Reconnaissance automatique de la parole à large vocabulaire : des approches hybrides aux approches End-to-End

Abdelwahab Heba

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *22/03/2021* **par :**

Abdelwahab HEBA

Reconnaissance automatique de la parole à large vocabulaire : des approches hybrides aux approches End-to-End

---

JURY

Yannick Estève	Professeur d'Université	Président du Jury
Yifan Gong	Directeur de Recherche Microsoft	Membre du Jury
Ronan Collobert	Directeur de Recherche Facebook	Membre du Jury
Ariane Nabeth-Halber	Directrice de recherche Bertin	Membre du Jury

---

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse

Directeur(s) de Thèse :
Thomas Pellegrini, *Jean-Pierre Lorré* et *Régine André-Obrecht*

Rapporteur :
Yannick ESTÈVE, Yifan GONG

Table des Matières

Résumé

Ma thèse s'inscrit dans le cadre du projet de recherche OpenPaasNG [1] qui a été lancé en 2015 pour une durée de 4 ans avec l'objectif de développer une nouvelle génération de plateforme collaborative proposant un outil de visio-conférence utilisant des technologies d'intelligence artificielle pour la transcription de la parole, l'extraction des mots clés et le résumé automatique des réunions. Dans ce contexte, l'objectif de cette thèse est d'étudier, d'approfondir, de construire et d'enrichir la brique de reconnaissance automatique de la parole (RAP) pour aider à l'exploitation de l'information verbale durant les réunions en temps réel et en off-line à des fins d'archivage.

Cette thèse porte sur l'étude des méthodes de modélisation acoustique pour la RAP. Le bouleversement des architectures neuronales séquentielles ont permis de nos jours des avancées majeures dans l'amélioration de la modélisation et l'apprentissage du modèle acoustique. Nous explorons dans ce travail les deux grandes familles des systèmes RAP à savoir : les approches *traditionnelles hybrides* et *End-to-End*.

Une première partie de cette thèse concerne les approches traditionnelles et est dédiée à la mise en place d'un système RAP large vocabulaire en français pour la parole spontanée, déployé dans le contexte industriel. Un grand travail de collecte de données, de traitement et de normalisation a été effectué dans un premier temps pour atteindre l'objectif des 1000 heures de parole annotée. Une évaluation des composants acoustique, lexical et linguistique est proposée pour affiner au mieux le choix et l'orientation de la modélisation hybride DNN-HMM pour la langue française. Nous décrivons dans cette partie le développement de la plateforme industrielle d'adaptation des composants hybrides appelée "LinSTT Model Factory" permettant une adaptation des modèles aux conditions d'utilisations, à savoir : un contexte acoustique particulier, un vocabulaire spécifique à un domaine cible.

Dans une deuxième partie, nous abordons la problématique de prédiction de représentation textuelle directement à partir des observations acoustiques. Pour cela, nous effectuons une étude approfondie des approches RAP End-to-End : comment pouvons nous apprendre des alignements séquentiels entre l'audio et le texte ? Quel type d'architecture utiliser ? Et surtout, quel type d'unités en sortie choisir (caractère, pièce de mot, mot) ? Nous répondons à ces questions avec un ensemble d'expérience sur les corpus TIMIT et LibriSpeech. Ces travaux ont été, dans une grande partie, menés au cours d'un séjour scientifique au laboratoire du Mila au Canada dans le cadre du développement de l'outil open-source "SpeechBrain".

Dans une troisième partie, nous explorons des approches *multi-tâche* sur les systèmes RAP End-to-End afin d'exploiter plusieurs représentations textuelles, dans notre cas, des sorties caractères et des sorties consonnes/voyelles. Nous avons proposé une nouvelle technique de combinaisons des représentations textuelles pour l'amélioration des performances de reconnaissance.

Mots-Clefs— Reconnaissance Automatique de la Parole, Deep Learning, système RAP hybride, systèmes RAP End-to-End, Approche multi-tâche

1. WebSite OpenPaas : `https://open-paas.org/`, `https://research.linagora.com/display/openpaasng/OpenPaaS+NG+Overview`

Abstract

At Linagora, the OpenPaasNG project was launched in 2015 for a period of 4 years with the purpose of building a new generation of collaborative platform which provides a videoconferencing tool using artificial intelligence technologies such as speech transcription, keyword extraction and automatic meeting summary. In this context, my industrial thesis tackled Automatic Speech Recognition (ASR) for virtual meetings in real time and offline.

This thesis deals with the study of acoustic modeling methods for ASR. The upheaval of sequential neural architectures has nowadays allowed major advances in the improvement of modeling and learning of acoustic models. I explored in this work the two main families of RAP systems : *hybrid* and *End-to-End* systems.

A first part of this thesis concerns traditional approaches and is dedicated to the implementation of a large vocabulary RAP system in French for spontaneous speech, which has been deployed in several industrial use cases. A large work of data collection, processing and standardization was carried out in a first step to reach the goal of 1000 hours of annotated speech. An evaluation of the acoustic, lexical and linguistic components is proposed to refine the choice and orientation of the hybrid DNN-HMM modeling for the French language. I proposed for this part an industrial platform for the adaptation of hybrid components called "LinSTT Model Factory" allowing an adaptation of the models to the conditions of use, namely : a particular acoustic context, a vocabulary specific to a target domain.

In a second part, I approached the problem of automatically transcribing speech directly from acoustic observations. To do so, I conducted an in-depth study of the End-to-End RAP approaches : how can we learn sequential alignments between audio and text ? What type of architecture to use ? And especially, what type of output units to choose (character, word piece, word) ? I tried to answer to these questions with a set of experiments on the TIMIT and LibriSpeech datasets. A large part of this work has been conducted during a scientific stay at the Mila laboratory in Canada, where I actively contributed to the development of the open-source tool "SpeechBrain".

In a third part, I report multi-task learning experiments using end-to-end systems in order to exploit several output representations, in our case characters and consonant / vowel categories. I proposed a new technique of combining thses representations for improving recognition performance.

Mots-Clefs — Automatique Speech Recognition, Deep Learning, ASR Hybrid systems, End-to-End ASR systems, Multi-Task learning approach for ASR

Remerciements

On y est, quatre années de thèse remplies de savoir et de partage. Une grande chance d'avoir été encadré par Régine André-Obrecht, Thomas Pellegrini (au sein de l'IRIT), et de Jean-Pierre Lorré (au sein de Linagora). C'est avec une grande satisfaction personnelle que j'écris ces mots pour dire MERCI pour votre confiance et votre accompagnement tout le long de ma thèse.

Je remercie M. Yifan Gong et M. Yannick ESTEVE d'avoir accepté d'être raporteurs de ma thèse, mais également pour leurs corrections et remarques pertinentes vis-à-vis de mon manuscrit. Je remercie également les membres de mon jury : Ronan Collobert, Ariane Albert-Nabeth et Jean-Pierre Lorré.

Je tiens à remercier Mirco Ravanelli et Titouan Percollet de m'avoir donné l'opportunité de les rejoindre pour le développement du projet SpeechBrain au MILA. Une expérience au plan humain et professionnel très enrichissant avec des collaborateurs des 4 coins du monde, Le soleil ne se couche jamais avec SpeechBrain... Je tiens à adresser mes remerciements à l'ensemble des thésards et personnels avec qui j'ai partagé ce séjour scientifique (dans le désordre) : Peter Plantinga, Loren Lugosch, Jianyuan Zhong, Ju-Chieh Chou, Nauman Dawalatabad, Aku Rouhe. Un grand merci au Professeur Renato De MORI et au Professeur Yoshua Bengio pour leur accompagnement scientifique du projet.

J'ai eu la chance de rejoindre l'entreprise Linagora pour mes années de thèse industrielle qui m'a beaucoup tant apporté au niveau professionnelle que humain, j'ai tant appris de vous, les geek de l'open-source. De bons moments très mémorables partagés avec vous et une équipe à la hauteur des attentes technologiques. Un grand merci à mes collègues : Sonia, Ilyes, Sami, Nicolas, Damien, Sarra, Yohann, Flo, Bertrand, Topi, Sébastien, Omar, Jordy, Zied, Hamza, Kate, Julie, Romain et Albin.

SAMOVA, ce nom représente pour moi le sens des valeurs humaines et du partage scientifique. Une équipe pleine de valeur que je ne saurai oublier, un grand merci pour votre accueil et encouragement à aller de l'avant. Je tiens à remercier tous mes collègues thésards : Benjamin, Léo, Lucile, Vincent, Estèle, Nicolas, Sébastien, Imed, Jim, Thimoty, Thomas, Timothy, Sébastio, et Robin.

Je tiens à remercier mes parents qui m'ont toujours soutenu pour mes choix et encourager pour avancer toujours dans le monde du savoir et du partage. Merci à toi ma soeur Nassiba pour tes encouragements et à toi Ikram la femme de ma vie du soutien que tu m'as apporté.

Chapitre 1

Introduction générale

La communication humaine, parler naturellement, être compris...

... processus que l'humain a développé pendant son évolution et qui fait de la thématique du traitement automatique du langage et de la parole, un domaine des plus convoités pour sa compréhension.

La production de la parole permet d'encoder diverses informations :
— Le message transmis, on parle de *l'information verbale*.
— La manière avec laquelle le message est transmis, on parle alors de *l'information non-verbale*.
La question principale que nous nous posons lorsque nous cherchons une **représentation textuelle** de l'information verbale est : comment pouvons-nous construire un ensemble de traitement pour transformer les ondes acoustiques perçues (échantillons audio) en une représentation textuelle la plus fidèle possible satisfaisant les règles grammaticales de la langue cible ?

Nous détaillons dans cette thèse, le fruit de nos recherches axés sur la reconnaissance automatique de la parole (RAP) permettant de représenter textuellement l'information verbale présente dans la parole.

Je propose en annexe une étude de l'information non-verbale et plus précisément une contribution sur la détection automatique des mots emphasés (accentués)/importants dans la parole.

1.1 Contexte Scientifique

La Reconnaissance Automatique de la Parole (RAP) est devenu aujourd'hui un moyen largement utilisé pour répondre à des besoins d'interaction homme-machine. Le canal vocal représente en effet le canal favori de l'humain pour communiquer, et plus important encore, le canal vocal représente le canal contenant le plus grand volume d'information par unité de temps : on peut prononcer jusqu'à 200 mots en moyenne pour une durée de 1 minute de parole ; cependant, pour la même durée de temps, on peut écrire uniquement 40 mots en moyenne.

Le domaine de la Reconnaissance Automatique de la Parole (RAP) a bénéficié d'avancées scientifiques considérables durant ces trente dernières années. Ces avancées sont dues aux évolutions technologiques en lien avec le développement d'ordinateurs de plus en plus puissants, à l'émergence des méthodes et architectures neuronales du domaine Deep Learning (Goodfellow et al., 2016; Deng and Yu, 2014), à l'explosion des quantités de données vocales recueillies et aux explorations/expérimentations plus exhaustives des techniques de traitement des données séquentielles dans le domaine de la RAP.

Le problème du domaine de la RAP se résume à faire correspondre les représentations audio et textuelles, pour ce faire, deux phases principales sont nécessaires : 1) une phase d'alignement des représentations des deux séquences, 2) une phase de reconnaissance de la représentation textuelle à partir de l'observation audio.

Le principe de la RAP se basant sur la formulation statistique bayésienne introduite en 1976 a grandement été bouleversé par l'arrivée des modélisations neuronales. Deux grandes familles ont pu émerger :

— Les approches traditionnelles nécessitant 3 composants : acoustique, phonétique et linguistique pour la prédiction des mots à partir du signal audio.
— Les approches neuronales appelée "End-to-end" permettant de prédire directement les mots à partir du signal audio.

Les approches traditionnelles ont longuement été développées durant les 4 dernières décennies (Baum et al., 1970; Rabiner and Juang, 1986; Baum, 1972; Theodoridis and Koutroumbas, 2003; Woodland et al., 1994; Povey et al., 2011) ; cette approche consiste à modéliser le problème de la RAP en le découpant en trois composants : acoustique, phonétique et linguistique en utilisant la formulation statistique bayésienne. Le **composant acoustique** permet de résoudre le problème de prédiction de la représentation phonétique à partir du signal audio, le **composant phonétique** permet de lier la représentation phonétique aux mots phonétiquement possibles à l'aide d'un dictionnaire de prononciation phonétique, le **composant linguistique** quant à lui permet de sélectionner le mot le plus probable sachant le contexte d'apparition du mot dans la phrase (grammaire et conjugaison).

La gestion des alignements des données séquentielles avec cette approche a été effectuée par l'introduction des modélisations cachés de Markov ("Hidden Markov Model (HMM)"). La partie reconnaissance, quant à elle, a été traitée par l'introduction de densité de probabilité pour la prédiction de la représentation phonétique (phone, tri-phone..) à partir des données d'apprentissage : les modèles de mélange de lois gaussiennes ("Gaussian Mixture Model (GMM)") ont précédé les approches neuronales.

La combinaison des modèles neuronaux et des modèles HMM est appelée dans la littérature "modèle hybride" : on associe les approches neuronales (pour la reconnaissance) à la modélisation HMM (pour l'apprentissage des transitions/alignements).

Plusieurs contraintes ont été identifiées dans cette approche traditionnelle :

1. (Trad-P1) Le modèle de prononciation (dictionnaire) est figé, cela limite les systèmes RAP utilisant cette approche à reconnaître uniquement les mots existant dans ce dictionnaire.
2. (Trad-P2) L'optimisation des trois composants ne résout pas le problème de base de la RAP. Le composant acoustique traite la problématique d'alignement et de reconnaissance de données séquentielles et il permet la prédiction de représentation phonétique et non la prédiction des mots.

L'approche traditionnelle permet à ce jour d'apprendre à partir des données la dépendance acoustique et la dépendance linguistique est traitée avec un modèle probabiliste estimé à partir des fréquences de mots dans un corpus de données textuelles.

Les approches "End-to-End" ont bouleversé le domaine en permettant d'apprendre à prédire des sorties textuelles directement à partir d'entrées audio sans avoir à construire des composants intermédiaires. Les problèmes (Trad-P1 et P2) ont été résolus par le développement de ces architectures neuronales, notamment avec :

— L'introduction de la fonction de perte CTC (Graves et al., 2006; Graves and Jaitly, 2014; Amodei et al., 2016; Seide et al., 2011) permet l'apprentissage de la représentation textuelle directement à partir de l'observation acoustique avec l'introduction d'une prédiction fictive permettant de gérer l'alignement des deux séquences d'entrée et de sortie. On parle d'alignement monotone.
— L'introduction de l'architecture RNN-T (Graves, 2012; Li et al., 2019b; Rao et al., 2017) permet de joindre l'information issue de deux composants neuronaux : acoustique et linguistique.
— L'introduction de l'architecture seq2seq avec mécanisme d'attention (Bahdanau et al. (2016); Chorowski et al. (2015)) permet d'encoder l'information acoustique avec un composant neuronal appelée "Encodeur" et produit la séquence de sorties avec un deuxième composant neuronal appelée "Décodeur". Cette architecture permet de réaliser la tâche de RAP en deux phases, une première d'encodage de l'information acoustique puis une deuxième phase de reconnaissance de la séquence de sorties avec un mécanisme d'attention.

Plusieurs problèmes ont été identifiés avec l'avènement de cette approche, notamment :

1. (E2E-P1) Le choix de l'unité de prédiction de la modélisation E2E. Le choix de ces unités ainsi que la façon avec laquelle le système RAP combine ces unités pour représenter les mots est un axe important à explorer.
2. (E2E-P2) L'important volume de données pour l'apprentissage de système E2E RAP.

Plusieurs solutions ont été proposées pour répondre au problème (E2E-P1) par l'introduction de prédiction au niveau : caractère, pièce de mot, mot. La prédiction au niveau mot produit un grand nombre de problèmes liés à l'apprentissage et au vocabulaire restreint (problème équivalent au Trad-P1). Les prédictions au niveau graphème ou pièce de mot peuvent être sous optimales.

Le problème (E2E-P2) représente un grand problème au niveau coût. Baidu dans le projet DeepSpeech (Hannun et al. (2014)) a construit un corpus de 5000 heures enregistrées par 9600 locuteurs ; le coût d'enregistrement a été évalué à 2 Millions de dollars et a permis en 2014 d'enregistrer le meilleur taux de performance des moteurs RAP industriels au monde. Microsoft utilise pour ses recherches sur les systèmes E2E un corpus d'environ 30000 heures issues de Cortana, Xbox et de données conversationnelles pour la langue anglaise.

Face à ce problème d'insuffisance de données et d'adaptation aux conditions réelles d'utilisation, goulet d'étranglement pour la construction de modèle E2E, nous trouvons dans la littérature plusieurs tentatives de résolution les problèmes :

1. Augmenter le corpus d'apprentissage
   — Utiliser des techniques de perturbation de données pour adapter aux conditions réelles Ko et al. (2015); Jaitly and Hinton (2013).
   — Enregistrer/synthétiser un ensemble de données vocales similaire au cas d'utilisation, pour adapter le modèle à toutes les conditions observées, accents, dialectes, environnements sonores Kim et al. (2017a); Ko et al. (2017).
2. Changer de technique et d'architecture par rapport aux volumes de données disponibles.
   — Utiliser des techniques plus adaptées aux petits/moyens corpus. (par exemple, la technique d'alignement/apprentissage proposée par Kaldi Povey et al. (2011))
   — Utiliser des techniques plus adaptées à la parole avec l'exemple des *sinc-net* par rapport aux CNN (Ravanelli and Bengio, 2018))
   — Utiliser des techniques de transfert learning et/ou d'apprentissage multi-tâche. (Krishna et al., 2018; Belinkov and Glass, 2017; Rao and Sak, 2017; Kim et al., 2017b)
   — Utiliser des techniques permettant d'inclure des données non supervisées. (Meng et al., 2019)

Un grand nombre de ces approches et techniques seront détaillées et analysées dans le chapitre 2, voire approfondies au cours des chapitres suivants.

1.2 Contexte Industriel

La maturité des solutions RAP durant cette décenie a permis leur avènement dans l'industrie et dans des applications grand public par le biais des assistants vocaux (Google home, Amazon Alexa, Siri de Apple). La brique RAP représente le pilier principal pour la compréhension de la communication humaine.

Ma thèse s'inscrit dans le cadre du projet de recherche **OpenPaas :NG** porté par les entreprises *Linagora*, *Xwiki*, *Nexidis*, ainsi que le laboratoire *LIX* de l'école polytechnique, le laboratoire *Loria* et le laboratoire *IRIT* dans le cadre de sa collaboration avec *Linagora*. Ce projet de recherche d'une durée de 4 ans vise à développer une offre **open source** française de plateforme de collaboration de nouvelle génération de type "plateforme as a service (PaaS)" (équivalente à Microsoft 365) pour les entreprises et les administrations, et proposer une alternative et indépendance technologique vis-à-vis des grands acteurs classiques du numérique. Son objectif est de fournir un bureau virtuel, qui offre l'ensemble des fonctionnalités nécessaires pour travailler, de manière collaborative, dans une organisation et/ou une entreprise : mail, gestion du planning, édition partagée de documents, vidéo-conférences.

La brique des systèmes intelligents a été au coeur de ce projet afin d'assister et d'automatiser certaines tâches, tels que :

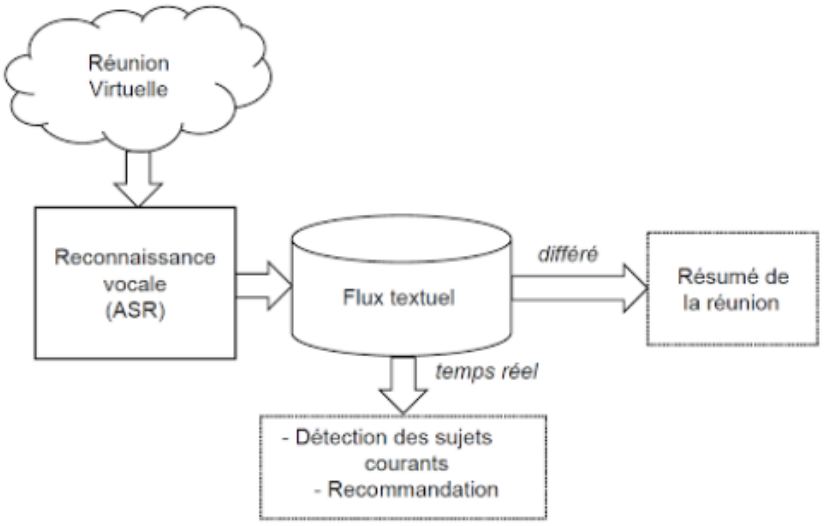


FIGURE 1.1 – Réunion virtuelle contextuelle

— La réponse intelligente (Smart reply) : la proposition de réponse automatique aux mails.
— La reconnaissance automatique de la parole (RAP) : transcription des réunions.
— La recommandation de documents : proposition de documents pertinents à partir de la transcription issue de la réunion.
— Le résumé automatique : génération de résumé automatique (abstractif/extractif) de la transcription issue de la réunion.

Ma thèse, sous contrat CIFRE, rentre dans le cadre d'une collaboration entre l'équipe SAMoVA du laboratoire IRIT et l'entité R&D de l'entreprise Linagora portant sur la partie RAP. Le besoin industriel du projet est de disposer d'un système RAP large vocabulaire :

— pour la parole spontanée à l'état de l'art pour la langue française.
— en mode 'temps-réel' et 'offline' scalable.

La satisfaction à ces besoins, identifiés dans le cadre du sous projet "réunion virtuelle contextuelle" (voir figure 1.1) permettra d'exploiter le flux audio issu de la vidéo-conférence pour l'analyse de son contenu, à savoir :

— d'une part l'analyse en temps-réel de la parole permet l'extraction de mot clé, et par voie de conséquence la recommandation immédiate de documents.
— d'autre part l'analyse de la parole à posteriori, avec un système plus complet intégrant le découpage en tour de parole, l'identification des locuteurs et la transcription de toute la réunion, peut être utilisée à des fins d'archivage et d'édition de résumé automatique.

Répondre au double challenge entre recherche et industrialisation de mes recherches et solutions technologiques a été un point majeur sur les choix des architectures, de mes orientations de recherche dans le domaine de la reconnaissance de la parole.

1.3 Contributions de la thèse de doctorat

Il est clair que l'avènement des techniques/modélisations neuronales séquentielles et profondes permet un apprentissage plus complexe des représentations. Le domaine de la RAP a grandement bénéficié des progrés du Deep Learning.

Ma thèse s'inscrit dans ce contexte ; j'ai exploré, sous l'angle "neuronal", à la fois les approches traditionnelles et les approches End-to-End et j'ai pu apporter des contributions dans chacun de ces axes de recherche.

1.3.1 Contributions sur les approches traditionnelles

Mes travaux empruntant l'approche traditionnelle avait pour principal objectif la réalisation d'un système RAP hybride pour la langue française ; ce choix a été fait dans le cadre du projet de recherche OpenPaas :NG. Le besoin lié à la mise en place de ce système consistait à **proposer un service RAP avec possibilité d'adaptation de la modélisation acoustique, du vocabulaire et du contexte linguistique** suivant le domaine des clients (banque et assurance, aéronautique, médecine...) et de traiter le problème (Trad-P1) décrit à la section 1.1.

Les anciennes modélisations GMM-HMM étaient très loin de satisfaire le besoin de robustesse de la modélisation acoustique. Nous avons donc focalisé nos efforts sur les facteurs de succès des modélisations DNN-HMM hybrides rapportés dans la littérature, et nous avons contribué à la mise à disposition d'une modélisation DNN-HMM hybride industrialisée pour la langue française et performante. Pour ce faire, nos actions se déclinent selon trois axes :

1. Sur le plan collecte de données : 1) construire un ensemble de corpus de 1000 heures de parole pour la langue française, 2) identifier et regrouper tous les supports textuels en langue française disponibles en open-source, 3) sélectionner et normaliser la masse de données parole et texte recueillies.
2. Sur le plan acoustique : 1) identifier les composants neuronaux les plus discriminants pour la modélisation acoustique. 2) explorer les différentes techniques de perturbation et augmentation de données.
3. Sur le plan lexical et linguistique : 1) produire un modèle de langage pour la parole spontanée en français. 2) sélectionner un large dictionnaire pour la langue française.
4. Sur le plan industriel : mettre en place une plateforme industrielle d'apprentissage et d'adaptation des modèles hybrides à la demande permettant de résoudre le problème Trad-P1. J'ai appelé cette solution “LinSTT Model Factory”.

En parallèle à mes recherches sur les systèmes RAP pour la langue française, je me suis intéressé, dans le cadre du projet de recherche OpenPaas :NG, à la détection d'emphase (accentuation) au niveau mot pour la langue française. J'ai proposé une métrique de l'analyse de l'information non verbale afin de l'exploiter pour améliorer la détection des mots clés au niveau NLP réalisé en collaboration avec l'école polytechnique.

1.3.2 Contributions sur les approches End-to-End

Dans le cadre des approches End-to-End, nous avons tenté de répondre à deux problèmes identifiés, à savoir :

— (Q1) Avec une ressource limitée de données, (dans le cas de Linagora, des données clients représentent un petit volume < 100 heures), comment pouvons nous exploiter des approches E2E “multi-taches” avec différentes représentations textuelles et permettre un meilleur apprentissage des systèmes RAP, comparé à l'approche traditionnelle hybride ?

— (Q2) Avec un volume de données suffisamment grand, quel type d'architecture E2E devons nous choisir ? quelle modélisation neuronale ? avec quelle représentation textuelle ?

Nous avons souhaité approfondir nos recherches sur les approches E2E à alignement monotone. L'architecture RNN-T comparée aux architectures seq2seq avec attention répond aux besoins industriels de décodage temps-réel. Cependant, l'architecture RNN-T est très complexe à mettre en place : très peu d'implémentation Open-Source de la fonction de perte existe ; l'implémentation parallélisée sur CPU du calcul de la fonction de coût est très coûteuse en temps. Les RNN-T sont aussi très gourmands en mémoire GPU ce qui induit un cycle d'apprentissage très long. D'où les questions suivantes :

— (Q3) Comment pouvons nous paralléliser le calcul de la fonction de perte sur GPU ?

