINTON, K. SHIKANO et K. J. LANG (1989). « Phoneme recognition using time-delay neural networks ». In : *IEEE transactions on acoustics, speech, and signal processing* 37.3, p. 328–339.

WANG, Y., R. J. SKERRY-RYAN, D. STANTON, Y. WU, R. J. WEISS, N. JAITLY, Z. YANG, Y. XIAO, Z. CHEN, S. BENGIO, Q. LE, Y. AGIOMYRGIANNAKIS, R. CLARK et R. A. SAUROUS (2017). « Tacotron : Towards end-to-end speech synthesis ». In : *arXiv preprint arXiv :1703.10135*.

YANG, X.-K., D. QU, W.-L. ZHANG et W.-Q. ZHANG (2016). « The NDSC transcription system for the 2016 multi-genre broadcast challenge ». In : *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, p. 273–278.

YANG, Y. et J. EISENSTEIN (2013). « A log-linear model for unsupervised text normalization ». In : *Empirical Methods in Natural Language Processing Conference (EMNLP)*, p. 61–72.

YAO, K. et G. ZWEIG (2015). « Sequence-to-sequence neural net models for grapheme-to-phoneme conversion ». In : *arXiv preprint arXiv :1506.00196*.

YOUNG, S. J., J. J. ODELL et P. C. WOODLAND (1994). « Tree-based state tying for high accuracy acoustic modelling ». In : *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, p. 307–312.

ZEILER, M. D. (2012). « ADADELTA : an adaptive learning rate method ». In : *arXiv preprint arXiv :1212.5701*.

ZEN, H., T. NOSE, J. YAMAGISHI, S. SAKO, T. MASUKO, A. W. BLACK et K. TOKUDA (2007). « The HMM-based speech synthesis system (HTS) version 2.0 ». In : *SSW*. Citeseer, p. 294–299.

ZHANG, Xiaohui, Vimal MANOHAR, Daniel POVEY et Sanjeev KHUDANPUR (2017). « Acoustic Data-Driven Lexicon Learning Based on a Greedy Pronunciation Selection Framework ». In : *Proc. Interspeech 2017*, p. 2541–2545. DOI : 10.21437/Interspeech.2017-588. URL : http://dx.doi.org/10.21437/Interspeech.2017-588.