— (Q4) Comment pouvons nous initialiser les modèles RNN-T pour réduire leurs temps d'apprentissage ?

Notre réponse à la question (Q1) s'inspire des travaux de Jimenez et al. (2018) sur la détection d'environnements sonores et la hiérarchisation des représentations. Nous avons étudié les approches multi-tâche (MTL) pour la RAP à base de modélisation CTC caractères et nous avons exploré la combinaison des sorties MTL en proposant l'approche Char+CV-CTC qui combine les représentations de la tâche primaire caractère et auxiliaire qu'est la reconnaissance des consonnes et des voyelles, pour la prédiction de la séquence de sortie caractère.

Une grande partie des réponses aux questions (Q2, Q3, Q4) sur les architectures RAP End-to-End a été traitée lors de mon séjour scientifique au Mila avec le développement d'un outil Open-source appelé "SpeechBrain" [1] pour le partage des technologies E2E vocales. J'ai contribué au projet :

1. En explorant les différentes architectures E2E et identifiant des points d'amélioration de la modélisation neuronale à utiliser sur les corpus de référence TIMIT et LibriSpeech.
2. En comparant les différentes méthodes de génération de sorties qui s'appuient sur les connaissances linguistiques (prédiction caractère, bout de mots). J'ai, en autres, intégré le module de génération des bouts de mots sur l'outil SpeechBrain.
3. En implémentant de l'architecture RNN-T avec une fonction de perte parallélisée sous GPU.
4. En étudiant et évaluant les différentes initialisations du RNN-T avec des modèles pré-entraînés versus une initialisation à base d'approche multi-tâche.

1.4 Organisation du manuscrit

Ce document est constitué de six chapitres et est organisé comme suit :

Le **chapitre 1** est dédié à l'état de l'art et consiste à donner une vue détaillée des évolutions des méthodes/techniques/modélisations et architectures appliquées dans le domaine de la RAP, ainsi que leurs principes et détails techniques pour pouvoir ainsi situer les travaux réalisés au cours de ma thèse.

Le **chapitre 2** détaille les corpus traités, oraux et textuels, en français et en anglais, pour l'apprentissage de systèmes RAP, ainsi qu'une description des étapes de pré traitement nécessaires pour les rendre utilisables.

Le **chapitre 3** reprend l'étude d'un système RAP, large vocabulaire, pour la langue française, fondé sur une modélisation hybride DNN-HMM afin d'approfondir les choix faits pour les modélisations acoustique, lexical et linguistique. Je décris la plateforme industrielle "Model Factory" pour l'adaptation des modèles aux cas d'usages industriel (adaptation acoustique, mise à jour du vocabulaire), plateforme que j'ai développée dans le cadre de ma thèse industrielle.

Le **chapitre 4** traite le passage d'une modélisation hybride phonétique à une modélisation E2E CTC à base de graphèmes. Ce passage apporte une dimension de complexité supplémentaire liée à l'apprentissage de représentations linguistiques directement à partir de la modélisation neuronale. Nous proposons d'explorer des approches "multi-tâche" prenant en compte plusieurs informations linguistiques afin d'améliorer la représentation linguistique apprise implicitement par le modèle CTC.

Le **chapitre 5** est consacré à mes contributions sur les approches E2E réalisées lors de mon stage au Mila dans le cadre du développement du futur outil SpeechBrain. J'ai pu proposer une comparaison des architectures E2E avec la modélisation CRDNN sur le corpus TIMIT et LibriSpeech en explorant differentes granularités des unités de prédiction. J'ai approfondi mes travaux sur l'approche RNN-T en proposant une implémentation parallélisée sous GPU du calcul de la fonction de perte, et en évaluant différentes méthodes d'initialisation des systèmes RNN-T.

1. WebSite SpeechBrain : `https://speechbrain.github.io/`

Le **chapitre 6** conclut le manuscrit de thèse avec une synthèse de mes contributions ainsi qu'une discussion des voies possibles pour la continuation de mon travail.

Chapitre 2

Des machines parlantes aux réseaux de neurones profonds

Introduction

Le domaine de la Reconnaissance Automatique de la Parole (RAP) a bénéficié d'avancées scientifiques considérables durant ces trente dernières années. L'évolution des technologies en relation avec celle des ordinateurs de plus en plus performants et rapides, a permis une exploration/expérimentation plus exhaustive des techniques dans ce domaine. Le principe de base de la RAP a particulièrement évolué depuis la formulation statistique bayésienne introduite en 1976, qui nécessite des composants acoustique, phonétique et linguistique. La modélisation neuronale "end-to-end" a bouleversé ce domaine en permettant d'apprendre à prédire des sorties textuelles directement à partir d'entrées audio sans avoir à construire des composants intermédiaires.

L'objectif de ce chapitre consiste à donner une vue détaillée de l'évolution des méthodes appliquées en RAP, ainsi que leurs principes et détails techniques pour pouvoir ainsi situer les travaux réalisés au cours de ma thèse. Cet aperçu fournira également aux lecteurs, tout le contexte scientifique et technique pour aborder des études sur les systèmes RAP.

2.1 Historique

Toutes les recherches existent dans un certain contexte historique, répondant à des questions pressantes de l'époque tout en utilisant les technologies existantes.

Durant les deux derniers siècles, l'évolution des recherches dans le domaine du traitement automatique de la parole (TAP) dont celui de la reconnaissance de la parole est liée à des problématiques graduellement plus complexes et des avancées majeures associées (voir figure 2.1).

Depuis les années 60, la RAP a évolué en prenant en compte l'intérêt stratégique, militaire, et commercial qu'elle a progressivement suscités. Les premières recherches portant sur le domaine de la RAP étaient orientés sur l'analyse de la télécommunication et de la défense militaire.

Plus récemment, une mouvance d'ouverture des technologies de la RAP ont permis de propulser les recherches dans ce domaine, dans le but commun d'offrir des systèmes RAP, pour toutes les langues au même niveau des performances humaines. Cette mouvance est reflétée par :

— Des initiatives de partage de corpus libre (VoxForge, Mozilla).
— Le partage des techniques d'apprentissage RAP (DeepSpeech Mozilla, Wav2Letter Facebook, ESPNET, Kaldi, SpeechBrain Project, etc..) en *open-source* par les géants de la technologie (GAFAM, BATX, etc..), les institutions de recherche (on peut citer CMU et le Mila), et les acteurs du logiciel

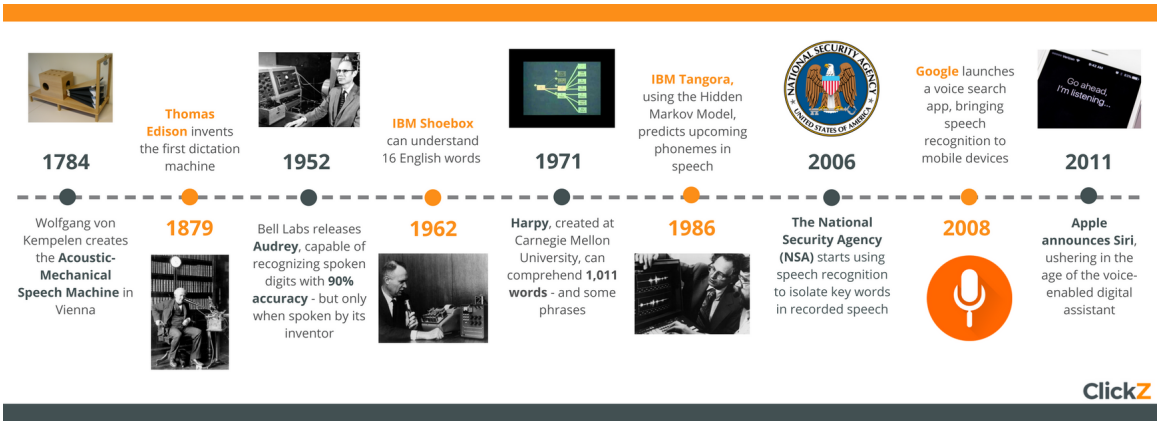


FIGURE 2.1 – L'évolution du domaine de la Reconnaissance Automatique de la Parole.

libre (Mozilla, Linagora, Snips) en traitement automatique du langage et de la parole.

Le TAP est donc loin d'être une nouvelle préoccupation, même si le rythme de l'évolution technologique n'a pas toujours été à la hauteur de l'intérêt théorique comme expérimental suscité par le sujet.

Dans ce qui suit, nous proposons une vue d'ensemble sur l'évolution du domaine RAP. On propose de commencer à décrire les évolutions à partir du 18e siècle, ce choix de la date et des découvertes historiques associées m'ont marqué lors de ma participation à *InterSpeech 2019 - Graz*.

2.1.1 18ème et 19ème siècle : Les machines parlantes

Les premiers progrès de la RAP remontent au 18ème siècle, notamment avec l'invention de Wolfgang Von Kempelen (1791) qui s'est concentré principalement sur la création de sons de voyelles. Ce système avait comme but d'imiter la production de la parole et ainsi permettre de définir une base pour le développement d'un système qui pourrait également apprendre à interpréter les phonèmes (les unités sonores élémentaires pour la production d'un mot). Thomas Edison Edison (1888, 1916) s'est intéressé à un dispositif capable d'enregistrer la parole humaine en 1889. Ce dispositif appelé "phonographe" (en anglais "Talking machine" ou bien "Dictation machine") a longtemps servi à enregistrer des notes au quotidien par les administrations et le corps de santé.

Les limitations technologiques de l'époque ont gêné la progression de ce domaine qui ne disposaient que de moyens élémentaires pour inventer des machines parlantes. Néanmoins, ils fournissent un contexte important pour des innovations plus récentes.

2.1.2 Reconnaissance de mots isolés : "Pattern Matching"

Les premiers travaux sur la RAP remontent aux années 50 et 60, et ils ont été effectués par l'équipe R&D de Bell Labs Davis et al. (1952). La machine appelée Audrey, avait pour tâche la détection de mots isolés, et pouvait reconnaître les chiffres de 0 à 9, avec un taux de précision de 90%. Il est intéressant de noter que ce niveau de précision n'était atteint que lorsque son inventeur parlait. On parle alors de système dépendant du locuteur ; les performances oscillaient aux alentours de 70 % lorsque d'autres personnes parlaient à Audrey.

Ces travaux, ainsi que la plupart des autres recherches durant cette période (Dudley and Balashek (1958); Forgie and Forgie (1959)) se basaient sur une modélisation qui traitait tout le segment du mot comme une unité de comparaison. La technique de comparaison "Pattern matching" en anglais, proposée à cette époque calculait des distances et utilisait le schéma de l'évolution des premières et deuxièmes formants du mot isolé. Pour ce faire, un exemple de chaque mot était prononcé et enregistré au moins une fois (et ce pour

chaque locuteur). Lorsqu'un utilisateur prononçait un mot nouveau inconnu dans le microphone, l'ordinateur comparait cet audio avec toutes les références. La correspondance la plus proche fournissait la reconnaissance du mot. Les résultats étaient très satisfaisants avec l'ensemble de 10 chiffres.

L'une des applications les plus marquantes durant cette période a été proposée par IBM. Via le projet *Shoebox* (1961), la reconnaissance de 16 mots, dont les chiffres de 0 à 9, ainsi que les opérateurs arithmétiques "plus", "moins", "totale" etc, était rendue possible pour faire le calcul d'opération mathématiques.

L'analyse du signal de parole par prédiction linéaire et la comparaison de formes acoustiques par programmation dynamique ont permis le développement de systèmes de reconnaissance performants pour des vocabulaires restreints au début des années 70 (Itakura 1975, Sakoe & Chiba 1971).

Simultanément, des systèmes fondés sur les phonèmes (la plus petite unité de son de parole) considérés comme unité d'analyse ont été élaborés (Potter et al. (1966); Forgie and Forgie (1959)). Leurs résultats étaient moins bons que la RAP fondée sur les modèles de mots directement.

La RAP a connu un excellent départ avec des taux de reconnaissance de près de 90 %, mais cette première approche de "Pattern matching" n'a pu être étendue pour plusieurs raisons :

— Cette approche s'appuyait sur les propriétés acoustiques des voyelles et leur évolution pour chaque mot, l'extraction de fréquences de formants comme caractéristiques ne suffit pas à différencier les mots (par exemple, "" et "") ;
— En plus, cela nécessite l'enregistrement de la représentation acoustique pour chaque mot. Si ce paradigme était étendu à un vocabulaire plus large, il se heurterait à des problèmes majeurs de complexité de temps et d'espace. La complexité de cet algorithme est proportionnelle au nombre de mots dans le lexique.

Il n'y a aucun doute sur la limitation de cette approche, et cela laisse entrevoir certains des défis persistants de la reconnaissance automatique de la parole : chaque individu a une voix différente et la langue parlée peut être très incohérente. Contrairement au texte, qui est beaucoup plus standardisé, la parole varie beaucoup en fonction des dialectes régionaux, de la vitesse d'élocution, de l'accent, voire de la classe sociale et du locuteurs. Par conséquent, la mise à l'échelle de tout système de reconnaissance automatique de la parole a toujours été un obstacle important à cette époque.

2.1.3 L'évolution de la RAP : Modélisation séquentiel par chaîne de Markov Cachées (HMM)

La phase d'expansion du domaine de la RAP a rapidement pris de l'essor à partir des années 70. Les agences gouvernementales telles que DARPA, ont lancé des programmes de recherches sur la compréhension de la parole (Spoken Language Understanding (SLU) en anglais). Des systèmes tels que "HearSay System I & II" (Maguire (1960); Erman et al. (1981)), "Dragon System" (Baker (1975)) et "Harpy System" (Lowerre (1976)) ont vu le jour.

Le phonème est devenu entre temps l'unité de base de modélisation (Liberman et al. (1957)). Le challenge des années 70 était de modéliser l'évolution de la séquence audio, ainsi que la chaîne à reconnaitre. Les domaines de recherches tels que : "*Network Data Representation*", "*Word Juncture Rule Generation*", "*Templete Generation*","*Lexical Generation*" ont quant à eux, beaucoup progressé durant cette décennie.

L'un des systèmes qui a marqué cette décennie, le système "Harpy", a été proposé par l'université de Carnegie-Mellon (CMU). Afin de traiter des phrases au lieu de mots isolés, en utilisant un vocabulaire plus important et des unités phonétiques en nombre limité, il a fallu procéder à une révision majeure du système de reconnaissance "classique" de mots isolés. James Baker (1975) a introduit la modélisation séquentielle par modèles de Markov cachés (Hidden Markov Models HMM) pour la RAP. Cette réussite a permis de projeter le domaine de la RAP dans une nouvelle ère. Le système RAP "Harpy" proposait :

— Une reconnaissance automatique des commandes pour les pilotes d'avion de chasse.
— De la parole continue sans segmentation de la phrase en mots isolés.
— Un vocabulaire de 1000 mots.
— Un taux d'erreur mot avoisinant les 10%.
— L'utilisation de 98 unités phonétiques (bi-phones) pour la phonétisation du vocabulaire.

— La reconnaissance de phrases formées avec au maximum 8 mots successifs, et construites en respectant une syntaxe contrainte "*Backus-Naur form (BNF) grammar*".
— Un réseau formé de 15000 états obtenu après compilation à partir des prononciations des mots et des phrases autorisées.
— Un décodage basé sur :
  — Le calcul des coefficients LPC sur chaque trame audio.
  — L'algorithme de "*Beam search*" pour la sélection de la meilleure chaîne possible.

Ce système dépendant du locuteur, devait enregistrer chaque locuteur pendant environ 30 minutes, puis forcer l'alignement de l'audio avec le graphe des phrases et permettre ainsi, à la phase de décodage, de comparer l'observation acoustique, avec les représentations d'observations phonétiques enregistrées. Ce système se basait toujours sur les algorithmes de type "*Templete matching*".

Le système "Harpy" décodait les phrases en environ 80 fois le temps réel (TR). Il ne pouvait pas être utilisé en pratique, et accélérer le process n'était pas une tâche simple. Outre le problème de la vitesse, la flexibilité grammaticale était une préoccupation majeure pour la réalisation des systèmes capables de reconnaître la parole spontanée. Les types de phrases reconnus par Harpy étaient déterminés par une grammaire de forme Backus-Naur (BNF) et il ne pouvait être utilisé que pour les commandes définies à l'avance. Durant cette période, les chercheurs ont réalisé qu'une telle grammaire n'était pas une option viable pour les systèmes RAP large vocabulaire.

Un changement majeur allait avoir lieu dans le monde de la RAP, s'éloignant de l'appariement des représentations acoustiques et des grammaires strictes pour se tourner vers un modèle statistique incluant non seulement les modèles acoustiques statistiques mais également les grammaires statistiques. De ce fait, le système assignerait une probabilité à la phrase, au mot, mais aussi aux segments sonores.

2.1.4 L'âge d'or de la RAP : la modélisation statistique avancée

Il a fallu attendre les années 90 pour voir émerger des systèmes RAP incluant des modélisations statistiques plus complexes telles que les mélanges de lois gaussiennes (Juang (1985)) (appelée en anglais : "Gaussian Mexture Model (GMM)") et des grammaires statistiques (modèle de langage) (Katz (1987)) qui ont été introduits durant les années 80. L'outil HTK Woodland et al. (1994) était l'un des premiers outils qui a concrétisé l'utilisation des architectures statistiques à base de HMM avec une estimation de la probabilité acoustique fondée sur des distributions gaussiennes, puis sur des mélanges de gaussiennes (GMM) incluant un modèle de langage.

Les solutions apportées pour l'approche statistique consistait à utiliser des algorithmes pour automatiser l'alignement des segments audio avec les états HMM et ainsi apprendre une modélisation acoustique d'une manière statistique (estimation de la densité) à partir d'un ensemble de données (segment audio, transcription et dictionnaire phonétique). L'algorithme le plus communément utilisé de Baum-Welch (appartenant à la catégorie des algorithmes "Expectation Maximization (EM)") proposé durant les années 70 Baum et al. (1970), consiste à re-estimer d'une manière itérative les paramètres du modèle acoustique (GMM-HMM) et ainsi améliorer l'alignement peu à peu entre la séquence phonétique et la séquence d'observations audio. Des techniques d'alignement Flat-Start on été proposées durant cette époque pour initialiser les systèmes RAP GMM-HMM à partir des données d'une manière très simplifiée. Cela consiste à générer la séquence phonétique des annotations à partir du dictionnaire phonétique et d'affecter la séquence d'observations audio à la séquence de phonèmes de gauche à droite, à raison d'un segment acoustique pour chaque phonème, segment de longueur uniforme (Baum et al. (1970); Rabiner and Juang (1986); Baum (1972); Theodoridis and Koutroumbas (2003)).

Pour rendre compte des probabilités des émissions, est apparu l'usage d'un réseau neuronal artificiel (ANN) et en particulier du perceptron multicouche (MLP). Un modèle hybride ANN-HMM a été utilisé pour la première fois pour la reconnaissance automatique de la parole en 1990 (Morgan and Bourlard, 1990). Ce type de méthodes n'a connu un véritable essor que durant ces dernières années ; les modèles acoustiques basés sur DNN pour créer des modèles hybrides DNN-HMM a entraîné des améliorations massives. Ces techniques seront examinées en détail dans la section 2.2.1.

2.1.5 Le bouleversement des techniques d'alignement neuronal

Le domaine de la RAP a radicalement changé à partir de 2012, les premières publications montrant des résultats prometteurs de la reconnaissance vocale avec l'utilisation de méthodes dites End-to-End, qui permettent de prédire des sorties textuelles directement à partir de l'audio (Graves et al. (2006); Graves (2012); Graves and Jaitly (2014); Hannun et al. (2014); Amodei et al. (2016); Collobert et al. (2016); Li et al. (2019a); Chiu et al. (2018)). Les anciennes approches consistaient à définir des modèles distincts pour les sons élémentaires (partie acoustique), les séquences de mots (partie linguistique) et les prononciations (lien entre la partie acoustique et linguistique). La maîtrise de chacun de ces modèles distincts prend un temps considérable pour un ingénieur, et constitue un fardeau pour la mise en oeuvre des systèmes RAP en production.

En outre, l'approche hybride traditionnelle est toujours compliquée à mettre en oeuvre car chaque modèle nécessite sa propre fonction d'optimisation lors de l'estimation des paramètres et, à ce titre, l'estimation conjointe de tous ces modèles n'est pas toujours optimale. Les approches hybrides traditionnelles ne gèrent pas les mots hors vocabulaire.

La RAP de type End-to-End, en revanche met en oeuvre une seule fonction d'optimisation pour l'estimation des paramètres. Cette fonction permet de modéliser directement le problème : trouver la meilleure transcription pour un segment audio donné. Cependant, les modèles de type End-to-End n'ont pas accès à l'information d'alignement généralement utilisée dans l'apprentissage des modèles traditionnels. Ainsi, le modèle de RAP End-to-End doit trouver un alignement du texte et de l'audio pendant l'apprentissage, ce qui n'est pas une tâche simple.

Les premières approches de la reconnaissance vocale End-to-End ont utilisé une fonction d'apprentissage appelée Classification temporelle (en anglais "Connexionist temporal classification" (CTC)) introduit par Graves et al. (2006). Le modèle CTC se base sur une fonction de coût et un encodeur neuronal récurrent pour modéliser la dépendance des séquences d'entrées et de sorties. Des modélisations plus évoluées à base de CNN et RNN ont vu le jour telles que l'architecture DeepSpeech 1 et 2 de Hannun et al. (2014); Amodei et al. (2016) ainsi que l'architecture Wav2Letter construite avec des couches purement convolutionnelles proposée par Collobert et al. (2016). En parallèle au CTC, un nouveau type d'approche appelée en anglais : "Transducer models" prenant en compte la dépendance linguistique (en anglais : "output-output dependency") a été introduit par Graves (2012). Cette approche a suscité beaucoup d'intérêt dans la communauté et a fait l'objet de plusieurs améliorations notamment pour le décodage en temps réel Battenberg et al. (2017); Rao et al. (2017).

À partir de 2014 (Chorowski et al. (2014); Bahdanau et al. (2016)), une extension notable a été proposée avec l'introduction des modèles séquence-à-séquence et mécanisme d'attention. L'un des modèles les plus marquants durant cette période, est le modèle dit "Listen, Attend and Spell" (LAS) (Chan et al. (2016)) proposé par Google.

Au cours des cinq dernières années, les modèles de type End-to-End sont devenus de plus en plus performants ; cependant ils sont généralement appris sur d'énormes quantités de données qui ne sont disponibles, pour la plupart des cas, que pour les géants de la technologie (GAFAM, BATX). Néanmoins, il existe des recherches actuelles sur la façon de faire fonctionner ces modèles de type End-to-End pour des ensembles de données plus petits. Le chapitre 5 de cette thèse se prête à cette problématique.

2.2 Approche statistique : principe et modélisation traditionnelle

L'approche statistique en RAP consiste à trouver la séquence de mots la plus probable sachant le signal de la parole (voir Figure 2.2), opération généralement appelée "Décodage". Ayant une suite P de vecteurs acoustiques $X = x_1, x_2, ..., x_p$, la phase de décodage permet de trouver une séquence de L mots telle que $\widehat{W} = w_1, ..., w_L$ soit la séquence la plus probable sachant l'observation X. On peut donc décrire cette assertion avec l'équation suivante :

$$\widehat{W} = \underset{w}{\operatorname{argmax}}\, P(W|X) \tag{2.1}$$

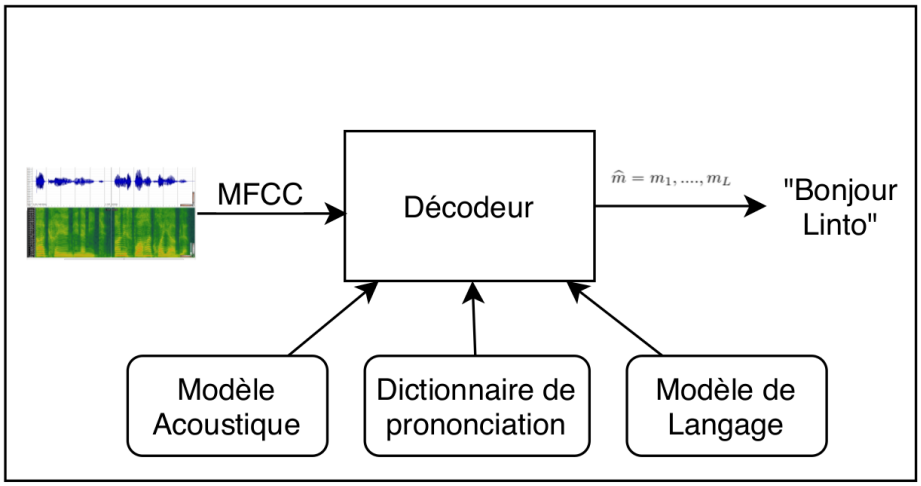


FIGURE 2.2 – Les composantes principales du processus de la reconnaissance automatique de la parole

Pour pallier à la difficulté de modéliser directement $P(W|X)$, l'approche traditionnelle adoptée et proposée par la communauté à partir des années 70,80 (Baum et al. (1970); Liporace (1982); Rabiner and Juang (1986)) consiste à appliquer la règle de Bayes à l'équation 2.1. Il s'en suit la décomposition de la probabilité $P(W|X)$ en trois composantes :

$$\widehat{W} = \underset{w}{\text{argmax}} \frac{P(W)P(X|W)}{P(X)} \tag{2.2}$$

Le dénominateur $P(X)$ est constant pour tous les séquences de mots possibles ; l'équation s'écrit sous la forme suivante :

$$\widehat{W} = \underset{w}{\text{argmax}}\, P(W)P(X|W) \tag{2.3}$$

La probabilité $P(W)$ est définie à partir d'un modèle de langage (LM). La probabilité conditionnelle $P(X|W)$ est quant à elle, calculée à partir des observations acoustiques et d'un modèle acoustique (AM).

Le lien entre le modèle acoustique (prédiction au niveau phonème) et le modèle de langage (prédiction au niveau mots) est réalisé par le biais d'un dictionnaire phonétique (appelé en anglais "Lexicon"). Ce dictionnaire permet d'établir une correspondance entre les mots et leurs prononciations. La figure 2.2 illustre l'architecture basique de l'approche statistique traditionnelle de la RAP.

2.2.1 Modélisation Acoustique

Le modèle acoustique est un élément central de la RAP ; il est construit à partir de la modélisation d'une unité élémentaire qui est le phonème ou une variante. Le choix de cette unité est motivé par le fait que le phonème est la plus petite unité sonore identifiable et que tout mot se décline en une ou plusieurs séquences phonétiques, représentatives de sa prononciation. Cela permet d'inclure d'une manière évolutive des nouveaux mots au vocabulaire existant, sans avoir à disposer de la prononciation de tous les mots d'une langue disponible dans les données d'apprentissage. L'estimation des paramètres du modèle acoustique se fait uniquement sur la prédiction des sorties phonétiques possibles pour chaque séquence de mots.

Cette approche phonétique permet de construire un modèle acoustique probabiliste pour la suite de mots, par concaténation de phonèmes et prise en compte des différentes prononciations de chaque mot et des coarticulations possibles.

Le problème principal de cette approche réside dans l'utilisation de modèles de phonèmes indépendants de leur contexte (appelé en anglais "monophones"), ce qui ne permet pas de capturer la variation de la pro-

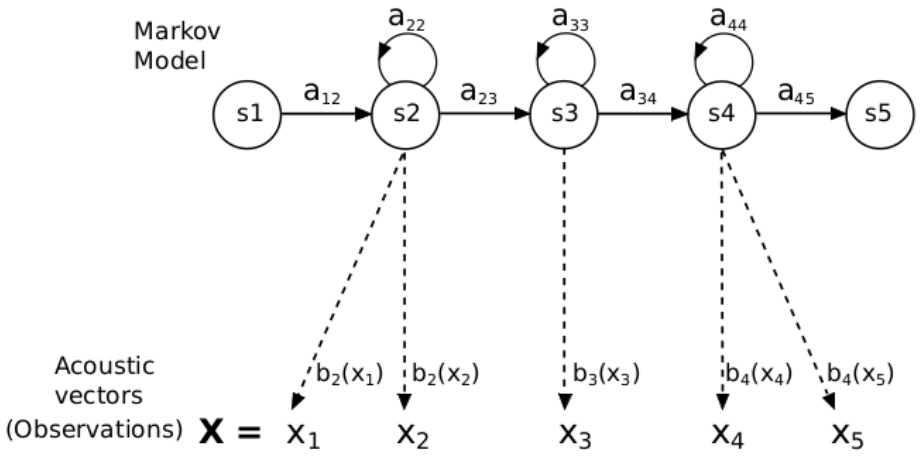


FIGURE 2.3 – Exemple de la modélisation acoustique avec la technique des HMM Povey et al. (2011)

nonciation des phonèmes dans leur contexte. Pour palier à ce problème, un contexte composé des phonèmes voisins avant et après est précisé pour capturer la variation de l'articulation. Cette modélisation contextuelle appelée en anglais : "context-dependent phone models" ou "triphone models", permettrait de constituer au maximum un ensemble de N^3 modèles pour N phonèmes de la langue cible ; des regroupements de contexte sont opérés pour réduire ce nombre.

Les différentes variantes utilisées pour la mise en place d'une modélisation acoustique sont basées sur un modèle fondamental, à savoir le **Modèle de Markov Caché (en anglais Hidden Markov Model - HMM)** . Les HMM ont pour principal intérêt de modéliser la temporalité des observations. La figure 2.3 illustre une modélisation HMM phonétique.

Un HMM permet de modéliser deux suites aléatoires : une suite d'états (s_t, t) et une suite d'observations (ou émissions) x_t, t, sous les hypothèses suivantes :

La première hypothèse est une hypothèse markovienne :

$$P(s_t|s_1^{(t-1)}) = P(s_t|s_{t-1}) \tag{2.4}$$

Où s_1^{t-1} représente la séquence d'états $s_1, s_2, ..., s_{t-1}$.

Cette hypothèse permet de limiter la dépendance d'un état à l'état précédant et non à tous les états précédents.

La deuxième hypothèse est liée à l'indépendance conditionnelle des émissions par rapport au passé qu'il s'agisse des états ou des émissions :

$$P(x_t|x_1^{t-1}, s_1^t) = P(x_t|s_t) \tag{2.5}$$

Où x_1^{t-1} représente la séquence des émissions $x_1, x_2, ..., x_{t-1}$.

Au final, un HMM (voir figure 2.3) est défini par un tuple de 4 familles de paramètres décrit comme suit :

$$\lambda = (S, A, B, \pi) \tag{2.6}$$

- $S = s_1, s_2, ..., s_N$ est l'ensemble des états. dans un temps t, le système est obligatoirement dans un des N états.
- $A = a_{i,j} = P(s_j|s_i)$ est la matrice de probabilité des transitions d'un état s_i à un état s_j dans S.
- $B = b_j(x_k) = P(x_k|s_j)$ est la matrice de probabilité d'émission. $b_j(x_k)$ est la probabilité d'émission de l'observation x_k dans l'état s_j.
- π est la matrice d'initialisation des probabilités dans S. $P(s_0 = p), \forall j \in [1, N]$. s_0 est l'état initiale du HMM.

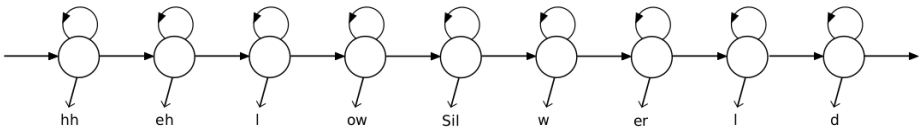


FIGURE 2.4 – Exemple de la modélisation HMM monophone pour la prononciation "Hello World" Povey et al. (2011)

Compte tenu de la définition ci-dessus du HMM, trois problèmes fondamentaux doivent être abordés afin de les appliquer à la RAP :

- Évaluation : Ayant une séquence d'observations $X = x_1, x_2, ..., x_p$ et une modélisation HMM $\lambda = (S, A, B, \pi)$. Cette tache consiste à déterminer la probabilité de $P(X|\lambda)$ qui est la probabilité que la séquence d'observation ait été généré par le modèle HMM. Ce problème est résolu par l'utilisation de l'algorithme "Forward-Backward Devijver (1985)"
- Décodage : Ayant une suite d'observations $X = x_1, x_2, ..., x_p$ et un modèle HMM $\lambda = (S, A, B, \pi)$. Cette tache consiste à déterminer la meilleure séquence d'états
- Apprentissage : Ayant une séquence d'observation $X = x_1, x_2, ..., x_p$ et un modèle HMM $\lambda = (S, A, B, \pi)$, Cette tache consite à optimiser les paramètres λ pour maximiser la probabilité d'observation X ; avec $\text{argmax}_\lambda P(X|\lambda)$. L'algorithme de "Baum-Welch" qui est un cas spéciale de la méthode "Expectation-Maximization (EM)" permet de résoudre ce problème d'apprentissage du modèle acoustique avec une modélisation HMM.

Pour plus de détails sur la description des algorithmes, nous orientons le lecteur vers les références (Baum et al. (1970); Gales et al. (2008)).

Les HMM communément utilisés pour la RAP pour modéliser chaque phonème (en contexte ou non) ont une topologie dite gauche-droite (voir figure 2.4) : les transitions sont uniquement autorisées dans le sens temporel causal (de gauche à droite) pour rendre compte de la séquentialité de la parole et des boucles sur chaque état permettent de prendre en compte la durée des sons. De fait, un HMM est capable de capturer la variabilité de la durée des phonèmes. Avec l'outil Kaldi (Povey et al. (2011)), trois états sont utilisés pour la modélisation de chaque phonème. Par exemple, en utilisant une modélisation monophone, le HMM pour la prononciation "Hello World" serait un HMM contenant les phonèmes 'hh', 'eh', 'l', 'ow', 'w', 'er', 'l' et 'd' ; comme illustré dans la figure 2.4.

La distribution probabiliste $b_j(x_k)$ évalue la probabilité d'observation de x_k dans un état s_j. Dans le domaine de la RAP, initialement il a été proposé que cette distribution soit la distribution gaussienne :

$$b_j(x_k) = \mathcal{N}(x_k; \mu^j, \sigma^j) \tag{2.7}$$

Où μ^j et σ^j sont les paramètres (moyenne, et covariance) de la gaussienne. À noter, au vu de la grande dimension du vecteur acoustique x_k, la covariance était initialement supposée diagonale.

La modélisation HMM avec une loi gaussienne comme loi d'observation convient pour des tâches de la RAP petit vocabulaire et de complexité limitée, elle ne fonctionne pas bien pour des tâches à large vocabulaire plus complexes telles que la transcription de la parole spontanée. Le problème identifié réside dans le fait que cette modélisation est insuffisante pour modéliser des formes acoustiques extrêmement variables. Différentes extensions proposant une estimation plus fine des probabilités d'observation ont été proposées pour rendre compte de conditions d'application plus larges et plus difficiles (variété des locuteurs, des environnements sonores, etc). Nous en donnons un bref aperçu ci après.

Approche GMM-HMM Il est très vite apparu qu'une meilleure approximation des lois d'émission serait obtenu avec des lois de type mélange de lois gaussiennes (en anglais "Gaussian Mixture Model (GMM)"). Cette extension permet au système RAP à base de HMM, de reconnaître des formes acoustiques plus complexes liées à la parole.

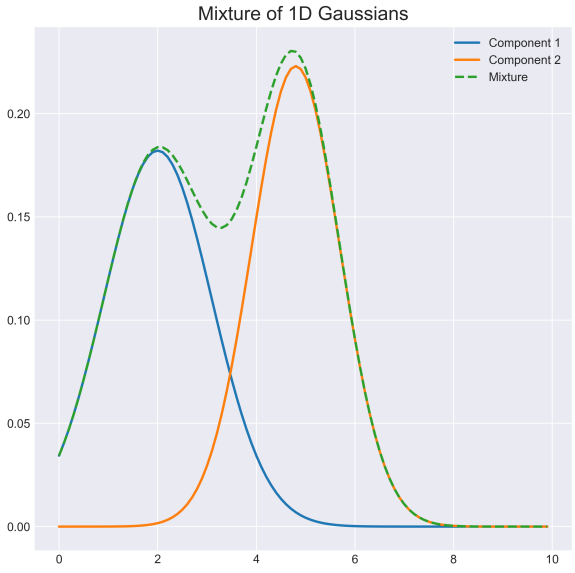


FIGURE 2.5 – Exemple de la modélisation d'une distribution à base de GMM 1D sous la forme d'une combinaison de deux composantes gaussiennes.

Chacun des M composants du modèle de mélange de loi gaussienne est une fonction de probabilité gaussienne. La probabilité d'émission de l'état s_j est la somme pondérée de toutes les probabilités des composants du mélange (voir Figure 2.5) :

$$b_j(x_k) = \sum_{m=1}^{M} c_{j,m}\mathcal{N}(x_k; \mu^{j,m}, \sigma^{j,m}) \tag{2.8}$$

Où $\mu^{j,m}$ et $\sigma^{j,m}$ sont les paramètres (moyenne et covariance) de la gaussienne m. $c_{j,m}$ est un coefficient de pondération pour chaque gaussienne, et qui satisfait, pour que $b_j(.)$ soit une densité de probabilité, la contrainte suivante :

$$\sum_{m=1}^{M} c_{j,m} = 1, c_{j,m} \geq 0 \tag{2.9}$$

Dans l'approche GMM-HMM, la difficulté a résidé et réside dans le nécessaire apprentissage des paramètres de ces lois, qui reste non optimal. En autres, il faut déterminer le nombre de composantes gaussiennes par état dans un système. Plusieurs approches ont été proposées pour pallier ce problème (Gales et al. (2008); Woodland et al. (1994)). Une approche simple consiste à utiliser le même nombre de composantes pour tous les états du système. Une autre approche consiste à faire en sorte que le nombre de lois gaussiennes attribuées à un état soit fonction du nombre d'observations attribuées à cet état (Woodland et al. (1994)). D'autres alternatives utilisent un critère bayésien pour l'affectation du bon nombre des composants GMM (Gales et al. (2006)). Généralement, le nombre de composants gaussiens pour la RAP est très large, dépassant les 100 000 composants. Ce nombre est nécessaire pour pallier aux problèmes de la variabilité acoustique de la parole. Cependant, un grand nombre de paramètres gaussiens peut engendrer des problèmes statistiques lors de leur estimation (Gales et al. (2008)).

Des techniques plus sophistiquées ont vu le jour durant l'émergence des systèmes GMM-HMM, impliquant un travail plus précis sur **l'espace de projection acoustique**. Citons :

- des matrices de covariance semi-fixes factorisées, appelées en anglais : "Semi-tied covariance Gaussians" (Gales (2001)).
- des techniques de normalisation pour éliminer la variation inter-locuteur, appelées en anglais "Vocal tract length normalization (VTLN) (Lee and Rose (1998); Zhan and Waibel (1997)"). Le VTLN permet d'estimer le conduit vocal du locuteur, et de l'utiliser pour déformer (en anglais : "warp") le signal et le rendre plus indépendant au locuteur.
- l'adaptation aux locuteurs, appelée en anglais "Speaker Adaptative Training (SAT) (Anastasakos et al. (1996)")
- les approches de normalisation de locuteur, appelée en anglais "Feature-space Maximum Likelihood Linear Regression 'fMLLR' (Gales (1998)"). Cela consiste à apprendre une transformation des paramètres du modèle GMM-HMM, de telle sorte que le modèle soit adapté au locuteur par une technique de maximisation de la probabilité d'observation. La formule ci-dessous précise la manière dans laquelle la transformation est réalisée (à noter que cette technique ne réalise pas un apprentissage des paramètres du modèle GMM-HMM, mais leur transformation pour un ensemble de données X) :

  $$\nu MLLR = \underset{\nu}{\operatorname{argmax}}\, P(X|\Theta, \nu)$$

  Avec ν les paramètres soumis à l'adaptation ; Θ représente les paramètres du modèle GMM non modifiés.
- les vecteurs d'identification, appelés "I-vectors", des vecteurs compacts de représentation du locuteur pour chaque prononciation. L'idée principale est d'apprendre à base de modélisation GMM à estimer un vecteur (i-vector) ajusté via une modélisation GMM indépendante du locuteur appelée en anglais"Ubiversal Background Model (UBM)". Cette modélisation est apprise à partir d'un corpus ayant un grand nombre de locuteurs (Saon et al. (2013)).

Approche hybride DNN-HMM L'application des réseaux de neurones (appelée en anglais : "Artificial Neural Network (ANN)") dans le domaine de la RAP remonte aux années 90 (Morgan and Bourlard (1990)). L'alternative au système traditionnel GMM-HMM consiste à développer des modèles appelés "Hybrides" combinant les avancées dans le domaine des réseaux de neurones et la modélisation HMM (Morgan and Bourlard (1990)).

L'application des ANN sur la RAP a été limitée dans le temps par plusieurs facteurs, à savoir :

— les réseaux de neurones n'utilisaient qu'une seule couche neuronale, ce qui limite l'apprentissage de représentation complexe.

— la puissance de calcul pour l'apprentissage du modèle neuronal était tout autant bloquante comparé à l'optimisation par l'algorithme EM des systèmes GMM-HMM.

Durant la dernière décennie, des avancées technologiques en terme de puissance de calcul ont été développées, premettant des optimisations de l'algorithme de rétro-propagation sous GPU, des techniques de mini-batch et de l'apprentissage stochastique. A partir de ce moment, apparaissent les réseaux de neurones profonds (en anglais Deep Neural Network, DNN) qui consistent à proposer plusieurs couches neuronales pour atteindre une modélisation plus complexe des représentations acoustiques. Dans les années 2000, apparait le système appelé système tandem (Hermansky et al. (2000)) appliquée à la RAP, qui consiste à utiliser un système GMM-HMM et à lier l'entrée des GMM à un vecteur de représentation appris à partir des observations acoustiques avec les DNN. Joffrey Hinton (Hinton et al. (2006, 2012a)) a été l'un des premiers à exploiter les techniques neuronales profondes en remplaçant le modèle acoustique GMM par des DNN dans le domaine de la RAP . L'approche "Deep Belief Network (DBN)" Hinton et al. (2006)) a été appliquée a la RAP en 2006 ; elle consiste à construire un modèle génératif à plusieurs couches qui est appris graduellement, couche par couche en utilisant les machines restreintes de Boltzmann (en anglais : "Restricted Bltzmann Machine (RBM)") à chaque couche. Il a été a observé que l'utilisation des paramètres d'un DBN pour initialiser un DNN, ensuite entraîné avec l'algorithme de rétro-propagation, permet d'obtenir de meilleures performances que les GMM-HMM (Mohamed et al. (2011)).

Plusieurs travaux ont démontré les avantages de l'utilisation de DNN pour les tâches de la RAP. Par exemple, des techniques appelées en anglais "Dropout" évitent le sur-apprentissage (Hinton et al. (2012b)) ; sont introduites de nouvelles fonctions d'activation (par exemple, appelée en anglais : "Rectified Linear Unit (RELU) (Nair and Hinton (2010))). Mais surtout l'application de nouvelles architectures neuronales à la RAP telles que les réseaux de neurones convolutionnels (CNN) (Zhang et al. (2017)), les réseaux de neurones récurrents (RNN) (Graves et al. (2013)), ainsi que les réseaux de neurones à délai ("Time Delay Neural Network (TDNN) (Povey et al. (2011))).

Modéliser des dépendances temporelles est cruciale pour la modélisation acoustique et les réseaux de neurones récurrents sont adaptés pour ce faire. Cependant, en raison des problèmes d'optimisation des RNN comme le problème de *vanishing* (convergence des paramètres vers 0) du gradient et d'explosion du gradient, une grande partie de la communauté s'est orientée vers l'utilisation des architectures TDNN, les précurseurs des réseaux convolutionels à une dimension.

Le TDNN est une architecture proposant des caractéristiques très intéressantes dans le cadre de son application au domaine de la RAP telles que :

— L'utilisation d'un contexte temprel local permet au réseau d'être moins sensible aux distortions temporelles, comparé aux autres types d'architectures RNN qui prennent en compte le contexte global des entrées. Pour la modélisation contextuelle dans un TDNN, chaque unité neurale de chaque couche reçoit une partie fixe de la série contextuelle des activations de la couche précédente. Cela permet aux couches se situant plus loin dans le réseau, d'avoir un contexte plus large pour la classification. Cette caractéristique est très intéressante pour traiter le problème du contexte phonétique (voir la figure 2.6).

— En raison des liaisons contextuelles du TDNN, les couches inférieures du réseau sont forcées d'apprendre des transformations temporelles **invariantes** des observations acoustiques.

L'un des grands problèmes des TDNN réside dans l'augmentation linéaire du nombre de paramètres par rapport au nombre de contextes d'entrée. La solution réside soit à : 1) augmenter le contexte des filtres existants (les noyaux), 2) augmenter la profondeur du réseau pour couvrir tous les contextes supplémentaires. Pour résoudre ce problème général des TDNN, qui induit a un très large chevauchement de contexte entre

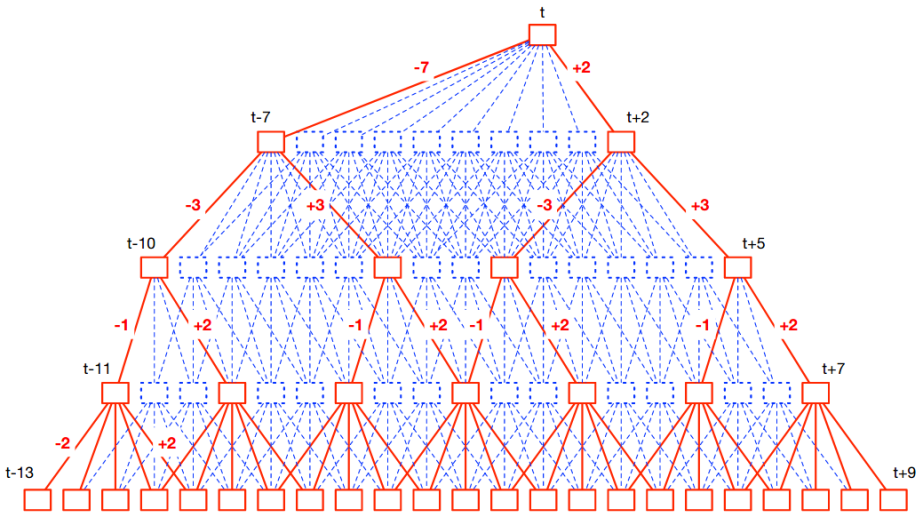


FIGURE 2.6 – L'apprentissage des TDNN avec la méthode de sous échantillonage (en rouge) et sans sous échantillonage (en bleu et rouge). Les nombres en rouge représentent la largeur du contexte choisi, à savoir : pour la couche 1, un noyau de $[-2, +2]$, pour la couche 2, un noyau $[-1, +2]$, pour la couche 3, un noyau $[-3, +3]$, et pour la couche finale, un noyau $[-7, +2]$.

les activations voisines de chaque couche, on peut réaliser une technique de sous échantillonage proposée par Vijayaditya Peddinti et al. (2015) en faisant l'hypothèse que les activations des contextes adjacents sont corrélées entre eux. Cette technique permet de gagner un ratio de 5 en temps de calcul comparé au TDNN standard. La figure 2.6 illustre cette proposition de technique de sous échantillonage des TDNN.

Une autre optimisation de factorisation des paramètres des TDNN a été proposée par Povey et al. (2018a). Le *Factorized-TDNN* consiste à découper le calcul de la matrice de paramètres M d'une couche standard du TDNN en plusieurs matrices successives $M = M_i, ..., M_k$. Povey et al. (2018a) propose d'effectuer 3 découpe (en anglais : "3-stage splicing") avec des matrices semi-orthogonales. Cette technique a l'avantage d'accélérer la phase de décodage et d'améliorer les performances de reconnaissance.

Une dernière optimisation dans l'état de l'art des approches hybrides pour la RAP, la modélisation appelée en anglais "Chain Model", a permis de remplacer les modèles conventionnels sur plusieurs aspects :

— à la place d'optimiser la fonction de minimisation du coût au niveau des fenêtres, la fonction est calculée sur la probabilité en log de la séquence correcte de phonème. Cela permet de réduire par 3 le nombre de sorties du réseau et surtout d'avoir une meilleure convergence.

— à cause de cette réduction de fréquence de fenêtre d'analyse, la topologie HMM n'a plus besoin de 3 états/transitions pour modéliser la temporalité. Un seul état suffit au vu de l'optimisation proposée au point précédent.

2.2.2 Modélisation Linguistique

Le modèle de langage (LM) a pour objectif de capturer automatiquement la connaissance linguistique (syntaxe, semantique, etc) d'une langue donnée. Un texte d'apprentissage permet d'estimer la probabilité de chaque mot pour par rapport à sont contexte. Son utilisation principale est de quantifier conditionnellement la probabilité d'un mot sachant la suite de mots donnée. La probabilité d'une séquence de mot $W = w_1, w_2, ..., w_k$, définie par $P(W)$ dans l'équation bayesienne 2.3 est calculée comme suit :

$$P(W) = \prod_{i=1}^{k} P(w_i|w_{i-1}, w_{i-2}, ..., w_1) \tag{2.10}$$

La longueur k d'une séquence de mots est inconnue et peut être importante ; le calcul de $P(W)$ devient très lourd ainsi que son apprentissage. Or, l'information linguistique d'une très longue séquence de mots n'est pas en soit très informative, le contexte d'une partie de phrase peut être compris avec plus ou moins 5 mots en général. Par conséquent, on utilise une modélisation appelée en anglais "n-gram LM" dans laquelle chacune des probabilités dans l'équation 2.10 est calculée en supposant que seuls les $n-1$ mots les plus récents sont pertinents pour la prédiction du mot suivant. Le nombre n est généralement appelé l'ordre du LM. Par exemple, si l'ordre du LM est 1, on parle d'un LM unigram (aucun mot précèdent n'est considéré ; $n = 1$), d'un LM bigram (un mot est considéré ; $n = 2$), d'un LM trigram ($n = 3$), etc. La probabilité de la séquence de mots $P(W)$ avec un modèle LM "n-gram" peut être calculé avec l'équation suivante :

$$P(W) = \prod_{i=1}^{k} P(w_i|w_{i-1}, w_{i-2}, ..., w_{i-n+1}) \tag{2.11}$$

Le modèle "n-gram" est construit à partir d'un texte d'apprentissage. Le calcul de ces probabilités est basé sur le comptage de "n" occurrences de la séquence précédant chaque mot. A titre d'exemple, pour le cas d'un LM 3-gram, $C(w_{i-2}, w_{i-1}, w_i)$ est le comptage du nombre d'occurrences de la séquence w_{i-2}, w_{i-1}, w_i, et le comptage $C(w_{i-2}, wi-1$ pour la séquence $w_{i-2}, wi-1$. On peut donc écrire l'équation de calcul suivante :

$$P(w_i|w_{i-1}, w_{i-2}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \tag{2.12}$$

2.2.3 Phase de Décodage

Comme mentionné dans l'introduction de cette section, le processus de décodage consiste à rechercher la séquence de mots la plus probable $\widehat{W}$, générée à partir d'observations acoustiques X parmi toutes les séquences d'états S des hypothèses des mots possibles.

Le Processus de décodage est lié au domaine de recherche dans les graphes. Dans la RAP, ce processus de recherche appelé aussi "decoding" doit être effectué dans un délai très court, alors que l'espace de recherche est généralement très grand à cause de deux facteurs principaux : le premier est lié à l'ordre du LM qui étend fortement l'espace de recherche. Le deuxième facteur est en lien avec les problèmes liés à la parole continue à large vocabulaire. L'espace de recherche des mots augmente de façon exponentielle en fonction du nombre de mots prononcés et les limites ambiguës des mots rendent la recherche plus difficile.

Un moyen efficace pour résoudre ce problème de recherche dans un graphe est d'appliquer l'algorithme "Viterbi search" (Povey et al. (2011)). Cet algorithme utilise une approche de type programmation dynamique pour calculer la meilleure extension étant donnée les hypothèses précédentes calculées à l'itération précédente. Cet algorithme, est couplé à une recherche en faisceau"beam research" qui définit un nombre maximal d'extensions du graphe à une itération donnée. Pour ce faire, une approche basée sur les automates à états finis (en anglais : "Weighted Finite State Transducer (WFST)") a été proposée. Cette approche offre une très grande flexibilité et une très grande efficacité d'optimisation du graphe combinant le graphe de prononciation (appelé en anglais : "Lexicon (L.fst)") permettant de lier la partie acoustique et le modèle de langage, ainsi que le graphe de langage (appelée en anglais : "Grammar (G.fst)").

Les sorties de la phase de décodage Bien que les décodeurs soient principalement conçus pour trouver la meilleure séquence de mots, il est quand même intéressant de générer plusieurs meilleures hypothèses, qui peuvent être presque aussi probables que la meilleure séquence. Cela est extrêmement utile et permet d'utiliser le graphe des hypothèses pour faire de la recherche de mots clés par exemple, mais aussi d'interpréter le processus de décodage et les difficultés liées à la confusion des mots.

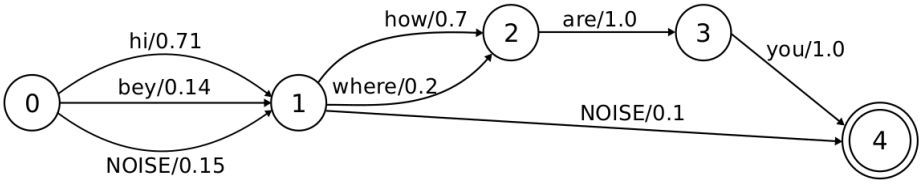


FIGURE 2.7 – Exemple du treillis de mots Povey et al. (2011)

Le graphe des hypothèses est couramment donné sous forme de treillis de mots, appelé en anglais : “word lattice” (Chappelier et al. (1999); Richardson et al. (1995); Povey et al. (2011)) et rassemble les N-meilleures hypothèses possible (en anglais : “N-best hypotheses”) possédant les plus grande probabilités a posteriori. L'exemple suivant représente les 3-meilleures hypothèses avec leur probabilité à posteriori :

— “hi how are you” 0.49
— “hi where are you” 0.14
— “bey how are you” 0.1

Cet exemple de treillis de mots est illustré à la figure 2.7. Un treillis de mots est composé d'un ensemble de noeuds représentant des états dans le temps et d'un ensemble d'arcs qui relient deux noeuds consécutifs. Le mot et sa probabilité a posteriori sont associés à chaque arc. Une hypothèse est une séquence d'arcs du début au noeud final.

Dans la pratique, la représentation en treillis de mots est très flexible. Elle peut être étendue pour permettre le recalcul des scores par un modèle linguistique d'ordre supérieur. De plus, le score acoustique et la probabilité *a posteriori* sont associées à chaque hypothèse pour en déterminer la qualité.

2.3 Approches End-to-End (E2E)

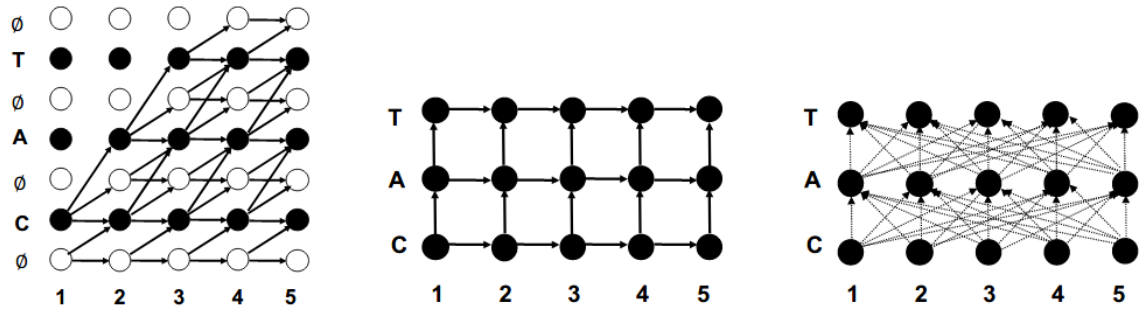
Les approches End-to-End représentent un nouveau paradigme émergeant des réseaux de neurones. Appliquées à la RAP, elles offrent l'avantage de lier directement la séquence audio avec la transcription, sans modélisation intermédiaire, alors que les systèmes "hybrides" traditionnels, au travers d'une approche bayésienne, impliquent deux modélisations acoustique et linguistique, supposées indépendantes qui rend nécessaire un apprentissage séparé de ces composants, comme vu au paragraphe 1.2.

L'approche E2E est une approche dite tout-en-un, avec une phase d'apprentissage beaucoup plus simple, qui permet de prédire des sorties graphèmes (caractères, en général) directement à partir des observations acoustiques. Cette approche a comme avantage de permettre l'optimisation conjointe de la partie acoustique et linguistique en un seul modèle. De plus, ce type d'approche profite du fait de disposer de sorties de type "graphème" pour proposer de nouveaux mots, alors que les approches traditionnelles se heurtent au problème des mots hors-vocabulaire.

Cependant,les systèmes RAP E2E actuels souffrent également de limitations importantes liées à la quantité de données d'entraînement nécessaires ; elle est généralement supérieure d'un ordre de grandeur, à celle des systèmes RAP "hybrides" pour obtenir un taux d'erreur de mots (Word Error Rate-WER) similaire. Cette limitation peut conduire à un sur-apprentissage des systèmes RAP E2E. Durant cette dernière décennie, ce problème a été plus au moins résolu pour des langues telles que l'anglais, le chinois et le japonais, oû l'on dispose de données en quantités suffisantes.

Les architectures RAP E2E relèvent de trois grandes catégories (voir la figure 2.8) :

— (a) l'approche “Connectionist Temporal Classification (CTC)” (Graves et al. (2006); Graves and Jaitly (2014); Amodei et al. (2016); Seide et al. (2011)) permet d'aligner des séquences d'entrées et de sorties en calculant tous les alignements possibles appelés chemins, puis en appliquant une agrégation de ces alignements. Cette technique se base sur l'ajout d'une sortie "blanc". Une limite du CTC réside dans l'hypothèse qui considère les sorties comme étant indépendantes entre elles.



(a) CTC (b) RNN-Transducer (c) Attention

FIGURE 2.8 – Aperçu des chemins possibles pour les trois approches RAP E2E : (a) l'approche CTC, (b) l'approche RNN-Transducer, (c) l'approche encodeur-decodeur avec attention.

— (b) l'approche RNN-Transducer (Graves (2012); Li et al. (2019b); Rao et al. (2017)) permet également au même titre que le CTC de calculer tous les alignements possibles et les regrouper. Le RNN-Transducer, à l'inverse du CTC, ne fait pas l'hypothèse de l'indépendance des sorties, la définition de chemin s'en trouve différente pour prendre en considération la dépendance entre sorties.
— (c) l'approche encodeur-decodeur avec "attention" (Bahdanau et al. (2016); Chorowski et al. (2015)) n'énumère plus tous les chemins possibles, mais elle utilise le mécanisme dit d'attention entre l'encodeur et le décodeur pour apprendre l'alignement entre la séquence d'entrées et de sorties.

Durant ma thèse, je me suis intéressé aux approches (a) et (b). De ce fait, je ne détaille ci-après que ces deux approches afin de préciser ensuite mes contributions : une première contribution sur l'apprentissage multi-tâche CTC est proposée au chapitre 5 ; une deuxième contribution sur la partie RNN-Transducer est décrite dans le chapitre 6. La modélisation séquentielle par réseau de neurone récurrent reste au coeur de ces approches et une connaissance de leur fonctionnement est indispensable ; j'en rappelle les bases dans la dernière section de ce chapitre 2.4.

2.3.1 Approche End-to-End CTC

Pour donner le principe E2E des CTC, leur modélisation et apprentissage, ainsi que les techniques de décodage liées, introduisons d'une part la séquence d'observations acoustiques $x = x_1, x_2, ..., x_T$ de longueur $|x| = T$, sa séquence de graphèmes associée $y = y_1, y_2, ..., y_U$ avec $|y| = U$, sa longueur correspondante. L représente l'ensemble des graphèmes (ou caractères alphabétiques) à prédire. Compte tenu de la contraction temporelle, naturelle en RAP, entre T et U, la modélisation E2E CTC, basée sur l'alignement, nécessite de rajouter à l'ensemble de sorties L une sortie additionnelle, appelée *blank* (Bridle (1990); Graves et al. (2006)), que nous noterons ϵ dans les équations suivantes.

Sachant la séquence acoustique x et l'ensemble d'apprentissage S, la probabilité conditionnelle d'observer une séquence d'étiquettes $\pi = \pi_1, \pi_2, ..., \pi_T$ de longueur T, étiquettes appartenant à l'ensemble $L' = L \cup \{\epsilon\}$, peut être trouvée en multipliant les probabilités élémentaires correspondantes $P(\pi_t|x, S)$; pour ce faire, les étiquettes à chaque instant t sont supposées conditionnellement indépendantes, étant donnés la séquence d'observation x et l'ensemble d'apprentissage S.

Plus formellement, on peut écrire :

$$p(\pi|x, S) = \prod_{t=1}^{T} P(\pi_t|x, S) \tag{2.13}$$

Dans le domaine de la RAP, il peut être avantageux de conditionner également les probabilités de transition entre étiquettes, en particulier pour traduire des relations connues entre les mots. Nous abordons cette question plus en détails dans la partie 2.3.3, où est développé l'algorithme de décodage CTC incluant des probabilités de transition.

Le CTC peut être mis en oeuvre avec n'importe quel modèle pour estimer les probabilités $P(\pi_t|x, S)$. Cependant, à cause de la propriété de dépendance conditionnelle à toute la séquence d'observations, les réseaux de neurones récurrents bidirectionnels sont très souvent utilisés : l'activation d'un neurone de sortie, associé à l'étiquette π_t, à l'instant t est interprétée comme la probabilité conditionnelle $P(\pi_t|x, S)$.

Terminologie et notation, à des fins de simplification d'écriture :

— *Un élément $\pi \in {L'}^T$ est appelé un* ***chemin****.*

— *Nous omettrons d'indiquer explicitement la dépendance des probabilités de sortie à l'ensemble d'apprentissage S.*

— *Afin de relier toute séquence de L' à une séquence possible de L, est introduite la transformation, notée B :*

$$B : L'^* \rightarrow L^*.$$

avec L'^* l'ensemble des suites de longueur finie d'éléments de L', L^* celui des suites définies à partir de L. Si $\pi \in {L'}^T$, $B(\pi) \in L^{\leq T}$. Exemple : $B(ab\epsilon) = B(\epsilon aa\epsilon\epsilon abb) = aab$).

La probabilité d'observer une suite d'éléments l de L, à partir de la suite d'observations acoustiques x s'écrit :

$$p(l|x) = \sum_{\pi \in B^{-1}(l)} p(\pi|x) = \sum_{\pi \in B^{-1}(l)} \prod_{t=1}^{T} P(\pi_t|x, S) \tag{2.14}$$

En théorie, la sortie du classifieur s'obtient en choisissant la suite l la plus probable.

$$l^* = \underset{l}{\operatorname{argmax}}\, p(l|x) \tag{2.15}$$

Les deux paragraphes suivants donnent les méthodes utilisées afin d'atteindre ces formules, à la base des algorithmes d'apprentissage et de décodage.

2.3.2 Calcul de $p(l|x)$: l'algorithme CTC forward-backward

L'équation 2.14 énonce que $p(l|x)$ se calcule en sommant sur tous les chemins correspondant à la dite séquence : pour une séquence de sorties de longueur U et une séquence d'entrées de longueur T, on aura à calculer $2^{T-U^2+U(T-3)} * 3^{(U-1)(T-U)-2}$ chemins possibles, ce qui est en soit très lourd en calculs, donc très problématique (voir Figure 2.9).

Ce problème peut être résolu avec des méthodes de type programmation dynamique similaire à l'algorithme forward-backward pour les HMM (Rabiner (1989)). L'idée principale consiste à transformer la somme sur tous les chemins correspondant à la sortie l en une somme itérative de chemins correspondant aux préfixes possibles de cette suite, puis appliquer le règle de Bayes. Néanmoins, compte tenu de la sortie additionnelle correspondant au caractère *blank*, il est nécessaire de raisonner sur une séquence d'étiquettes l' définie par l'addition de ϵ au début et à la fin de la séquence ainsi qu'entre chaque paire de graphèmes consécutifs. La longueur de l' est $2|l| + 1$. A titre d'exemple :

$$l = (l_1, l_2, l_3, l_4), l' = (\epsilon, l_1, \epsilon, l_2, \epsilon, l_3, \epsilon, l_4, \epsilon)$$

Pour une séquence l, on définit une variable $\alpha_{t,s}$ comme étant la somme des probabilités de tous les chemins partiels $\pi_{1:t}$ associés à la séquence $l'_{1:s}$, préfixe de la séquence l', sachant qu'est observée la suite x.

$$\alpha_{t,s} = \sum_{\pi_{1:t} \in B^{-1}(l'_{1:s})} \prod_{t'=1}^{t} P(\pi_{t'}|x) \tag{2.16}$$

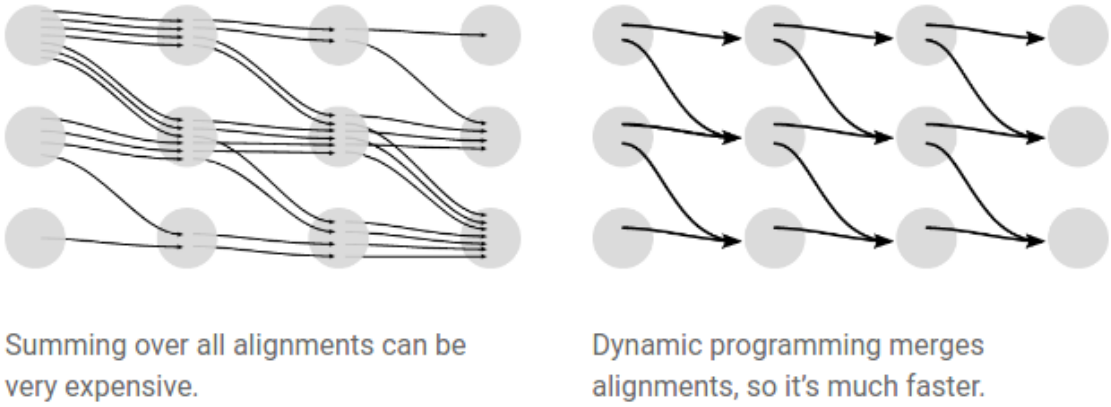


FIGURE 2.9 – Visualisation du calcul de la probabilité $p(l|x)$. La figure de gauche calcule la probabilité CTC pour une séquence d'observations et tous les chemins possibles sont calculés. La figure de droite utilise des techniques de programmation dynamique pour se limiter à un calcul récurrent (`https://distill.pub/2017/ctc/`)

.

Cette variable se calcule par récurrence, en fonction des chemins empruntés jusqu'à l'instant $t-1$ possibles, ce qui donne deux cas, selon la valeur de π_t et les valeurs de $l'_{(s-2):s}$ possibles :

— Cas (1) : si $l'_s = \epsilon$ (respectivement si $l'_{s-2} = l'_s$ (différent de ϵ)), les seuls chemins préfixes possibles vérifient $\pi_{t-1} = \epsilon$ ou $\pi_{t-1} = l'_{s-1}$ (respectivement $\pi_{t-1} = l'_{s-1} = \epsilon$ et $\pi_{t-1} = l'_s$).

— Cas (2) : dans tous les autres cas, s'ajoute un chemin vérifiant $\pi_{t-1} = l'_{s-2}$.

Il en résulte que la probabilité conditionnelle $\alpha_{t,s}$ du préfixe l'_s à un instant t est calculée à partir de la probabilité d'apparition $p(\pi_t = l'_s|x)$ du préfixe de l'_s, multipliée par la somme des probabilités des deux ou trois chemins possibles pour trouver la séquence l'_s, ce qui s'écrit :

$$\alpha_{t,s} = \begin{cases} (\alpha_{t-1,s-1} + \alpha_{t-1,s})p(\pi_t = l'_s|x) & (1) \\ (\alpha_{t-1,s-2} + \alpha_{t-1,s-1} + \alpha_{t-1,s})p(\pi_t = l'_s|x) & (2) \end{cases}$$

La figure 2.10 illustre les deux cas possibles pour le calcul de $\alpha_{t,s}$. La figure 2.11 illustre l'évolution du calcul des préfixes avec cet algorithme de programmation dynamique.

Il y a deux possibilités de départ à l'instant $t = 1$, ainsi que deux chemins possibles à la fin par rapport au rajout d'un blanc. Chaque alignement valide a un chemin dans ce graphe. Il en résulte que

$$P(l|x) = \alpha_{T,2|l|+1} + \alpha_{T,2|l|}$$

De manière analogue est définie la variable $\beta_{t,s}$

$$\beta_{t,s} = \sum_{\pi_{t:T} \in B^{-1}(l'_{s:|l|})} \prod_{t'=t}^{T} P(\pi_{t'}|x) \tag{2.17}$$

Elle se calcule par récurrence et elle sera utilisée, comme l'est la variable $\alpha_{t,s}$, dans la mise en oeuvre de l'algorithme d'apprentissage.

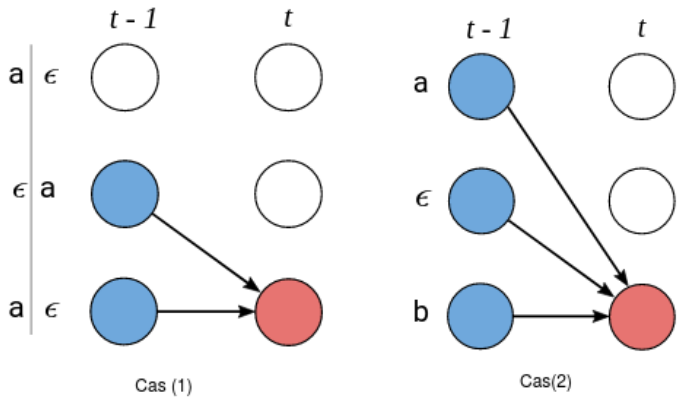


FIGURE 2.10 – Illustration des deux cas possibles pour l'extension d'un préfixe $\alpha_{t,s}$. (source : https ://distill.pub/2017/ctc/)

2.3.3 Apprentissage et Décodage dans le cadre CTC

Apprentissage CTC Nous situant dans le cadre d'une approche probabiliste, l'apprentissage du modèle RNN est réalisé par maximum de vraisemblance. Pour un ensemble d'apprentissage $S = (X, Y)$, les paramètres du modèle sont appris en minimisant la fonction de perte, égale à l'opposé du logarithme de vraisemblance, qui s'écrit comme suit :

$$\sum_{X,Y \in S} -\log p(Y|X) \tag{2.18}$$

La fonction de perte CTC est **différentiable** par le fait que l'équation est une combinaison de sommes et de produits entre les probabilités d'observations à chaque instant. Nous pouvons calculer analytiquement le gradient de la fonction de perte par rapport aux probabilités de sortie (non normalisées), qui s'expriment en fonction des variables α et β et, à partir de là, effectuer une rétro-propagation pour la mise à jours des poids du modèle.

Décodage CTC Une fois le modèle RNN entrainé, nous souhaitons à partir d'une séquence d'observations x en entrée, décoder la suite la plus probable de graphèmes associé l^*, ce qui s'écrit :

$$l^* = \underset{l}{\operatorname{argmax}}\, p(l|x) \tag{2.19}$$

Pour résoudre cette équation 2.19, deux solutions sont principalement proposées, basées sur des heuristiques classiques : la recherche du meilleur chemin et la recherche en faisceau.

Recherche du "meilleur" chemin Cette heuristique fait référence au décodage appelée en anglais "Best path décoding", cette technique permet de nous fournir un alignement de l'audio et de la sortie graphème en se basant sur la probabilité la plus haute à chaque instant t, on peut écrire :

$$l^* = B(\pi^*)$$
$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, p(\pi|x)$$

Le calcul du meilleur chemin est trivial puisqu'il suffit de concaténer les sorties π^* en enlevant toute les répétitions puis les blancs. Cependant, la solution est sous optimale au sens probabiliste : la meilleure suite de caractères devrait se déduire d'une somme sur tous les chemins antécédents π ; ce calcul sous optimale

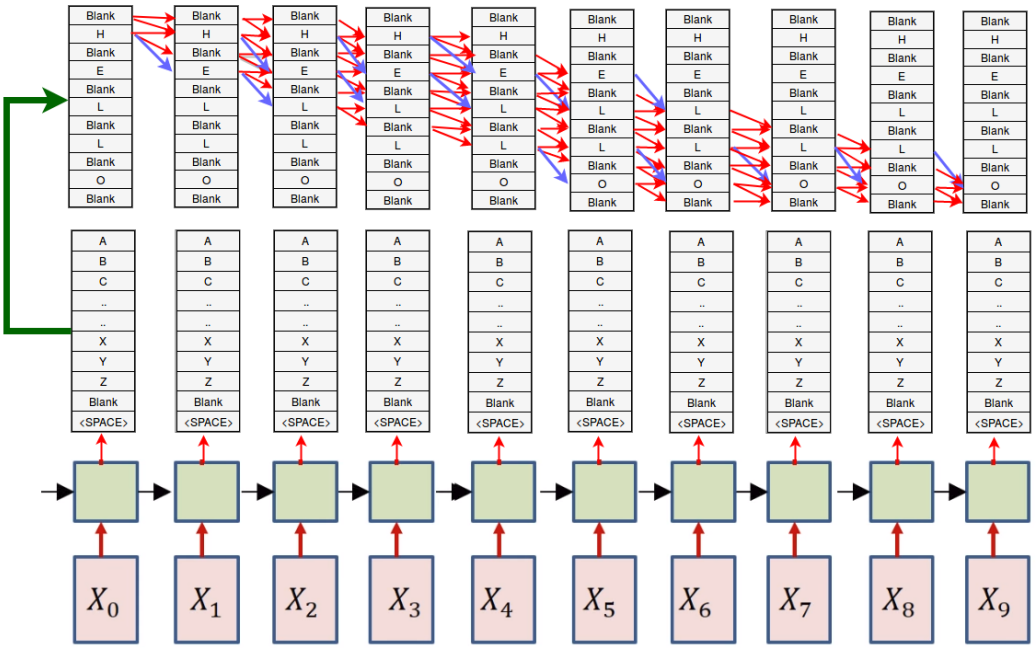


FIGURE 2.11 – Exemple de calcul de la probabilité CTC avec la technique de programmation dynamique pour la transcription $Y = [h, e, l, l, o]$.

peut commettre des erreurs, particulièrement si la probabilité du caractère n'est pas très haute sur plusieurs instants consécutifs (voir Figure 2.12).

Prefix beam search Avec la recherche en faisceau, appelée en anglais "Best-first search" (Norvig and Intelligence (2002)) (ou bien :"Vanilla beam search"), l'idée est de construire progressivement un arbre des étiquetages où chaque feuille d'un étiquetage donné permet d'étendre le préfixe avec une nouvelle étiquette. À chaque étape, cette méthode étend le préfixe par l'étiquette dont la feuille a la plus grande probabilité cumulée (voir figure 2.13) en explorant k feuilles à la fois, d'où le nom "beam" en anglais (ce "beam" est utilisé pour limiter le nombre d'extensions durant cette phase de décodage).

Une modification de l'algorithme "Prefix beam search" a été proposé (Graves et al. (2006); Graves and Jaitly (2014)) en prenant en compte tous les chemins possibles ayant la même sortie. Cette technique se base sur l'extension d'un chemin telle qu'utilisée dans le calcul de la variable α). Dans ce cas, au lieu de conserver une liste d'extension indépendante par "beam", sont stockées les préfixes de sortie après avoir réduit les répétitions et supprimé les *blank*. À chaque étape de la recherche, nous accumulons les probabilités pour un préfixe donné en fonction de tous les chemins qui lui correspondent et sont conservés les chemins composant les meilleurs sommes. La figure 2.14 illustre cette nouvelle modification de l'algorithme. Cette approche permet d'introduire un modèle de langage à l'algorithme de recherche. Un LM de caractères ou de mots peut être rajouté de la manière suivante : si le LM est au niveau caractère, la probabilité du LM est ajoutée à chaque extension. Si le LM est au niveau mot ou bien une contrainte binaire d'appartenance à un lexique, l'algorithme inclut cette probabilité à la prédiction d'un espace (transition entre deux mots).

2.3.4 Approche End-to-End RNN-Transducer

La solution proposée par l'approche CTC a accéléré le passage vers les approches RAP E2E durant les deux dernières décennies. Cependant, le CTC a deux contraintes majeures qui limitent son efficacité :

— Le étiquettes sont supposées conditionnellement indépendantes ; cette hypothèse nécessite l'inclusion

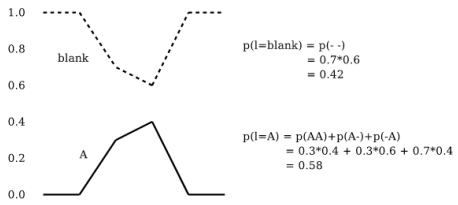


FIGURE 2.12 – **Problème du meilleure chemin de décodage :** Le chemin le plus probable ne contient pas d'étiquettes, et le meilleur décodage du chemin produit donc un étiquetage "vide". Toutefois, la somme des probabilités des chemins correspondant à l'étiquetage "A" est plus élevée. Graves et al. (2006)

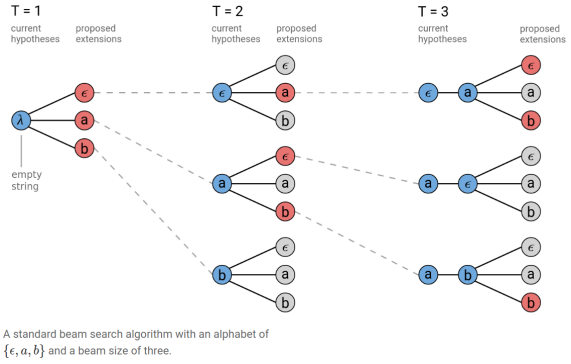


FIGURE 2.13 – Le **Beam search algorithm** ou algorithme de recherche en faisceau calcule un nouvel ensemble d'hypothèses à chaque étape d'exploration d'un arbre. Le nouvel ensemble d'hypothèses est généré à partir de l'ensemble précédent en étendant chaque hypothèse avec tous les caractères de sortie possibles et en ne conservant que les k meilleurs candidats à chaque instant (source : https ://distill.pub/2017/ctc/).

de l'étiquette *blank* entre chaque caractères de sortie pour résoudre le problème d'alignement de bout en bout, mais élimine l'apprentissage implicite du lien entre les caractères de sortie, tel qu'il existe avec un modèle de langage. Le modèle CTC doit être considéré comme un modèle acoustique et un décodage avec un modèle de langage externe tel qu'utilisé dans (Amodei et al. (2016)) ou avec un graphe WSFT (Miao et al. (2015)) est nécessaire.

— Le CTC peut lier la séquence d'entrée avec sa séquence de sortie correspondante uniquement si la longueur de sortie est inférieure à celle de l'entrée. Cela peut être bloquant pour une modélisation avec une réduction agressive de facteur 8 ou 16.

En RAP, il est clair que le premier point a un grand impact ; l'approche RNN-T proposée par Graves (2012) est définie comme une extension de l'approche CTC en associant deux RNN, l'un, l'encodeur" assurant la modélisation acoustique et l'autre, le "prédictif ou décodeur", rendant compte de la dépendance des sorties entre elles. Le RNN-Transducer (RNN-T) est défini comme un modèle auto-régressif joignant l'information acoustique et l'information linguistique pour prédire la séquence de sortie.

Plus précisément, l'architecture du modèle RNN-T se compose de trois réseaux de neurones (figure 2.15) :

— *l'encodeur*, en anglais *the Transcription Network (F^{enc})*. Ce réseau de neurones joue le rôle du modèle acoustique : il extrait des observations acoustiques $X = [x_1, ..., x_T]$ de longueur T l'information

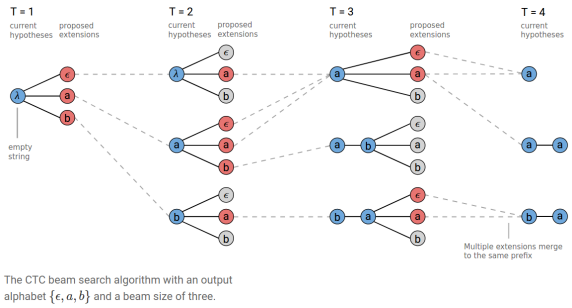


FIGURE 2.14 – **CTC Prefix beam search** est un algorithme modifié à partir de l'algorithme de *beam search* pour traiter les alignements multiples qui correspondent à une même sortie, pour un ensemble de caractères $\{a, b\}$, une séquence d'observations de longueur $T = 3$ et un beam de 3. Une extension du préfixe est autorisée si le dernier caractère du préfixe est répété. Cela est illustré dans le cas où "a" est proposé comme extension du préfixe [a] ([a] et [a, a] sont tous deux des valeurs de sortie valables pour l'extension proposée). La fusion des extensions de préfixes ayant le même préfixe est aussi autorisée par le PBS (voir l'arbre en bas à droite). (source : https ://distill.pub/2017/ctc/)

pertinente au travers d'une suite $h^{\text{enc}} = [h_1^{\text{enc}}, ..., h_T^{\text{enc}}]$ de meme longueur.

$$h_t^{\text{enc}} = F^{\text{enc}}(x_t) \tag{2.20}$$

— *le décodeur*, en anglais *the Prediction Network (F^{pre})*. Ce réseau de neurones joue le rôle du modèle de langage et modélise précisément la dépendance entre les sorties (*output-output dependency*). Ce réseau produit une représentation de haut niveau h_u^{pre}, prédiction de proche en proche, conditionnée par le caractère précédent y_{u-1}.

$$h_u^{\text{pre}} = F^{\text{pre}}(y_{u-1}) \tag{2.21}$$

— *la fusion*, en anglais *the Joint Network (F^{joint})*. Ce réseau de neurones conclut le processus de classification à partir de la combinaison des sorties de l'encodeur h^{enc} et des sorties du décodeur h^{pre}, ce qui conduit à l'évaluation de la probabilité a posteriori de chaque label. Cette combinaison peut être calculée par la somme ou la concaténation des deux représentations. A titre d'exemple, dans le cas d'une fusion par somme, la probabilité a posteriori est obtenue de la manière suivante :

$$z_{t,u} = F^{\text{joint}}(h_t^{\text{enc}}, h_u^{\text{pre}}) \tag{2.22}$$

$$= \tanh\left(U h_t^{\text{enc}} + V h_u^{\text{pre}} + b\right) \tag{2.23}$$

où U et V et b sont les paramètres de la transformation linéaire appliquée.
La probabilité de distribution des labels en position u à l'instant t s'écrit :

$$h_{t,u} = W_y z_{t,u} + b_y \tag{2.24}$$

$$P(k|t,u) = \text{Softmax}\left(h_{t,u}\right) \tag{2.25}$$

où $h_{t,u}$ est obtenu par une transformation linéaire avec les paramètres W_y et b_y.

On peut constater que $P(k|t,u)$ est une fonction de h^{enc} et h^{pre}. Au regard de l'architecture RNN-T, h_t^{enc} provient de la séquence d'entrée à l'instant x_t, alors que h_u^{pre} à la position u nous provient de l'historique de séquence de labels $[y_1, ..., y_{u-1}]$. Le réseau "Joint network" permet de joindre les deux informations au temps t et à la position u pour produire la probabilité de distribution du label courant, on parle alors de modèle auto-régressif.

Avec cette modélisation, le RNN-T permet de calculer tous les chemins possibles (figure 2.16) en se basant sur la matrice des probabilités P de dimension $(T, U+1, |L'|)$ avec T la longueur maximale de la

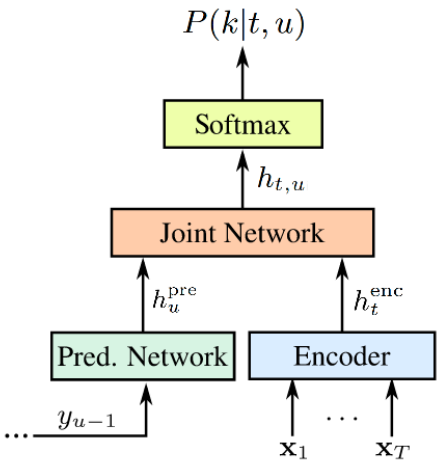


FIGURE 2.15 – Structure du **RNN-Transducer** (RNN-T) (Prabhavalkar et al. (2017)).

trame audio, U la longueur de la séquence de sortie y (l'étiquette $\varnothing$ est ajoutée au début pour la prédiction de la séquence) ainsi que $|L'|$ le cardinal de l'ensemble L' des étiquettes possibles, $\varnothing$ inclus. Comme dans le cadre du CTC, la fonction de coût RNN-T est égale à l'opposé du logarithme de vraisemblance de la séquence de sortie y^* sachant la séquence d'entrée x et les paramètres du modèle sont appris en la minimisant. Tous les calculs s'appuient sur un algorithme de type "Forward-Backward", comme résumé ci après :

$$L = -\ln P\left(y^*|x\right) \tag{2.26}$$

$$P(y^*|x) = \sum_{(t,u):t+u=n} \alpha(t,u)\beta(t,u) \tag{2.27}$$

$$\tag{2.28}$$

où les matrices 'forward' α et 'backward' β sont calculées par récurrence comme suit :

$$\alpha(t,u) = \alpha(t-1,u)\varnothing(t-1,u) + \alpha(t,u-1)y(t,u-1) \tag{2.29}$$

$$\beta(t,u) = \beta(t+1,u)\varnothing(t,u) + \beta(t,u+1)y(t,u-1) \tag{2.30}$$

avec

- $\alpha(1,0) = 1$
- $y(t,u-1)$ la probabilité de prédiction du label u à l'instant t
- $\varnothing(t,u)$ la probabilité de prédiction de n'observer aucune nouvelle étiquette à l'instant $t+1$
- $\beta(T,U) = \varnothing(T,U)$

De manière pragmatique, le RNN-T en phase de décodage permet, à chaque fois qu'une entrée x_t est encodé par le Transcription Network, le Prediction Network prédit une suite d'étiquettes jusqu'à ce que l'étiquette $\varnothing$ soit prédite avec la probabilité $\varnothing(t,u)$, le RNN-T passe à l'observation suivante x_{t+1} et le processus est itéré. Étant donnée qu'une entrée x_t peut produire une sous-séquence de label, le RNN-T peut gérer la situation où la longueur de la sortie est supérieure à celle de l'entrée. Cette propriété du RNN-T fait de lui un système auto-régressif.

À titre d'exemple, est présenté ci-joint, l'algorithme "Greedy Search" 1 utilisé en phase de décodage d'un système RNN-T : à chaque instant t, le décodage GS d'un système RNN-T permet d'émettre une ou plusieurs prédictions, représenté par la boucle "While", la prédiction du RNN-T continue en prenant

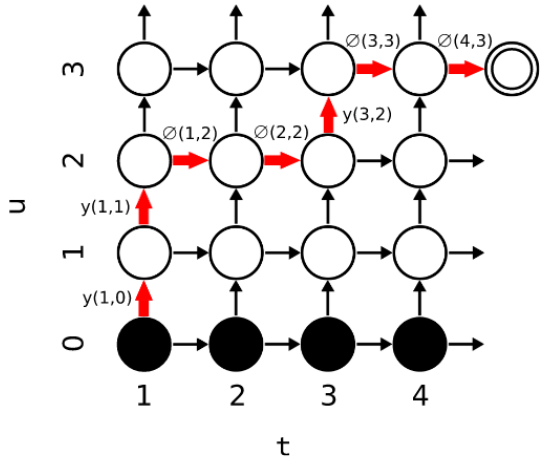


FIGURE 2.16 – Calcul de tous les chemins possibles avec l'algorithme forward-backward. À chaque noeud (t, u) est associée la probabilité d'observer les u premières étiquettes connaissant la séquence d'entrée $x_{1:t}$. Le passage horizontal à partir du noeud (t, u) implique l'observation d'aucune nouvelle étiquette ; le passage vertical représente la probabilité de prédiction du label $u + 1$ de la séquence y. (Graves (2012)).

en compte l'historique de la séquence de prédiction au temps t tant que celui-ci n'a pas prédit d'émission $\varnothing(t, u)$. Cette dernière représente le passage au décodage de l'observation au temps $t + 1$.

Algorithm 1 Soit une matrice d'observation acoustique X (à deux dimensions) $[T, O]$ représentée par (la longueur de l'observation et la dimension du vecteur d'observation). L'observation X est encodée par le composant encodeur du RNN-T F^{enc} et permet de générer la représentation h^{enc} de dimension $[T, \text{hidden\_size}]$. Le décodeur F^{dec} du RNN-T est initialisé par un historique $Null$ au début de la séquence et la probabilité associée $\varnothing(0,0)$. Le décodage GS sur le RNN-T parcours la sortie de l'encodeur h_t^{enc} à chaque instant t pour réaliser la jointure entre le vecteur de représentation caché de l'encodeur h_t^{enc} et le vecteur de représentation caché du décodeur h_{curr}^{dec} à la position courante $curr$. La boucle $WhileLoop$ génère plusieurs prédictions au même temps t et le passage d'un temps t à un temps $t+1$ se fait obligatoirement par le biais d'une transition $\varnothing(t, curr)$.

hyp $\leftarrow \{\}$
$h^{enc} \leftarrow F^{enc}(X_{1,\dots,T})$
$h_{curr}^{dec}, hidden \leftarrow F^{dec}(\varnothing(0,0), hidden = Null)$
for $h_t^{enc} \in h^{enc}$ **do**
  $h_{t,curr}^{joint} \leftarrow F^{joint}(h_t^{enc}, h_{curr}^{dec})$
  $k \leftarrow \underset{k}{\text{argmax}}\,\text{Softmax}\left(h_{t,curr}^{joint}\right)$
  while $k \neq \varnothing$ **do**
    hyp $\leftarrow$ hyp $+k$
    $h_{curr}^{dec}, hidden \leftarrow F^{dec}(k, hidden = hidden)$
    $h_{t,curr}^{joint} \leftarrow F^{joint}(h_t^{enc}, h_{curr}^{dec})$
    $k \leftarrow \underset{k}{\text{argmax}}\,\text{Softmax}\left(h_{t,curr}^{joint}\right)$
  end while
end for

Graves (2012) propose l'utilisation de l'algorithme beam-search pour la prédiction de la séquence de sortie $\mathbf{y}$. Le calcul de la probabilité $P(y)$ se fait par l'extension de y par le label k au temps t, on peut donc décrire l'extension de y par son préfixe $\hat{y}$ avec $P(y|\hat{y}, t) = \prod_{u=|\hat{y}|+1}^{|y|} P(y_u|y_{0:u-1}, t)$.

Je propose à la section 6.3.1 une version parallélisée sur GPU de l'algorithme "Forward-Backward" pour accélérer l'apprentissage de nos modèles.

2.4 Modélisation séquentielle avec des réseaux de neurones

L'apprentissage profond (*Deep Learning*), domaine émergeant de l'apprentissage automatique, a pour objectif d'apprendre plusieurs niveaux de représentations des données. Ces représentations sont automatiquement découvertes et composées ensemble dans les différentes couches d'un réseau. Chaque niveau produit un ensemble de représentations abstraites découvert à partir des représentations issues du niveau précédent ; le niveau d'abstraction augmente avec chaque couche et permet d'extraire un sens. La modélisation neuronale est au coeur de l'apprentissage profond.

Plusieurs architectures sont offertes en *Deep Learning* pour le traitement des données séquentielles selon la nature des entrées-sorties et leur degré de séquentialité, comme l'illustre la figure 2.17 :

- Type 1 (one-to-one) : c'est la modélisation la plus répandue dans le domaine de l'apprentissage automatique supervisé : à partir de données non-séquentielles avec une entrée de taille fixe, le modèle prédit une sortie de taille fixe (exemple : reconnaissance d'objet à partir d'une image).
- Type 2 (one-to-many) : la modélisation de type réseau de neurones récurrent (RNN) permet de générer une séquence de sorties à partir d'une seule entrée (exemple : génération de légende-phrase à aprtir d'une image).
- Type 3 (many-to-one) : à l'inverse du type 2, cette modélisation prédit, à partir d'une séquence de données temporelles, une seule sortie (exemple : détection d'émotion à partir d'un signal de parole).

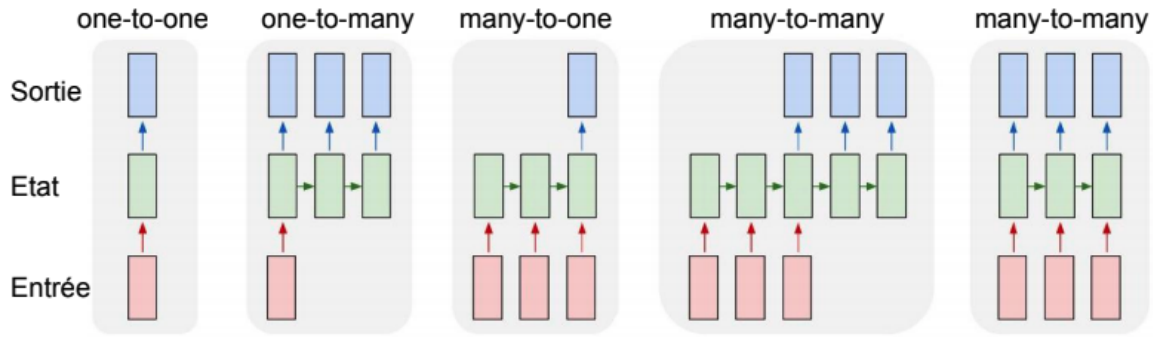


FIGURE 2.17 – Différents types de réseaux de neurones en fonction des entrées/sorties des données de taille fixe (one-to-one) et séquentielles (one-to-many, many-to-one, many-to-many).

— Type 4 (many-to-many) : cette architecture permet de lier deux séquences (entrée-sortie) de **longueurs différentes** : une séquence d'entrée de longueur N avec une séquence de sortie de longueur L. Le traitement utilise deux modèles récurrents : un encodeur pour la séquence d'entrée et un décodeur auto-régressif pour la génération de la séquence de sortie. Deux exemples applicatifs très étudiés sont la traduction automatique d'une phrase en français en une phrase en anglais), et la reconnaissance automatique de la parole par approche E2E *seg2seg*.
— Type 5 (many-to-many) : à la différence du Type 4, les deux séquences (entrée-sortie) sont de **même longueur**. Cette modélisation utilise un seul modèle récurrent appelé encodeur pour la prédiction de la séquence de sortie, par rapport au contexte séquentiel des données en entrée. La tâche de reconnaissance automatique de la parole avec une approche E2E CTC en est un exemple d'application.

Compte tenu de mon domaine d'études, la RAP, j'aborde, dans ma thèse, le traitement de données séquentielles par le type 5 pour l'approche RAP E2E CTC et RNN-T, et par le type 4 pour l'approche seq2seq encodeur-décodeur.

Ces approches exploitent de manière experte les réseaux de neurones récurrents ("Recurrent Neural Networks (RNN)") que nous présentons dans cette section ; nous détaillons leur apprentissage avec la rétro-propagation du gradient à travers le temps. Nous décrivons les problèmes de dépendances à long terme des séquences d'entrée/sortie liés à cette modélisation et nous présentons les deux variantes les plus répandues qui permettent de résoudre ce problème de dépendances : la cellule "Long-Short-Time-Memory (LSTM)" Hochreiter and Schmidhuber (1997) et la cellule "Gated Recurrent Unit (GRU)" Cho et al. (2014).

2.4.1 Modèles récurrents - RNN

Un réseau neuronal récurrent (RNN) est dit récurrent car il effectue le même traitement pour chaque élément d'une séquence d'entrée et la sortie au temps t dépend de tous les calculs précédents $(1...t-1)$. Cette caractéristique du RNN à capturer l'information séquentielle est aussi appelée "mémoire".

Un RNN est décrit à l'aide de trois transformations linéaires U, W et V. L'équation du RNN s'énonce comme suit :

$$h_t = \tanh(Ux_t + Wh_{t-1}) \tag{2.31}$$
$$y_t = f(Vh_t) \tag{2.32}$$

où $X = [x_1, x_2, ..., x_T]$ est la séquence d'entrée, $Y = [y_1, y_2, ..., y_T]$ la séquence de sortie, $H = [h_1, h_2, ..., h_T]$ l'état historique interne et f une fonction d'activation non linéaire. Le paramètre U représente la transfor-

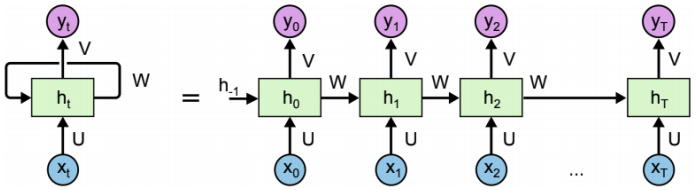


FIGURE 2.18 – Traitement par un réseau RNN d'une séquence d'entrée X produisant une séquence de sortie Y de même longueur.

mation linéaire liée à l'observation X à chaque temps t. le paramètre W représente la transformation linéaire liée à l'état historique précédent h_{t-1}. La transformation linéaire V produit la sortie du RNN en prenant en compte l'information contextuelle de h_t. La figure 2.18 illustre le **partage** des paramètres U, W, V à travers le temps.

Entraînement des RNNs

L'apprentissage des paramètres partagés U, W et V du modèle RNN résulte de la minimisation d'une fonction d'erreurs, E, représentant l'erreur globale entre la séquence de prédiction et la séquence de label de toutes les sorties. Cela implique le calcul des gradients de E par rapport aux paramètres U, W, V :

$$E = \sum_{t=0}^{T} E_t \tag{2.33}$$

$$\frac{\partial E}{\partial U} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial U} \tag{2.34}$$

$$\frac{\partial E}{\partial W} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial W} \tag{2.35}$$

$$\frac{\partial E}{\partial V} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial V} \tag{2.36}$$

où $\frac{\partial E}{\partial U}$, $\frac{\partial E}{\partial W}$, et $\frac{\partial E}{\partial V}$ représentent respectivement le gradient de l'erreur E par rapport à U, W, V.

L'algorithme nécessite une phase de propagation (*forward pass*) en se déplaçant de gauche à droite (sens temporel croissant), suivie par une phase de rétro-propagation, appelée en anglais "back-propagation through time (BPTT)", de droite à gauche, illustrées par la figure 2.19. La complexité du calcul est en $O(T)$ et ne peut être réduite par parallélisation du fait de la dépendance séquentielle du RNN.

Nous prenons comme exemple dans la figure 2.19 la rétro-propagation du gradient $\frac{\partial E}{\partial U}$ pour $T = 2$; le calcul se fait de la manière suivante :

— Le calcul de $\frac{\partial E_2}{\partial U}$ consiste à propager l'erreur jusqu'à l'état interne h_2, l'erreur peut en suite être propagée sur U au temps $t = 2$. U intervient également dans le calcul de h_1 et h_0, on rétro-propage l'erreur à travers le temps avec :

$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}(x_2^T + \frac{\partial h_2}{\partial h_1}(x_1^T + \frac{\partial h_1}{\partial h_0} x_0^T)) \tag{2.37}$$

— Le calcul de $\frac{\partial E_1}{\partial U}$ consiste à propager l'erreur jusqu'à l'état interne h_1, l'erreur est en suite propagée sur U au temps $t = 1$. Le paramètre U intervient également dans le calcul de h_0, on rétro-propage

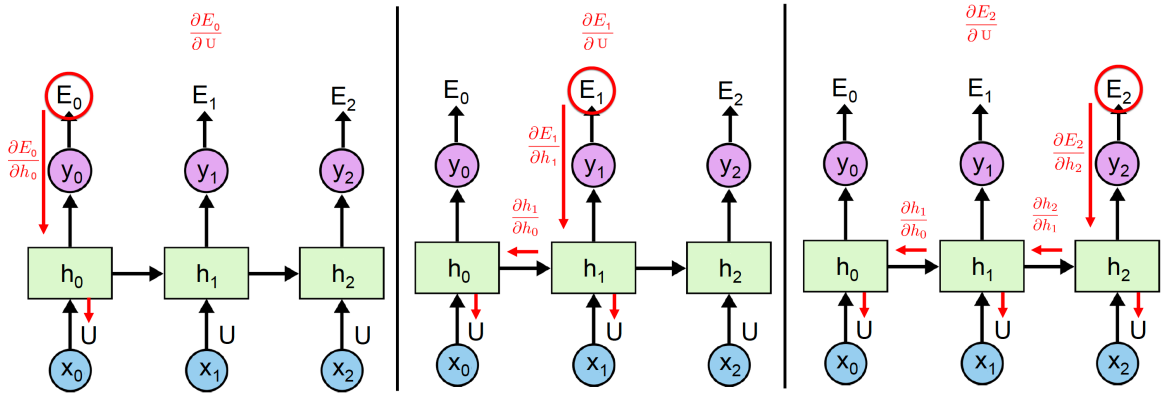


FIGURE 2.19 – Rétro-propagation du gradient pour le paramètre U partagé dans le temps.

l'erreur à travers le temps avec :

$$\frac{\partial E_1}{\partial U} = \frac{\partial E_1}{\partial h_1}(x_1^T + \frac{\partial h_1}{\partial h_0} x_0^T) \tag{2.38}$$

— Le calcul de $\frac{\partial E_0}{\partial U}$ consiste à propager l'erreur jusqu'à l'état interne h_1, l'erreur est en suite propagée sur U au temps $t = 0$ avec :

$$\frac{\partial E_0}{\partial U} = \frac{\partial E_0}{\partial h_0} x_0^T \tag{2.39}$$

Le gradient $\frac{\partial E}{\partial U}$ sur toute la séquence d'entrée est obtenu en sommant l'ensemble des gradients à chaque temps t comme décrit dans l'équation 2.33. Les gradients sur les autres paramètres W et V sont calculés de la même manière que U.

Problème de dépendance à long terme et rétro-propagation du gradient

La modélisation de données séquentielles à base de RNN doit permettre de mémoriser les dépendances à **court** et à **long** terme entre les séquences d'entrée et de sortie.

La figure 2.20 illustre la dépendance à long terme des RNN. Si la dépendance à court terme est satisfaite, représentée par les nuances en noire/gris montrant l'influence importante de l'entrée $t = 1$ sur les sorties pour t proche de 1, cette influence décroît au cours du temps et la dépendance à long terme peut devenir faible.

Il est clair que des difficultés d'apprentissage sur les RNNs surviennent notamment à cause de cette dépendance à long terme. Cela est, entre autres, du à la rétro-propagation du gradient à travers de nombreux pas de temps qui induit des instabilités comme le montre le calcul de $\frac{\partial y_t}{\partial x_0}$ qui s'écrit (figure 2.21) .

$$\frac{\partial y_t}{\partial x_0} = \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} ... \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial h_0} \frac{\partial h_0}{\partial x_0} \tag{2.40}$$

pour $t = 1, ..., T$.

Ce calcul peut conduire à deux problèmes

— Explosion du gradient : dans le cas où le gradient est trop grand à chaque temps t, le calcul du gradient est amplifié à chaque temps t et induit son explosion et la divergence des paramètres. Une solution simple à ce problème consiste à appliquer la technique du "gradient clipping" qui consiste à normaliser le gradient.

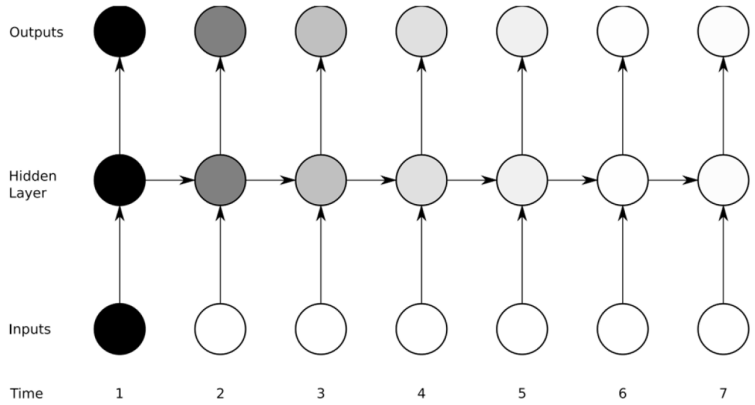


FIGURE 2.20 – Influence de l'entrée $t = 1$ du RNN sur la séquence sortie, quantifiée en niveaux de gris (noir = forte influence.

— Dissipation du gradient (vanish gradient) : dans le cas contraire où le gradient est trop petit (entre 0 et 1) à chaque temps t, la multiplication des gradients induit une diminution drastique jusqu'à la dissipation du gradient et empêche la mise à jour des paramètres. Aucune solution simple n'est possible pour résoudre ce problème.

Une variante : le RNN bidirectionnel

Le RNN bidirectionnel consiste à ajouter au RNN basic qui lit la séquence d'entrée de gauche à droite, un second RNN qui lit la séquence d'entrée à l'inverse de droite à gauche. Cette modélisation RNN bidirectionnelle permet de prendre en compte l'information contextuelle gauche et droite à chaque instant t. Les deux RNN ont des paramètres différents et les sorties sont sommées au niveau de la sortie y (figure 2.22).

2.4.2 Ajout du mécanisme de porte

Comme décrit à la section 2.4.1, l'explosion et la dissipation du gradient représentent des gros problèmes pour l'apprentissage des dépendances à long terme du RNNs. Plusieurs approches ont été explorées pour

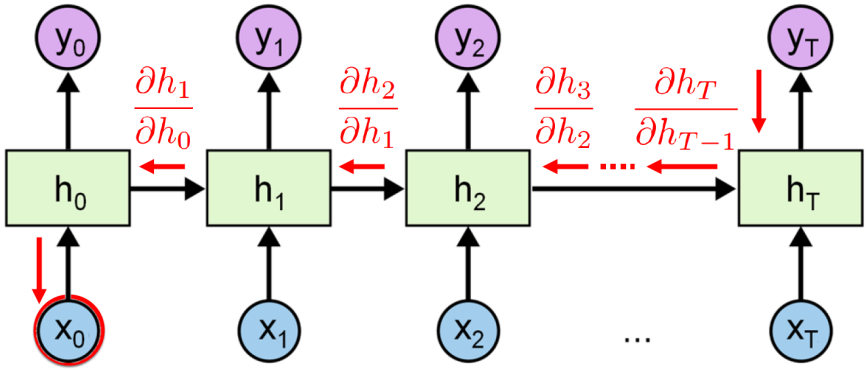


FIGURE 2.21 – Problème à long terme dans le calcul du gradient $\frac{\partial y_T}{\partial x_0}$. Problème 1 : dissipation du gradient, problème 2 : explosion du gradient.

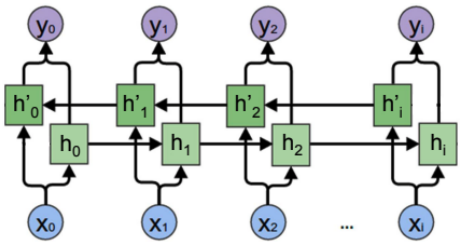


FIGURE 2.22 – RNN bidirectionnel

palier à ces problèmes notamment par Bengio et al. (1994) avec le "simulated annealing", le "discrete error backpropagation", le "time-weighted pseudo-Newton optimization" et l'"EM approach (Bengio and Frasconi (1994))". L'introduction de **portes (gates)** par Hochreiter and Schmidhuber (1997) a permis de proposer une solution efficace à ces problèmes. Le mécanisme d'ouverture de portes permet de filtrer de manière contrôlée l'information long terme.

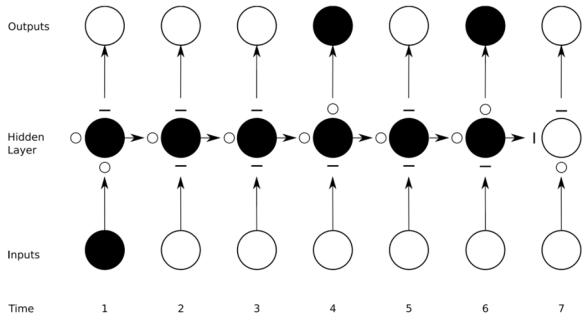


FIGURE 2.23 – Ajout de 3 gates (o ouvert ; - fermé) pour contrôler l'entrée, la sortie et l'effacement de l'état et propager plus ou moins l'information dans le temps.

La figure 2.23 illustre schématiquement la propagation de l'information de l'entrée $t = 1$ au fil du temps au travers d'un mécanisme de portes. Trois portes sont représentées dans la figure 2.23 : une à l'entrée, une la sortie et une à l'entrée de l'état caché ; les portes sont ouvertes ou fermées en fonction du symbole (o) ou (-) à l'instant t.

Long short-T memory (LSTM)

La cellule LSTM proposée par Hochreiter and Schmidhuber (1997) est un exemple de cellules utilisant le mécanisme de portes avec 3 portes **"Input gate (porte d'entée)", "Output gate (porte de sortie)", "Forget gate (porte d'oubli)" et une variable mémoire**.

Chaque porte de la cellule LSTM est activée avec une fonction sigmoïde (bornée entre 0 et 1), la variable mémoire c_t est considérée comme le pivot de la cellule LSTM et elle fait passer l'information temporelle en utilisant les 3 portes pour réduire le flux d'erreur ("Error flow") partagé dans le temps. Le calcul au sein de

la cellule LSTM se fait de la manière suivante (figure 2.24) :

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{2.41}$$
$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \tag{2.42}$$
$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \tag{2.43}$$
$$g_t = \tanh(U_g x_t + W_g h_{t-1} + b_g) \tag{2.44}$$
$$c_t = i_t \odot g_t + f_t \odot c_{t-1} \tag{2.45}$$
$$h_t = o_t \odot \tanh(c_t) \tag{2.46}$$

où, à chaque instant t, :

- x_t représente l'entrée du LSTM
- i_t représente l'activation de la porte d'entrée et permet de contrôler le rajout de l'information à la cellule.
- f_t représente l'activation de la porte d'oubli et permet de contrôler la dissipation de l'information à l'intérieur de la cellule.
- o_t représente l'activation de la porte de sortie et permet de pondérer ce qui sort de la cellule (cf equation 1.46).
- g_t appelée "cell candidate" représente l'information à rajouter à la variable mémoire (cf equation 1.45).
- c_t correspond à la mise à jour de la variable mémoire à l'aide de la pondération de la porte d'entrée i_t et la porte d'oubli f_t sur les paramètres g_t (le rajout d'information) et c_{t-1} (l'historique de la variable mémoire au temps $t-1$).
- h_t représente le vecteur de sortie de la cellule LSTM pondérée par la porte de sortie o_t.

Les paramètres partagés U, W et le vecteur biais b sont estimés sur les données d'apprentissage.

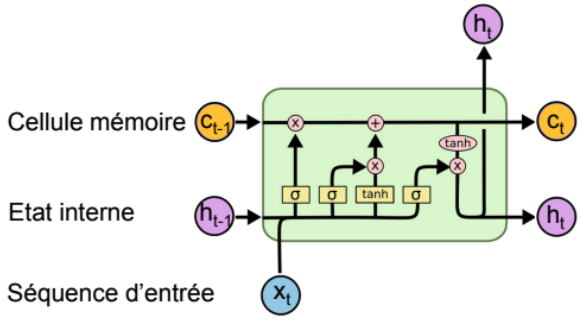


FIGURE 2.24 – Cellule Long short-T memory (LSTM)

Gated Reccurent Unit - GRU

La cellule GRU introduit par Cho et al. (2014) en 2014 est un autre exemple de cellules utilisant le mécanisme de portes, similaire au LSTM : la propriété introduite par les LSTM sur la porte d'oubli "Forget gate" est maintenue ; cependant, la porte de sortie "Output gate" et la variable "mémoire" ont été enlevées. Le GRU gère la modélisation séquentielle à l'aide de deux portes appelées la porte de mise à jour "Update gate" et la porte de remise à zéro "Reset gate", selon le schéma suivant :

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z) \tag{2.47}$$
$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r) \tag{2.48}$$
$$g_t = \tanh(U_g x_t + W_g(r_t h_{t-1}) + b_g) \tag{2.49}$$
$$h_t = z_t \odot g_t + (1 - z_t) \odot h_{t-1} \tag{2.50}$$

où , à chaque instant t :
- x_t représente l'entrée du GRU
- z_t représente l'activation de la porte de mise à jour
- r_t représente l'activation de la porte de remise à zéro pour l'historique de l'état interne h).
- g_t propage l'information
- h_t représente la sortie calculée par pondération de g_t et de l'état précédant h_{t-1}

Les paramètres partagés U, W et le vecteur biais b sont estimés sur les données d'apprentissage. La figure 2.25 illustre la révision du LSTM effectué par Cho et al. (2014) pour obtenir une modélisation plus compact (à 2 portes).

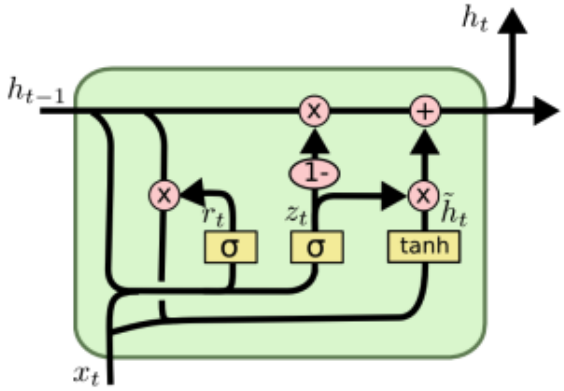


FIGURE 2.25 – Gated Reccurent Unit - GRU

La modélisation GRU permet d'offrir une cellule plus compact et donc plus rapide dans l'apprentissage et l'inférence que les LSTM. Cependant, certains recherches (Weiss et al., 2018; Su and Kuo, 2019) affirment que les performances des LSTM sont strictement supérieures à celles du GRU pour la modélisation du langage. Nous verrons ci-après une confirmation de cette position dans le cadre de notre travail de recherche sur le choix des modélisations RNN pour la RAP E2E (cf Tableau 6.4).

Light Gated Reccurent Unit - Li-GRU

Une évolution de la cellule GRU adaptée au traitement du signal a été proposée par Ravanelli et al. (2018) et permet de surpasser les performances du GRU et du LSTM sur la tâche de la RAP. La modélisation Li-GRU propose deux modifications du GRU : 1) réduire la modélisation à 2 portes du GRU en enlevant la porte de remise à zéro "Reset gate" et en ne laissant qu'une seule porte "Update gate" avec l'introduction d'une normalisation par batch sur cette porte. 2) utiliser la fonction d'activation "ReLU" à la place de la tangente hyperbolique "Tanh". Les équations deviennent :

$$z_t = \sigma(\mathrm{BN}(U_z x_t) + W_z h_{t-1} + b_z) \tag{2.51}$$
$$g_t = \mathrm{ReLU}(U_g x_t + W_g(r_t h_{t-1}) + b_g) \tag{2.52}$$
$$h_t = z_t \odot g_t + (1 - z_t) \odot h_{t-1} \tag{2.53}$$

où , à chaque instant t :
- x_t représente l'entrée,
- $BN(.)$ est une fonction de normalisation par batch,
- z_t est l'activation de la porte de mise à jour,
- g_t est l'information accumulée dans l'état interne

— h_t est la sortie de la cellule Li-GRU pondéré par l'activation de la porte de mise à jour z_t et son inverse $(1 - z_t)$.

2.5 Conclusion

Dans ce chapitre, en effectuant l'historique du domaine de la RAP, j'ai pu avoir une vue générale de l'évolution des approches et modélisations utilisées dans ce domaine de la RAP, et j'en ai été impressionné ; ma participation à InterSpeech 2019 Graz m'a permis de l'être un peu plus. La deuxième partie a été consacrée aux approches hybrides à base de HMM ; ces systèmes ont fait leurs preuves au cours des 40 dernières années. Un de leurs grands avantages est de pouvoir faire bénéficier des progrès réalisés dans le domaine du Deep Learning pour la modélisation acoustique et de profiter des approches statistiques en recherche et exploration dans les graphes pour la génération d'hypothèses.

Le courant de recherche émergeant durant la dernière décennie estime qu'il faut favoriser les systèmes "End-To-End" basées entièrement sur l'apprentissage et que les approches hybrides avec des algorithmes de recherche dans des graphes peuvent être considérées comme des algorithmes de "force brute" n'ayant aucune flexibilité dans l'exploration des hypothèses. Le slogan “We believe more in Learning than in Searching” peut être attribué à ce courant de recherche que j'expose dans la troisième partie. Les systèmes RAP E2E ont évolué dans un premier temps vers la prédiction de graphèmes ; ce challenge a permis d'introduire la fonction de coût CTC en proposant, avec l'introduction de label *blank*, un alignement monotone entre l'entrée et la sortie résolvant ainsi le problème de reconnaissance de deux séquences de longueur différente.

Nous présentons par la suite une évolution de l'approche CTC pour prendre en compte la dépendance entre les prédictions de la séquence de sortie. Le RNN-T avec ces trois composants, d'encodage de l'observation acoustique, de prise en compte de la séquence de sortie précédente par le décodeur, et de fusionner les deux informations acoustique et linguistique, offre un système adapté au domaine de la RAP. Nous proposons par la suite d'approfondir ce type d'architecture.

Chapitre 3

Ressources pour la reconnaissance automatique de la parole

3.1 Introduction

La collecte, la création et la préparation de corpus d'apprentissage est un processus important dans la phase de développement des systèmes RAP. Ce processus a pour objectif de structurer les données audio/transcription, de telle manière à faciliter l'étape d'apprentissage en général fortement supervisé des modèles RAP.

Ces modèles RAP impliquant la technologie neuronale sont capables d'apprendre à partir d'un large ensemble de donnée. La question qui se pose alors est : combien d'heures faut-il et quels types de données audio ou textuelles sont nécessaires pour réaliser ces apprentissages et donc apprendre un langage ?.

Apprendre une langue n'est pas une tache facile, même pour les humains. D'après les recherches de Moore (2003), la moyenne du nombre d'heures de parole auxquelles l'humain est exposé tout au long de sa vie, est approximativement de :

— 2 ans : dans les 1000 heures (6M de mots par an)
— 10 ans : dans les 10 000 heures (14M de mots par an)
— 50 ans : dans les 100 000 heures (14M de mots par an)
— 80 ans : dans les 150 000 heures (14M de mots par an)

Ces statistiques nous informent sur la quantité d'heures perçues par l'humain tout au long de sa vie. Une étude plus approfondie de Lippmann (1997) sur la performance de l'humain pour la tache de reconnaissance de parole a été proposé dans les années 2000. La performance de l'humain sur tout type de parole est évaluée au alentour de 5.6% de WER. L'adaptation de l'humain à des conditions et des environnements bruités est très évoluée.

Récemment, il a été estimé qu'il fallait environ 300 heures pour l'apprentissage d'un système RAP E2E et son étude de faisabilité (en anglais : "Proof of Concept"), et plus de 3000 heures de parole pour permettre de l'industrialiser. Actuellement la mise en place de systèmes RAP industrialisés implique un volume de données compris entre 10 000 et 100 000 heures de parole pour leur apprentissage, comme par exemple les systèmes RAP (Hannun et al., 2014; Amodei et al., 2016; Chan et al., 2016).

Idéalement, pour construire un système RAP large vocabulaire (en anglais : "Large Vocabulary Speech Recognition (LVCSR)"), lors de la collecte de données d'apprentissage, il est préférable de sélectionner des données enregistrées dans des conditions les plus proches possible du domaine d'application cible (au niveau acoustique et langagier). Les principaux points concernant la sélection des corpus, à prendre en considération, sont les suivants :

— l'enregistrement de plusieurs locuteurs (homme, femme) de différents âges.
— l'utilisation d'un large éventail de microphones différents, dans des environnements variés.

— une grande couverture des accents de la langue cible.
— la qualité d'annotation mesurée au travers d'une marge d'erreur des annotations minime.

Ces critères de sélection sont particulièrement sensibles pour les supports gratuits (en anglais : "Open Source").

Dans ce chapitre, nous détaillons les corpus traités, oraux et textuels, en français et en anglais, pour l'apprentissage de système RAP, ainsi qu'une description des étapes de pré traitement nécessaires pour les rendre utilisables.

3.2 Corpus de parole

Construire un système RAP à partir de donnée non commerciale est une priorité, voire un a priori, pour la société Linagora. C'est pourquoi, durant cette thèse et à des fins de comparaison, j'utilise à la fois des corpus de références académiques pour le français et l'anglais et j'applique mes développements sur les données libres.

3.2.1 Corpus de parole en langue française

Le corpus BREF-80 est un sous-corpus de 10 heures issues du corpus BREF-120 [1]. Celui-ci a été réalisé par le LIMSI-CNRS (Lamel et al., 1991) et conçu pour fournir un large corpus de textes lus pour le français. Les textes sont issus du journal LE MONDE et permettent d'atteindre un vocabulaire de plus de 20000 mots (Gauvain et al., 1990). BREF-120 contient 100 heures de parole enregistrées par 120 locuteurs (65 femmes et 55 hommes) avec environ 650 phrases par locuteur. Les phrases du corpus BREF-80 sont prononcées par 80 locuteurs, à raison de 65 par locuteur. Elles ont été choisies pour couvrir le plus d'énoncés possibles.

Le Corpus ESTER 1 a été réalisé dans le cadre de la première compagne d'évaluation des systèmes de transcription enrichie d'émissions radiophoniques (ESTER) du projet EVALDA [2]. Cette campagne d'enregistrement et d'annotation de support radiophonique en français a commencé en 2003, avec l'enregistrement de 100 heures de parole composé de 37 heures de la radio France Inter (entre 1998-1999 et 2003), de 12 heures de France Info (2003-2004), de 27 heures de la radio RFI (entre 2000 et 2003, 2004), de 25 heures de la radio RTM ainsi qu'une heure de la radio France culture et une heure de la radio Classique. Des méta données telles que les tours de parole, l'identification du locuteur, le suivi des segments de musique/parole, l'annotation du thème et des sujets de parole sont adjointes. Durant la même période, une collecte de 1600 heures d'enregistrement de radio, non annotées, est associée a cette campagne d'évaluation.

Le Corpus ESTER 2 a été réalisé dans le cadre de la deuxième campagne d'évaluation "ESTER" du projet EVALDA entre 2008-2009. Cette campagne d'enregistrement et d'annotation vise à récolter des supports radiophoniques en langue française à partir de radios africaines avec environ 100 heures de parole annotées. En plus des méta-données proposées dans ESTER 1, ESTER2 propose une annotation des entités nommées. Les émissions enregistrées contiennent des émissions d'information et d'actualité internationales. (Galliano et al., 2009)

Le Corpus EPAC contient environ 100 heures de transcriptions de parole conversationnelle issue d'émissions radiophoniques où l'aspect spontané des discussions est beaucoup plus présent que lors des émissions utilisées dans les corpus ESTER 1 & 2. Le corpus contient 30 heures d'enregistrement de la radio France Inter, 40 heures de France Culture et 25 heures de RFI. Les émissions principalement utilisées pour EPAC sont : "Le téléphone sonne", "Quartiers d'Été", "Sous les étoiles exactement", "Culture vive" et "Les matins" (Esteve et al., 2010).

1. BREF-120 : `https://catalogue.elra.info/en-us/repository/browse/ELRA-S0067`
2. EVALDA : http ://www.elda.org/en/projects/archived-projects/evalda/

Le corpus TCOF "Traitement de Corpus Oraux en Français" est né de la volonté de conserver des corpus oraux collectés dans les années 80-90 à des fins de recherches. Le corpus mis à disposition comporte deux grandes catégories : des enregistrements d'interaction adulte-enfant (enfants jusque 7 ans) et des enregistrements d'interaction entre adultes. Les enregistrements sont de durées diverses, de 5 à 45 minutes au plus. Le corpus TCOF contient près de 517 séances d'enregistrement pour une durée totale de 124 heures et 675 locuteurs. Le corpus est disponible en télechargement sur ORTOLANG.

Le Corpus ASCYNT est un corpus de parole en langue française qui contient 124 000 mots. Il est composé de trois cas d'usages possibles : la lecture de texte (1 locuteur), des présentations monologuées (2 intervenants) et des entretiens guidés (2 intervenants). Les données ont été enregistrées auprès de 23 locuteurs originaires du Sud Ouest de la France pour un total de 4 heures (DELAIS-ROUSSARIE et al., 2013).

Le Corpus Mozilla Common Voice pour le français est un corpus libre de droit contenant des enregistrements de volontaires au travers de la plate-forme "Common Voice" [3]. Cette plate-forme proposée par Mozilla, consiste à donner l'accès à des contributeurs afin d'enrichir les supports audio dans leur langue et aider au développement des systèmes RAP pour toutes les langues du monde. Sur cette plate-forme, il est proposé des phrases à des volontaires pour récolter leurs prononciations et enrichir la collecte de données. Une phase de validation des enregistrements est proposée sur cette plate-forme pour confirmer la paire audio/transcription. En février 2020, on pouvait compter plus de 350 heures d'enregistrement pour la langue française. Des méta données telles que l'âge, le sexe, la localisation (information sur le type de dialecte et accent) sont disponibles pour pouvoir réaliser une panoplie d'études sur les systèmes RAP (Ardila et al., 2019).

Le Corpus Librivox Français est un corpus contenant 140 heures d'enregistrement sélectionnées à partir de livres français lus par des locuteurs (natifs ou pas) dans le but d'ouvrir l'accès de ces ouvrages aux personnes malvoyantes. Le projet Librivox [4] est un projet libre de droit qui contient des livres appartenant au domaine publique. Les enregistrements sélectionnés sont segmentés, puis alignés automatiquement. Ce corpus contient toutes les méta-informations nécessaires pour la construction de systèmes RAP, à savoir l'identification du locuteur par segment, les limites de chaque segment ainsi que l'alignement textuel associé.

Corpus Corpus Corpus	Type de parole	Nombre de phrases	Taille du vocabulaire	Durée moyenne	Nombre d'heures utilisées	Total d'heures
Mozilla FR	Lu	15K	14K	2.5s	_	350H
BREF-80		15K	14K	2.5s	10H	10H
Librivox		16K	28K	11.1s	40H	140H
TCOF		15K	14K	2.5s	_	100H
ACSYNT		15K	14K	2.5s	_	4H
ESTER 1	Spontané	93K	38K	3.4s	100H	100H
ESTER 2		97K	42K	3.9s	90H	100H
EPAC		94K	35K	3.4s	100H	100H
Corpus Combiné	L & S		81K		340H	900H

TABLE 3.1 – Liste des corpus collectés et utilisés pour la langue française

Le tableau 3.1 illustre la liste des corpus français collectés et utilisés durant ma thèse. Pour la majorité des corpus, est proposée une partition en ensemble d'apprentissage, de développement et d'évaluation.

3. Common Voice : `https://voice.mozilla.org`
4. LibriVox : `https://librivox.org`

3.2.2 Corpus de parole en langue anglaise

Le corpus TIMIT est des premiers corpus conçu pour fournir, à la communauté scientifique, des données vocales pour des études acoustiques-phonétiques et pour le développement et l'évaluation de systèmes de reconnaissance automatique de la parole (1988). Il a été progressivement enrichi ; il contient une sélection d'enregistrements de parole lue de 630 locuteurs de 8 dialectes majeurs de l'anglais américain, chacun lisant dix phrases phonétiquement riches. Le corpus TIMIT comprend des annotations orthographiques, phonétiques et l'alignement temporel des mots prononcés sur un ensemble de 4 heures d'enregistrement.

Le corpus Librispeech contient une sélection d'enregistrement de lecture de livres, issue du projet Librivox [5]. Comme dit précédemment, ce projet met à disposition des malvoyants un ensemble de 15000 livres audio à partir de livres libres de droit. Le texte de chaque livre est lu par des volontaires (natifs ou pas). Le signal de parole a été segmenté et aligné avec le texte en utilisant des techniques d'alignement automatique. Au final, ce corpus est composé de 1000 heures d'enregistrement avec leurs références textuelles. A noter, un certain nombre de livres sont anciens, du fait des droits d'auteurs, et le vocabulaire peut être très ancien et des styles de prononciation narrative sont adjointes pour aider à la compréhension de l'ouvrage, par exemple dans un livre de Shakespeare (Panayotov et al., 2015).

Le corpus Wall Street Journal Cambridge (WSJCAM0) est un corpus de parole [6] en anglais britannique réalisé par l'université de Cambridge. Il est issu de la lecture en anglais américain du journal "Wall Street Journal (WSJ0)" (Paul and Baker, 1992). Il a été spécifiquement conçu pour la construction de système RAP indépendant du locuteur et compte environ 80 heures de lecture par 140 locuteurs, à raison de 110 phrases chacun (Robinson et al., 1995).

Le corpus Mozilla Common Voice est issu du projet "Common Voice" de Mozilla décrit au paragraphe 3.2.1. En février 2020, plus de 1118 heures d'enregistrement étaient transcrites pour la langue anglaise réalisées par 51072 donateurs de différents continents. Les méta données ciblées sont identiques en français et en anglais (Ardila et al., 2019).

Corpus Corpus Corpus	Type de parole	Nombre de phrases	Taille du vocabulaire	Durée moyenne	Nombre d'heures utilisées	Total d'heures
TIMIT	Lu	4,7K	53K	7.9s	4H	4H
WSJCAM0	Lu	4,7K	53K	7.9s	84H	84H
Librispeech	Lu	292K	382K	12.3s	1000H	1000H
Mozilla EN	Lu	292K	382K	12.3s	_	1118H

TABLE 3.2 – La liste des corpus anglais collectés et utilisés durant ma thèse.

Comme pour le français, le tableau 3.2 précise la liste des corpus anglais utilisés durant ma thèse.

3.3 Corpus textuel pour le français

Pour construire un modèle linguistique statistique, il est indispensable de disposer d'une vaste collection de textes propres. Il est possible de collecter la transcription de textes issus de projets comme Librivox, les podcasts transcrits, mettre en place une collecte de données sur le web, en transcrivant des enregistrements existants ou en les générant artificiellement avec des scripts. voire essayer de contribuer à Voxforge.

5. LibriVox : `https://librivox.org`
6. WSJCAM0 : https ://catalog.ldc.upenn.edu/docs/LDC95S24/wsjcam0.html

Le contexte de ma thèse concerne les systèmes RAP en français, la parole étant captée lors de réunions professionnelles. Nous avons donc essayé de collecter des corpus aussi proches que possible de ce contexte, que nous avons du rendre propres.

Le corpus du journal Le Monde (1987-2003) est un corpus d'archivage des articles du journal français "Le Monde" de 1987 à 2003. 200 articles sont archivés par jour et cet ensemble constitue la plus grosse base de données en texte intégral de la presse française. Ce corpus contient près de 300 millions de mots et correspond à un vocabulaire 900k mots.

Le corpus FRWac est un corpus [7] de textes écrits en français et collectés sur internet (issus d'un grand nombre de sites dont le nom de domaine est de type *.fr). Une recherche par mots clés issus du journal "Le Monde Diplomatique" a permis de sélectionner les sites web en question. Cette collection de données textuelles compte environ 1,3 milliards de mots, pour 55 millions de phrases. (Ferraresi et al., 2010)

Le corpus OpenSubtitle est une collection de données "open-source" de sous titrages multi-langues de films. Nous nous basons sur la version 2018 qui contient des sous-titrages en 62 langues dont la langue française avec plus de 3,7 millions de films.

Le corpus Wikipedia Français est constitué à partir du dump [8] de la version française de l'encyclopédie Wikipedia du 02/04/2019. Ce *dump* contient un pré-traitement minimal de découpe des articles wikipedia en sous parties textuelles (sommaire, sections, etc...). Ce corpus contient une collection de données textuelles d'environ 418 Millions de mots avec un vocabulaire de 1,5 millions (à noter la présence de beaucoup de noms propres et de mots en langue étrangère).

Le corpus Gigaword Français est une archive complète de données textuelles [9] de l'actualité de presse française collectée par le Consortium des Données du Langage (en anglais "Linguistic Data Consortium (LDC)" à partir de supports de "L'agence France-Presse 94-2010", ainsi que de "The Associated Press French Service (apw FR) 94-2010". Ce corpus contient près de 700 millions de mots avec un vocabulaire avoisinant les 2,5 millions.

Le corpus Est Républicain est une collection de données textuelles [10] extrait du journal l'Est Républicain publié entre 1999 et 2003. Cette collection a été réalisée par le Laboratoires d'Analyse et Traitement Informatique de la Langue Française (ALTIF, Nancy, France), ainsi que le laboratoire CLLE-ERSS - Cognition, Langue, Langages, Ergonomie de l'université Toulouse II - Jean Jaures. Ce corpus contient environ 126 millions de mots avec un vocabulaire de 474K mots(Gaiffe and Nehbi, 2009).

Le corpus Parlement Européen est constitué des retranscriptions des échanges au sein du parlement européen entre 1996-2003. chaque intervention est traduite en onze différentes langues dont le français. Nous avons extrait uniquement le texte français associé et constitué un sous corpus contenant une collection de données textuelles d'environ 55 Millions de mots, avec un vocabulaire de 124K mots(Koehn, 2005).

3.4 Processus de pré-traitement

La préparation méticuleuse des données est essentielle pour une bonne mise en oeuvre d'un système RAP, quel qu'il soit. Dans cette section, nous détaillons les techniques et normalisations que nous utilisons systématiquement afin de disposer d'un standard commun à tous les corpus que nous traitons (parole et texte).

7. FRWac : `https://corpora.dipintra.it/public/run.cgi/corp_info?corpname=frwac_full`
8. Corpus Wikipedia : `https://dumps.wikimedia.org/`
9. Gigaword Français : `https://catalog.ldc.upenn.edu/LDC2011T10`
10. Corpus Est Républicain : `https://www.ortolang.fr/market/corpora/est_republicain`

3.4.1 Pré-traitement de l'audio

Le pré-traitement des enregistrements audio pour la phase d'apprentissage et de décodage consiste à :
- convertir les fichiers audio en format WAV.
- modifier l'échantillonnage des audios en 16kHz, si nécessaire.
- convertir les enregistrements stéréo en mono.

De plus, les corpus audio définis dans la Section 3.2.1 sont annotés en utilisant le logiciel "Transcriber" qui consiste à faciliter l'annotation avec une interface graphique et un formalisme XML. Des balises tel que "Turn" pour identifier un ensemble de phrases du même locuteur, "Sync" pour identifier une prononciation et une balise "Section" pour identifier un sujet de discussion entre locuteurs, sont proposées. D'autres balises pour l'identification des bruits, des entités nommées, des passages de musique/téléphone sont aussi rajoutées comme méta données pour certains corpus.

Nous avons mis en place une procédure d'extraction d'information pour convertir ces annotations en entités exploitables telles que :
- Une entité "Texte" contenant l'identifiant de la prononciation et le texte associé.
- Une entité "Utt2spk" contenant pour chaque identifiant de prononciation, le locuteur associé.
- Une entité "Spk2utt" contenant pour chaque locuteur, tous les prononciations associées.
- Une entité "Segments" contenant le lien entre chaque identifiant de prononciation et l'identifiant de l'enregistrement associé avec les temps de début et fin.
- Une entité "spk2gender" contenant l'information du sexe de chaque locuteur.
- Une entité "wav" contenant l'identifiant de chaque enregistrement avec le fonction de lecture du fichier à partir de son chemin.

La normalisation de ces informations les rend exploitables lors de l'apprentissage des modèles acoustiques.

3.4.2 Pré-traitement des données textuelles

Compte tenu de l'important volume de données, quatre milliards de mots, en entrée des modules d'apprentissage des modèles de langage, il fallait trouver un moyen pour convertir les différents formats brut (XML, JSON, Texte brut, etc...) de chaque corpus en un unique format. Nous avons proposé d'utiliser un format spécial, dit **pivot** qui est très utilisé dans le domaine du TAL dans le cadre de l'étiquetage morpho-syntaxique (en anglais : "Part-Of-Speech Tagging (POS)"). Ce format consiste à associer à chaque mot, les informations suivantes (voir exemple Figure 3.1) :

1. Identifiant.
2. Lemmatisation (information grammaticale associée).
3. Booléen de détection de ponctuation.
4. Booléen de détection d'espace.
5. Majuscule/Minuscule.
6. POS.
7. les informations détaillées du POS.

À partir de ce **format pivot**, je propose d'appliquer la normalisation suivante :
- Élimination de toutes les ponctuations (à chaque point, on effectue un retour à la ligne).
- Conservation de toutes les hésitations.
- Conversion des chiffres en mots.
- Écriture de tous les mots en minuscules.
- Conversion des caractères spéciaux (€ en "euro", % en pourcentage, @ en "arobase" etc..).

3.5 Conclusion

Nous avons présenté dans cette section les différents corpus utilisés durant notre thèse. Un très grand travail de normalisation des corpus a été réalisé pour arriver à atteindre 900H de corpus de parole en français

```
Demain  0       Demain  False   False   Xxxxx   PROPN   PROPN___
je      7       il      False   False   xx      PRON    PRON__Number=Sing|Person=1
travaille       10      travailler      False   False   xxxx    VERB    VERB__Mood=Ind|Number=Sing|Person=1|Tense=Pre
s|VerbForm=Fin
à       20      à       False   False   x       ADP     ADP___
la      22      le      False   False   xx      DET     DET__Definite=Def|Gender=Fem|Number=Sing|PronType=Art
maison  25      maison  False   False   xxxx    NOUN    NOUN__Gender=Fem|Number=Sing
.       31      .       True    False   .       PUNCT   PUNCT___
```

FIGURE 3.1 – Découpe de la phrase “Demain je travaille à la maison” avec la technique de Tokenisation de SpaCy.

et près de 4 milliards de mots en corpus textuels. Cet ensemble constitue un très bon ensemble pour réaliser l’apprentissage de système RAP à large vocabulaire en français.

Chapitre 4

Approche hybride large vocabulaire pour le français

Introduction

Dans ce chapitre, nous reprenons l'étude d'un système RAP, grand vocabulaire, pour la langue française, à base de modélisation hybride DNN-HMM afin d'approfondir les choix faits pour les modélisations acoustique, lexical et linguistique. D'un point de vue logiciel, j'exploite l'outil open-source Kaldi Povey et al. (2011).

Sur le plan acoustique, j'ai cherché à identifier les composants neuronaux les plus discriminants en usant d'une démarche de type boostrap : à chaque étape, un modèle acoustique plus complexe est proposé et l'alignement des données d'apprentissage, est revu assurant un meilleur apprentissage. J'ai exploré des architectures TDNN et TDNN-Factorized proposées par Povey et al. (2018a) et j'ai intégré des techniques de self-attention proposées par Povey et al. (2018b).

Sur le plan lexical et linguistique, j'ai produit un modèle de langage pour la parole spontanée en français avec notre corpus de 4 milliards de mots après sélection d'un large dictionnaire pour la langue française (réalisée par mon stagiaire à Linagora).

Au final, j'ai évalué les les techniques d'augmentations de données pour statuer sur les plus intéressantes et conclure sur un système LVCSR hybride en langue française performant.

4.1 Du GMM au TDNN : une modélisation acoustique progressive

Comme annoncé ci dessus, l'apprentissage du modèle acoustique se fait avec une technique de boostrap pour complexifier la modélisation acoustique et pour estimer les paramètres sur des données aussi correctement alignées que possible : chaque étape correspond à un type de modélisation acoustique et le système appris est réutilisé pour générer des alignements des données d'apprentissage, plus précis ; ces nouveaux alignements servent pour l'apprentissage du système suivant. Pour disposer d'un premier alignement des données d'apprentissage en entrée de notre système hybride DNN-HMM pour le français, j'ai développé plusieurs systèmes classiques GMM-HMM en mettant également en oeuvre un apprentissage par bootstrap.

Au cours de chaque partie de cette section, sont décrites les principales étapes de cette modélisation acoustique .

4.1.1 Systèmes GMM-HMM

Les systèmes GMM-HMM proposés prennent en entrée des observations de type MFCC : de base, 13 MFCC sont extraits de 25 filtres mels auxquels sont ajoutées les dérivée première et seconde. L'approche bootstrap appliquée sur ces systèmes est constituée de quatre étapes correspond à quatre types de modèles acoustiques, graduellement plus complexes.

Système GMM-HMM-monophone

Le premier système est basé sur des modèles acoustiques "mono phone" : 1000 lois gaussiennes sont apprises avec un réalignement des données et une re-estimation des paramètres à chaque itération durant les 10 premières itérations, puis un re-alignement toutes les deux itérations jusqu'à atteindre 40 itérations. On utilise pour cet apprentissage, 5% de notre quantité totale de données, soit 17 heures de données d'apprentissage.

Un alignement de tout l'ensemble d'apprentissage (340 heures) par ce premier système est réalisé pour son utilisation à la prochaine étape d'apprentissage.

Système GMM-HMM-Triphone

Le deuxième système est basé sur des modèles acoustiques "tri-phone" (HMM de 3 états) : 40k lois gaussiennes sont apprises sur toutes les données avec une augmentation graduelle du nombre de lois : le re-alignement des données se fait toutes les 10 itérations jusqu'à atteindre le nombre maximal d'itérations (30 dans notre cas).

Un alignement avec ce deuxième système GMM-HMM-Triphone est réalisé sur toutes les données d'apprentissage.

Système GMM-HMM-Tri-MLLT

Le troisième système est également basé sur des modèles acoustiques "tri-phone" mais la prise en compte du contexte acoustique est élargi : 7 fenêtres de signal sont utilisées pour obtenir un total de 273 paramètres (13 * 3 (observation, dérivé 1er et 2nd) * 7 (nombre de fenêtre)). Une LDA est appliquée sur ce vecteur acoustique pour réduire sa dimension à 40 paramètres. Une matrice de transformation acoustique (MLLT en anglais “Maximum Likelihood Linear Transform (MLLT) estimation“) est appliquée sur la sortie LDA pour améliorer l'indépendance aux locuteurs. Ce système est appris après 30 itérations et le re-alignement des données est réalisé comme à l'étape précédente toutes les 10 itérations.

Un alignement avec ce troisième système GMM-HMM-Tri-MLLT est réalisé sur toutes les données d'apprentissage.

Système GMM-HMM-Tri-SAT

Un quatrième système GMM-HMM, qui sera noté GMM-HMM-Tri-SAT, est appris avec la technique “Speaker Adaptative Training (SAT)” : l'adaptation des paramètres des lois par rapport aux locuteurs est apprise. 30 itérations sur l'ensemble d'apprentissage sont réalisées pour l'estimation du modèle avec un re-alignement des données toutes les 10 itérations.

Les données d'apprentissage utilisées pour aborder la réalisation des systèmes hybrides DNN-HMM sont alignées avec ce dernier système GMM-HMM-Tri-SAT incluant des modèles triphones, le contexte acoustique élargi et l'adaptation des lois au locuteur. Au delà du meilleur alignement, nous générons le graphe d'hypothèses (*Lattice*) afin de rendre plus flexible les apprentissages des systèmes DNN-HMM, comme nous le verrons ci-après.

4.1.2 Systèmes DNN-HMM

Les systèmes hybrides DNN-HMM que je propose pour la langue française incluent une modélisation acoustique guidée par l'architecture TDNN qui permet une meilleure prise en compte du contexte acoustique (voir section 2.2.1). Comme pour l'approche GMM-HMM, les systèmes prennent en entrée des observations issues d'un calcul des MFCC. Alors que dans l'approche GMM-HMM, les performances ne dépendent pas du nombre de MFCC pris de base, il n'en est pas de même pour l'approche DNN-HMM ; j'ai donc privilégié la paramétrisation la plus performante, à savoir 40 coefficients MFCC. À chaque instant, un vecteur de dimension 840 est obtenu en prenant en compte les dérivées premières et secondes ainsi qu'un contexte acoustique élargi à 7 fenêtres d'analyse (4°*3*7). Une LDA permet de réduire cette dimension à 40. Pour prendre en compte l'adaptation au locuteur, l'approche i-vecteur est privilégié pour fournir à chaque réseau de neurones étudié, un vecteur de dimension 140 (LDA-40, i-vecteur-100).

Trois modélisations DNN-HMM sont proposées pour la mise en place de systèmes LVCSR en français.

Système DNN-HMM-TDNN

La modélisation acoustique de ce premier modèle est construit avec une topologie de type "chain model" et un réseau neuronal TDNN avec 6128 sorties tri-phones. Le réseau TDNN contient 7 couches à 650 dimensions par couches, le noyau de prise en compte du contexte de chaque couche de TDNN est ([-1,1], [-1,1], [-3,3], [-3,3], [-3,3], [-3,3]).

Système DNN-HMM-TDNN-F

La modélisation acoustique de ce deuxième modèle apporte une modification à l'architecture du modèle DNN-HMM-TDNN classique. Je mets en oeuvre l'architecture appelée "TDNN-Factorized" proposée par Povey et al. (2018a). Cette modélisation permet d'introduire une matrice semi-orthogonale à l'intérieur des couches TDNN ; nous étudions plusieurs dimensions de cette matrice à savoir 160 et à 256, dans nos expérimentations. Comme nous le verrons lors de l'évaluation, cette approche a pour objectif de réduire le temps d'inférence, sans perte de performance.

Système DNN-HMM-TDNN-F-selfAttn

La modélisation acoustique de ce troisième type de modèle consiste à rajouter une couche dite de "self-attention" (Povey et al. (2018b)) à la modélisation DNN-HMM-TDNN-F. Cette couche d'attention est constituée de 15 têtes d'attention avec un vecteur valeur de dimension 80 et un vecteur clé de dimension 40. Le contexte gauche/droit est respectivement de 5 et 2 avec un recouvrement de 3.

Par ailleurs, j'utilise les *lattices* générés par le système GMM-HMM-Tri-SAT pour effectuer l'apprentissage des modèles acoustiques neuronaux, cela permet une meilleure flexibilité du choix des l'alignement (supervision) des données par la modélisation neuronale.

Les évaluations de chacun des systèmes DNN-HMM proposés, sur la langue française, font l'objet du paragraphe . Pour chaque système, un modèle de langage est nécessaire : ce modèle de langage a été fixé après une évaluation faite avec le système GMM-HMM-Tri-SAT, qui fait l'objet du paragraphe suivant.

4.2 Évaluation de la modélisation linguistique

Le modèle de langage (LM) proposé dans mon travail est estimé à partir de sous ensembles d'un gros corpus de données en langue française de 4 Milliards de mots. Le travail de collecte et de nettoyage de données est décrit dans la section 3.3 et il a été réalisé conjointement par mon stagiaire à Linagora et moi même. Cet ensemble rassemble les corpus FRwac, Wikipédia, Gigaword FR, le journal "Le monde" ainsi que le journal "L'est républicain", les transcriptions des débats parlementaires européens ainsi que la transcription textuelle des ensembles d'apprentissage des corpus de parole.

Nous proposons plusieurs modèles de langage relevant d'une approche statistique de type n-gram. Les modèles diffèrent par la valeur de n qui prendra les valeurs 3, 4 et 5. L'algorithme Kneser-Ney Discount à été utilisé pour construire nos modèles de langage.

Pour la valeur n=3, nous étudions l'influence de la taille du vocabulaire et de la nature des données d'apprentissage. Pour cela, je compare la génération d'un LM estimé sur l'ensemble textuel issu de tous les corpus de parole (appelé ci-après "corpus parole") par rapport à un LM estimé à partir de l'ensemble des corpus textuels issus de texte écrit (appelé ci-après "corpus écrit"). Au final un modèle de langage est estimé sur l'ensemble des deux corpus (appé ci-après "corpus global"), avec un élagage ("prunning") d'un minimum de 10 occurrences par mot pour construire le dictionnaire lexical.

À des fins de comparaison, un ML 3-gram de base est appris sur le seul corpus ESTER1/2. J'utilise les conclusions pour la valeur n=3 afin de fixer les caractéristiques de l'apprentissage des modèles de langage 4-gram et 5-gram.

Les évaluations de ces modèles sont basées sur deux mesures : la perplexité et le WER (Word Error Rate) dans une tâche de RAP. Même si plusieurs modélisations acoustiques sont présentement proposées et ont été évaluées, nous limitons, dans ce document, la présentation des évaluations avec le système GMM-HMM-tri-SAT 4.1.1.

L'ensemble des résultats évalué sur le corpus de test de ESTER 1 est rassemblé dans le tableau 4.1. On remarque que la génération d'un LM 3-gram avec l'ensemble d'apprentissage ESTER permet d'obtenir une perplexité assez élevé de 134 et un WER de 22.3%. Lorsque nous examinons l'influence de la nature du corpus d'apprentissage (corpus parole contre corpus écrit), nous observons de grandes différences en terme de perplexité, de taille de vocabulaire et de WER. En terme de perplexité, on remarque que le modèle LM 3-gram estimé sur le "corpus écrit" est moins performant que celui estimé sur le "corpus de parole" : les WER sont respectivement de 23.6% vs 20.8% avec respectivement 135 vs 113 en perplexité et 300k vs 88K pour la taille du vocabulaire. Cela est dû au fait que le LM estimé sur le corpus texte écrit est très rigide pour la transcription de parole spontanée malgré sa richesse du vocabulaire.

Le "corpus global" associé à la technique d'élagage permet de générer un vocabulaire de 250K de mots ; la perplexité 76 du ML est relativement faible et le WER de 19.6% est diminué.

<table>
<tr><th>Modèle</th><th>type</th><th>Corpus apprentissage</th><th>Nb mots apprentissage</th><th>Taille vocabulaire</th><th>Perplexité</th><th>WER</th></tr>
<tr><td>Sphinx-LM [1]</td><td>3-gr</td><td>ESTER 1/2 + Le monde</td><td>-</td><td>114K</td><td>156</td><td>21.1</td></tr>
<tr><td rowspan="6">IRIT-LM</td><td rowspan="3">3-gr</td><td>ESTER1/2</td><td>5M</td><td>38K</td><td>134</td><td>22.3</td></tr>
<tr><td>corpus oral</td><td>20M</td><td>88K</td><td>113</td><td>20.8</td></tr>
<tr><td>corpus écrit</td><td rowspan="3">∼4000 M</td><td>300K</td><td>135</td><td>23.6</td></tr>
<tr><td rowspan="3">corpus global</td><td rowspan="3">prune 250K</td><td>76</td><td>19.6</td></tr>
<tr><td>4-gr</td><td>82</td><td>20.1</td></tr>
<tr><td>5-gr</td><td>95</td><td>21.2</td></tr>
</table>

TABLE 4.1 – Évaluation des performances sur l'ensemble de test ESTER 1, des modèles de langage en terme de perplexité et de WER, avec le système GMM-HMM-Tri-SAT.

L'extension de la modélisation LM sur ce corpus mixte (parole + texte écrit) à des n-gram plus grands (N=4, N=5) se traduit par une baisse de performance en termes de WER. Cela est dû à la fois à une plus grande rigidité du modèle de langage dès lors que n augmente, ce qui réduit la flexibilité nécessaire en parole spontanée (remplie d'hésitation), mais cela reflète aussi la limite des N-gram à apprendre des statistiques sur les fréquences des mots sans prendre en compte un apprentissage de représentation sémantique plus intéressant.

Je m'arrête dans le cadre de ma thèse à ce stade de l'exploration des modèles de langages pour me concentrer d'avantage sur la partie acoustique. Dans nombre des expériences suivantes, le modèle de langage

1. Sphinx-LM : `https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/French/`

retenu sera le modèle 3-gram estimé sur le "corpus global" avec élagage (perplexité 76 sur l'ensemble test ESTER1).

4.3 Évaluation de la modélisation acoustique

Nos systèmes LVCSR GMM-HMM et DNN-HMM sont appris sur un ensemble de corpus de 340 heures issues des corpus ESTER 1,2, EPAC, BREF-80 et Librivox français (cf section 3.2.1) ; l'évaluation de ces modèles est réalisée sur les corpus de test suivants : ESTER Test 1 et 2 pour la parole préparée, Bref-80 pour la parole lue, ETAPE et EPAC pour la parole spontanée. Les durées de chacun des corpus sont rassemblées dans le tableau 3.1.

Dans cette série d'expérimentations, j'examine uniquement les variantes pour la modélisation acoustique (modèles et technique d'apprentissage). Pour cela, j'utilise pour mes modèles GMM-HMM une extraction MFCC de 13 coefficients avec les dérivées premières et secondes et un contexte de 7 fenêtres (3 à gauche, 3 à droite), une LDA est appliquée pour réduire le vecteur acoustique en une dimension de 40. L'extraction des paramètres acoustiques pour mes modèles hybrides est similaire au GMM-HMM avec 40 coefficient MFCC à la place de 13. Comme dit précédemment, le modèle de langage est le modèle statistique 3-gram estimé sur le "corpus global" avec élagage.

Les expérimentations proposées ont été réalisées de manière progressive ; nous rapportons ainsi :

1. Une évaluation des modèles acoustiques GMM-HMM et DNN-HMM de base et une comparaison de nos systèmes avec ceux issus des travaux de laboratoires français.
2. Une mesure de l'apport des variantes sur les modèles eux-mêmes.
3. Une expérimentation de la technique d'apprentissage basée sur l'augmentation/perturbation de données pour accroître la robustesse acoustique.

4.3.1 Évaluation des systèmes GMM-HMM et DNN-HMM de base

Le tableau 4.2 présente une vue d'ensemble sur les performances de nos modèles de base GMM-HMM et hybrides DNN-HMM sur l'ensemble d'évaluation d'ESTER 1. Nous y rapportons les performances du système développé par le LIMSI Galliano et al. (2005), le système le plus performant, présenté dans le cadre de la campagne d'évaluation ESTER. Avec une modélisation de type GMM-HMM et l'utilisaton de triphones, le système du LIMSI obtient un WER de 11.9, avec un corpus de données additionnelles d'une durée de 100H, privé au LIMSI. L'ensemble des participants à cette compagne d'évaluation ont un WER de 23,6% et de 26,7% voir plus.

Laboratoire-Système	Modèle	Données	Adapt Loc	WER
LIMSI(Galliano et al., 2005)	GMM	ESTER + priv 100h	SAT	11.9
LIA(Galliano et al., 2005)		ESTER		26,7
LIUM(Galliano et al., 2005)				23,6
GMM-HMM-Tri-MLLT	GMM	w/o	MLLT	24.8
GMM-HMM-Tri-SAT			SAT	19.6
DNN-HMM-TDNN	TDNN		I-vect	11.9
		Perturbation		11.03

TABLE 4.2 – Résultats des performances de nos meilleures systèmes de base en terme de WER sur l'ensemble de test ESTER 1. Le modèle de langage à 76 de perplexité est utilisé pour le décodage et le triphone est l'unité élémentaire.

À titre d'information, lors de la mise en place du premier système GMM-HMM-monophone utilisant le meilleur LM 3-gram exploré précédemment, nous avons obtenu un WER de 40.4% ; les modèles sont appris sur un sous ensemble d'apprentissage restreint à 17H. Les améliorations de ce système GMM-HMM-monophone obtenu par passage d'une modélisation acoustique de monophone à triphone avec rajout de la transformation MLLT permet un gain significatif avec un WER de 24.8%. La prise en compte de l'adaptation au locuteur par la technique SAT permet de disposer d'une bon système baseline de 19.6% en WER. À cette étape, nous n'utilisons pas de perturbation de données : la modélisation GMM n'est pas assez complexe pour espérer une généralisation correcte à partir des données perturbés.

Je rappelle que je génère, à partir du système GMM-HMM-Tri-SAT, les "lattices" sur toutes les données d'apprentissage pour permettre leur utilisation en apprentissage par des systèmes hybrides avec la fonction LF-MMI.

Le système hybride DNN-HMM-TDNN présenté dans le tableau 4.2 utilise une modélisation du locuteur par i-vecteur. On observe qu'un gain absolu de 7.2% de WER est obtenu sans augmentation de données (w/o) à partir de la simple modélisation "chain-model" TDNN, décrite dans la section 4.1.2. Nous sommes aperçus assez rapidement que la perturbation des données d'apprentissage pouvait être intéressante. Nous avons augmenté le volume des données d'apprentissage en en effectuant 9 perturbations ((3x) pour la perturbation du rythme et (3x) pour la réverbération du signal de parole par le corpus "Simulated RIRS" (Ko et al., 2017)), pour atteindre un total de 3000heures. L'introduction de cette technique dans le système DNN-HMM-TDNN permet un gain de 0.9% de WER. Nous l'avons par la suite systématiquement utilisé pour apprendre les différentes variantes de modélisation.

Le temps de décodage est un facteur important pour des applications "temps réel". Pour le système DNN-HMM-TDNN, il apparaît que le RTF mesuré est de 0.6 qui est certes acceptable mais loin des attentes temps réel de notre projet de recherche OpenPaas :NG.

4.3.2 Évaluation des variantes des systèmes de type DNN-HMM

La solution proposée par Povey et al. (2018a) qui permet de découper les couches TDNN avec une matrice semi orthogonal, technique appelée TDNN-Factorized, a conduit à l'élaboration du système DNN-HMM-TDNN-F. Nous avons exploré la dimension de la matrice semi orthogonale : il apparaît d'après les évaluations rapportées dans le tableau 4.3 que la performance du TDNN-F en terme de WER est sensiblment la m ême avec un WER de 10.8 %, que la dimension soit prise égale à 256 ou 160. Cependant, la performance en temps de décodage est nettement amélioré avec un RTF de 0.4 pour une dimension "bottleneck" de 160 sur la matrice semi orthogonale du TDNN-F.

Laboratoire-Système	Variante Acoustique	Data Aug	WER	RTF
LaBRI Boyer and Rouas (2019)	TDNN	Spd + Vol (231hx3)	13.7	
DNN-HMM-TDNN	TDNN	w/o	11.9	0.6
		Spd + Rvrb (340x9)	11.03	
DNN-HMM-TDNN-F	TDNN-F dim=256		10.8	0,45
	TDNN-F dim=160		10.8	0,4
DNN-HMM-TDNN-F-selfAttn	+ Self Attn		10.5	0,45

TABLE 4.3 – Résultats des performances de systèmes hybrides en terme de WER sur l'ensemble de test ESTER 1. Le modèle de langage à 76 de perplexité est utilisé pour le décodage. Les données d'apprentissage sont augmentés, quelque soit le système. La mesure RTF en anglais "Real Time Factor" permet d'évaluer le temps de transcription des audios (plus le RTF est bas, plus le système est rapide).

L'évaluation du système DNN-HMM-F-selfAttn utilisant la technique self-attention décrite dans la section 4.1.2, montre une amélioration de 0.3% en terme de WER avec une performance de 10.5% et un RTF

de 0.45, ce qui correspond à notre meilleur résultat en termes de WER.

Nos systèmes trois hybrides peuvent être comparés au système développé par le LABRI Boyer and Rouas (2019) ; pour ce système, un modèle hybride TDNN-HMM à 7 couches similaire à notre TDNN est appris sur les corpus ESTER 1 et 2, EPAC et REPÈRE pour un total de 230 heures et une perturbation de rythme et de volume (x3) est appliquée aux données. D'après la littérature, ce système affiche un WER de 13.7% qui doit être comparé aux performances de notre système DNN-HMM-TDNN avec perturbations de données.

4.3.3 Analyse approfondie du système DNN-HMM-TDNN-F-selfAttn

Notre système DNN-HMM-F-selfAttn permet de surpasser l'état de l'art actuelle avec une performance de 10.5% de WER sur l'ensemble ESTER1. Je propose donc d'évaluer notre système LVCSR en français sur tous les ensembles de test des corpus de références pour la langue française et d'en analyser les erreurs.

Influence du type de parole (lue, préparé, et spontanée)

Le tableau 4.4 présente l'évaluation de notre système sur les corpus regroupés par type de parole (parole lue, préparée et spontanée) en mettant en avant l'intèrêt d'utiliser les techniques de perturbations ; pour rappel, nous avons agi sur le rythme et la réverbation. le "Signal Noise Ratio (SNR)" calculé sur l'ensemble des prononciations des corpus a pour valeur moyenne 17 db avec une variance de 15%.

Est obtenu par le système DNN-HMM-TDNN-F-selfAttn un WER inférieur à 10% sur de la parole lue (avec 9.7% de WER sur le corpus BREF). Concernant la parole préparée, j'ai proposé de perturber le corpus ESTER 1 avec du bruit de parole de fond "babble noise" (SNR de 10 db), on remarque que le système hybride appris avec des perturbations de rythme et de réverbération permet de garder pratiquement la même performance obtenue sur ESTER clean avec 12.7% de WER contre 21.5 % sur ESTER bruité. L'intérêt d'utiliser ce type de perturbation est devenue obligatoire pour faire face à la robustesse acoustique de nos systèmes RAP. Concernant la parole spontanée, ETAPE et EPAC sont deux corpus issue respectivement d'émissions télévisés et radiophoniques en parole spontanée, on reporte une performance de 22,3 % et de 16,6 % en WER sur les deux corpus respectifs.

Corpus de test	Type de parole	Data Aug	
		Sans	Avec
BREF-80	Lu	10.5	9.7
ESTER 1	Préparé	11.1	10.5
ESTER 1 + Babble Noise (SNR 10)		21.5	12.7
ESTER 2		12.6	11.7
ETAPE	Spontané	26.44	22.3
EPAC		18.1	16.6

TABLE 4.4 – Evaluation du modèle Chain TDNN-F + self attn LF-MMI sur les différents corpus de la langue française par type de parole.

Analyse des erreurs

Je propose d'analyser quelques erreurs de notre système LVCSR pour comprendre les types d'erreurs. Le tableau 4.5 présente des exemples de transcriptions issus du corpus ESTER 1 mettant en avant les erreurs réalisées par notre modèle. On observe des erreurs d'accord singulier/pluriel, des erreurs de conjugaison singulier/pluriel ainsi que des substitutions moins fréquentes mais plus graves de type "pose-t-il"/"poser".

Grand nombre d'erreurs sont imputables à la modélisation linguistique ; il semble que la façon avec laquelle la préparation de données a été faite en séparant les liaisons de mots "n' est", "qu' il", "aujourd' hui" etc... génère une dégradation de performance de la modélisation linguistique de notre modèle. Il est important pour la langue française d'étudier la normalisation du texte pour offrir une meilleure intégration entre la modélisation acoustique, lexical et linguistique.

Exemple	Prononciation
Original	comme les forces de l' ordre sont soucieuses de neutraliser les suspects plutôt que de les tuer il est prévu d' expérimenter de nouvelles armes non **létales**
Prédiction	comme les forces de l' ordre sont soucieuses de neutraliser les suspects plutôt que de les tuer il est prévu d' expérimenter de nouvelles armes non **létale**
Original	alors la vie quotidienne maintenant parce que c' est comme ça la vie quotidienne c' est les pv aussi euh plus de pv sur les **pare-brises**
Prédiction	alors la vie quotidienne maintenant parce que c' est comme ça la vie quotidienne c' est les pv aussi euh plus de pv sur les **pare-brise**
Original	bonjour le gouvernement et l' opposition **enterrent** la polémique sur la mission de didier julia pour libérer les otages en irak
Prédiction	bonjour le gouvernement et l' opposition **enterre** la polémique sur la mission de didier julia pour libérer les otages en irak
Original	c' est aujourd'hui le parisien **qui l'** annonce à la une
Prédiction	c' est aujourd'hui le parisien **qu' il** annonce à la une
Original	peut-être se **pose-t-il** la question aussi de la présidence de la république
Prédiction	peut-être se **poser** la question aussi de la présidence de la république
Original	vous nous appelez du **val-d'** oise
Prédiction	vous nous appelez du **val d'** oise
Original	un spécialiste de la nutrition **le dit** au figaro l' alimentation est en cause dans trente à soixante pour cent des cancers c' est dire si c' est important
Prédiction	un spécialiste de la nutrition **l' audit** au figaro l' alimentation est en cause dans trente à soixante pour cent des cancers c' est dire si c' est important
Original	d' autant plus qu' on nous présente tout ça comme une panacée alors que le nucléaire rappelons-le n' est qu' un pis-aller
Prédiction	d' autant plus qu' on nous présente **euh** tout ça comme une panacée alors que le nucléaire rappelons-le n' est qu' un pis-aller

TABLE 4.5 – Exemple de phrase issue du décodage de l'ensemble de test ESTER 1 par notre système LVCSR chain TDNN-F + Self Attn.

Conclusion

Au vu des résultats intéressants que notre système DNN-HMM-TDNN-F-selfAttn basé sur un modèle hybride permet d'obtenir, il a été décidé, dans le cadre de ma thèse, de mettre en production ce modèle chez l'entreprise Linagora via le projet OpenPaas :NG et Linto. La section 4.4 détaille l'effort d'industrialisation réalisé pour la mise en production d'un service RAP adapté au besoin métier.

4.4 Application industrielle et adaptation aux contextes clients

Compte tenu du contexte de ma thèse (CIFRE avec l'entreprise Linagora), j'ai participé aux différents transferts technologiques liés à l'exploitation industrielle de mes recherches sur les systèmes RAP hybrides. D'une part j'ai développé une plate-forme "LinSTT" pour "Linagora Speech To Text" [2]) afin de proposer une solution open-source pour définir un système hybride de RAP. D'autre part j'ai réalisé plusieurs projets

2. Linagora Innovation : `https://linagora.com/innovations`

industriels ayant comme support cette plate-forme.

4.4.1 La plate-forme LinSTT

La solution open-source "LinSTT" regroupe l'industrialisation de la phase d'apprentissage (appelée LinSTT Model Factory [3]) et celle de décodage (appelée LinSTT Engine) des systèmes RAP hybrides basés sur l'outil Kaldi. La solution en question a pour but de proposer aux clients de Linagora au travers de la plate-forme LinSTT, un ensemble d'outils pour l'adaptation d'un système RAP à différents stades jusqu'à sa mise en production.

La figure 4.1 représente une vue générale de la solution LinSTT. En suivant le cadre formel défini par cette approche, il est possible de découpler l'ensemble des contraintes fonctionnelles liées à la prise en compte du contexte métier spécifique à un cas d'usage (traitées par LinSTT Model Factory), de l'ensemble des contraintes techniques de déploiement propres aux cas d'usages retenus (traitées par LinSTT Engine). Nous ne détaillons pas, dans le cadre de cette thèse, les travaux d'implémentation disponibles en open-source sur github et des logiques métiers spécifiques propres au traitement des requêtes utilisateurs (bloc jaune dans la figure).

Cette solution permet aux clients un accès total au modèle RAP à différents niveaux, à savoir :

- avec le service LinSTT Model Factory :
  - La mise à jour du modèle acoustique avec le rajout de données réelles à partir de des flux audio privés/internes du client.
  - La mise à jour du dictionnaire de prononciation pour la prise en compte du vocabulaire spécifique d'un client donné (vocabulaire technique dédié).
  - La mise à jour du modèle de langage par un ensemble de document permettant l'adaptation du système RAP large vocabulaire au besoin du client. Dans le cas d'identification d'entité spécifique, il est inclus la création d'un graphe "FST" pour la reconnaissance d'entités.
- avec le service LinSTT Engine : Sont proposées des API de décodage offline et streaming avec un ration RTF de 0.45 sur un ensemble de système RAP dédié à des taches données, par exemple un service LinSTT Engine LVCSR, un service LinSTT Engine d'identification de matricule, un service LinSTT Engine d'identification de nom prénom, etc...

Beaucoup de challenge R&D ont été abordés pour la réalisation de la plateforme LinSTT, l'adaptation des modèles RAP est un enjeu majeur pour palier aux limites des systèmes LVCSR sur des taches précis et des conditions acoustiques connues. LinSTT Model Factory permet de prendre en charge de nouveaux données en environnement réel pour l'amélioration des performances du système RAP, l'adaptation lexical est linguistique proposée par LinSTT Model Factory pour la reconnaissance de commande est d'entité (alphanumérique, nom et prénom, etc..) a connu un grand interêt dans le monde de l'industrie.

4.4.2 Deux exemples de transfert

Durant ma thèse, j'ai contribué à la réalisation de 5 projets clients de co-innovation, avec leurs équipes R&D, en mettant en place le service LinSTT chez Société Générale, Air France pour la création de systèmes RAP dédiés à des cas d'usage différents.

Société générale , deux cas d'usage ont été étudiés :

- Cas d'usage 1 : Système RAP pour l'appel automatique des collaborateurs en interne avec la reconnaissance des noms et prénoms. Ma contribution a consisté à adapter le dictionnaire pour le rajout/suppression de la phonétisation des noms prénoms avec LinSTT Model Factory et à assurer le déploiement en continu des nouveaux modèles ; cette API a été intégrée avec l'outil Lyx à la Société Générale pour simplifier l'appel des collaborateurs.

3. LinSTT Model Factory : `https://github.com/linto-ai/linstt-factory`

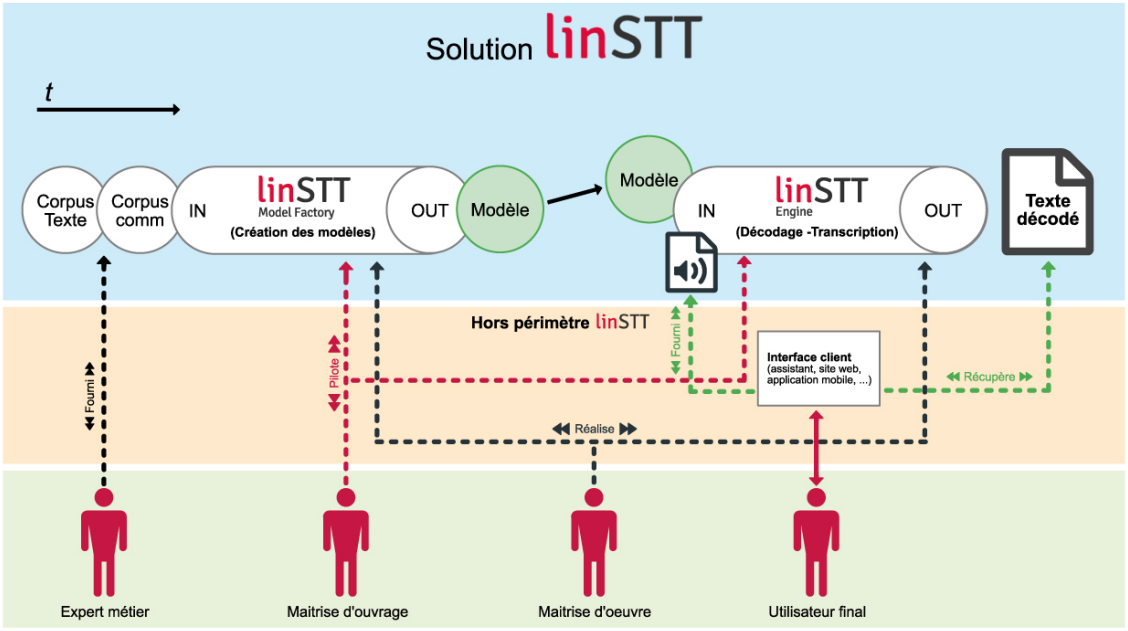


FIGURE 4.1 – Vue générale de notre méthodologie en découplant les tâches et composants qui relèvent du traitement des modèles et données via LinSTT (bloc bleu), du traitement fonctionnel des requêtes métiers de l'utilisateur final (bloc jaune) et des acteurs travaillant conjointement à mener l'approche (bloc vert).

— Cas d'usage 2 : Système RAP pour la gestion de l'application mobile de la Société Générale par la voix. Ma contribution a consisté à adapter le graphe et le dictionnaire pour la gestion vocale des fonctionalités de l'API de la Société Générale (suivi de solde, des virements..).

Air France , trois projets ont été menés.

— Cas d'usage 1 : Système RAP pour l'identification du matricule collaborateur. Ce système a été intégré à un agent conversationnel d'information des jours de congés du collaborateur. J'ai adapté le système RAP pour la reconnaissance numérique.

— Cas d'usage 2 : Système RAP LVCSR pour la transcription de notes effectuées par le chef de cabine des avions AirFrance au cours du vol. Ma contribution a porté sur le système RAP LVCSR et LinSTT Engine.

— Cas d'usage 3 : Système RAP LVCSR pour le call center AirFrance afin de suivre les échanges clients/opérateurs. J'ai contribué à l'adaptation avec LinSTT Model Factory pour l'apprentissage d'un modèle 8khz avec l'encodage "A-law" pour l'adaptation à la téléphonie.

En conclusion, la solution LinSTT a été un grand succès de transfert technologique de la recherche à l'industrie ; beaucoup de cas réels liés à la robustesse acoustique et à la gestion des graphes dans mes systèmes RAP hybrides ont été abordés et traités. Cette solution a été la base de l'initiation des travaux sur le projet de recherche Linto [4] ; ce projet reprend les briques de base de LinSTT Model Factory/Engine pour la construction d'une plate-forme d'agents conversationnels pour les entreprises (avec l'intégration de la NLU).

De nombreuses perspectives s'offrent pour la prochaine génération de LinSTT Model Factory pour le "Federated Learning" ; cela sera abordé dans la section 7.2 avec le projet de recherche "Le voice Lab" financé par BPI France.

4.5 Conclusion

Comme nous l'avons vu, j'ai proposé une étude approfondie sur la modélisation RAP hybride DNN-HMM pour la langue française. Nous avons choisi une modélisation acoustique GMM-HMM pour générer graduellement des alignements de nos données ainsi que la génération des "lattices" et pour atteindre un apprentissage neuronale plus flexible avec la technique "Lattice Free MMI". Notre choix de modélisation acoustique neuronale repose sur l'architecture TDNN qui est très adapté pour l'encodage du contexte acoustique.

Nous avons proposé 3 variantes pour la modélisation DNN-HMM à base de TDNN et avons évalué leur performance en termes de WER et de temps de décodage sur le corpus ESTER 1 ; il apparaît que l'architecture TDNN-Factorized + self attention permet d'établir un nouveau état de l'art pour la RAP en français avec 10.5% de WER. Nous reportons les résultats de ce système DNN-HMM-TDNN-F-selfAtt sur tout type de parole : lue, préparée et spontanée, et nous le proposons comme baseline pour les corpus de parole en Français : Bref, ESTER, EPAC, ETAPE.

Une partie industrialisation a été au coeur de ma contribution sur les systèmes hybrides ; la mise en place de LinSTT model factory permet de proposer un ensemble de modules pour l'adaptation acoustique, lexicale et grammaticale à des cas d'usage entreprise.

Cependant, il ressort de nos travaux que le composant lexical joue un rôle clé pour lier la modélisation acoustique et linguistique. En particulier, nous avons observé une grande difficulté dans la mise en oeuvre de notre plateforme Model Factory 4.4.1 pour l'adaptation des modèles hybrides à des applications industrielles. Il apparaît par exemple que l'extension du vocabulaire par des termes spécifiques à un domaine/entreprise doit faire l'objet d'un processus spécifique d'extension qui consiste à :

1. Vérifier l'existence de la prononciation dans le dictionnaire lexical du système hybride.
2. Générer automatiquement la phonétisation à l'aide d'un modèle Grapheme-To-Phoneme des mots inexistants. Cette deuxième tache pose beaucoup de problèmes dans le sens où la génération automatique peut être erronée. Aussi l'utilisateur de la plateforme de mise à jour du modèle de prononciation doit être un expert du domaine pour valider ces modifications.
3. Mettre à jour le modèle de langage en rajoutant du contenu textuel permettant d'estimer l'apparition des nouveaux mots par rapport à leur contexte.

Ce processus doit être réalisé à chaque fois que l'utilisateur de la plateforme LinSTT souhaite adapter le composant lexical à son cas d'usage. D'autres recherches telles (Bisani and Ney, 2005; Parada et al., 2010; Qin, 2013) se sont intéressées à la détection et la récupération des mots hors du vocabulaire dans les systèmes hybrides. Malgré ces efforts de recherche, il est clair que l'arrivée des approches End-to-End pour la RAP proposant une modélisation graphème (caractère, subword, ou mot) directement à partir des observations audio ouvre de nouvelles perspectives avec une modélisation dite "Lexicon-Free", d'où l'orientation de mes recherches comme cela est décrit dans les deux chapitres suivants.

4. Linto :`https://linto.ai`, `https://www.irit.fr/projet-pia-linto/`

Chapitre 5

Char+CV-CTC : approche multi-tâche dans un système End-to-End

Introduction

Nous étudions s'il est possible (et surtout intéressant en termes de difficulté de mise en oeuvre et de performances) de remplacer la modélisation acoustique phonétique du système DNN-HMM hybrides à trois composants (acoustique, lexical, et linguistique) par une modélisation acoustique CTC basé sur les "caractères". Comme indiqué au paragraphe 2.3.1, l'approche End-to-End pour la RAP permet effectivement d'accéder à une modélisation graphème (caractère, subword, ou mot) directement à partir des observations audio et de se libérer de la contrainte de pré-définition d'un dictionnaire. En d'autres termes, la modélisation neuronale End-to-End apprend les représentations linguistiques avec le réseau de neurones, seul composant pour la reconnaissance des graphèmes à partir de l'audio.

Il est clair que le passage d'une modélisation phonétique avec un pipeline d'alignement forcé/re-estimation du modèle hybride DNN-HMM vers une modélisation E2E CTC à base de graphèmes apporte une dimension de complexité supplémentaire qui est liée à la difficulté d'apprendre les représentations linguistiques directement à partir du réseaux de neurones. Pour cela, nous nous sommes orienté vers l'exploration des approches dites "multi-tâche" pour proposer une solution exploitant plusieurs informations linguistiques et améliorant la représentation linguistique apprise implicitement par le modèle CTC.

Ce chapitre est organisé comme suit : après avoir défini un système de base CTC-graphème qui nous servira de référence (section 5.1) et après avoir donné quelques exemples de systèmes CTC-MLT, nous décrivons les différentes variantes proposées du modèle MTL (section 5.2). Nous évaluons les modèles sur le corpus du Wall Street Journal, et comparons les différentes approches multi-tâches dans la section 5.3.

5.1 Le système CTC-GRU de référence

Nous nous intéressons dans cette première section à la modélisation CTC "caractère". Pour ce faire, nous utilisons comme unité de prédiction l'ensemble des 26 caractères de l'alphabet avec 3 autres unités : l'espace, l'apostrophe et le "blanc". Nous présentons en détail dans ce qui suit le corpus et la modélisation neuronale utilisés.

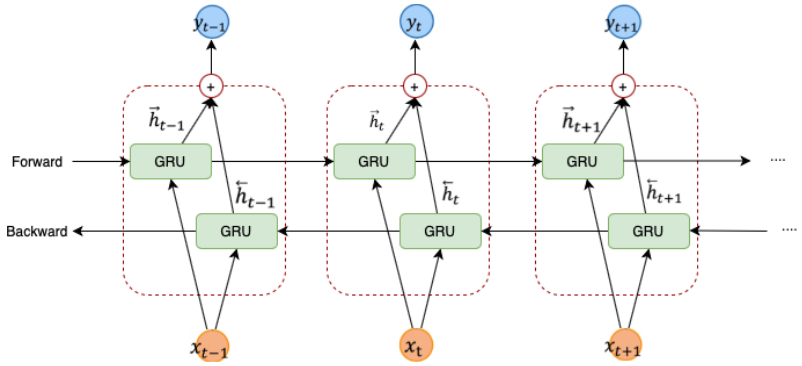


FIGURE 5.1 – Exemple de notre architecture GRU bidirectionnelle à deux cellules sur 3 étapes temporelles.

5.1.1 Corpus d'étude et traitement de signal

Mon étude est réalisée sur le corpus Wall Street Journal (WSJ, LDC93S6A). Nous utilisons la méthode de préparation des données de Kaldi WSJ. Les 81 heures de l'ensemble d'apprentissage sont divisées en sous-ensembles d'apprentissage (95 %) et de développement (5 %) et elles correspondent aux mêmes sous-ensembles utilisés dans le projet Eesen(Miao et al., 2015). L'évaluation est effectuée sur l'ensemble *Eval92*. L'ensemble *Dev93* est utilisé pour fixer la valeur optimale du poids de mélange λ pour les modèles MTL (section 5.2).

Les coefficients filter-bank Log-Mel avec 40 bandes de fréquences sont extraits avec les premières et secondes dérivées du signal audio. Ces paramètres sont ensuite normalisées par soustraction de la moyenne et normalisation de la variance (CMVN). Aucune technique d'adaptation aux locuteurs n'est utilisée. Cependant, pour avoir plus de contexte et réduire la phase d'apprentissage, comme dans Krishna et al. (2018), des paires de filter-bank en entrée du réseau sont concaténées.

5.1.2 Le modèle BiGRU

Le modèle de base est un modèle de type CTC-GRU (section 5.1). J'ai choisi d'utiliser des couches récurrentes bidirectionnelles GRU (Chung et al., 2015) tout en gardant le même nombre de paramètres du modèle par rapport aux travaux à l'état de l'art avec lesquels je me compare, à savoir entre 8M et 9M de paramètres.

Après avoir expérimenté plusieurs configurations de notre modèle Bi-GRU, nous optons pour un modèle avec 4 couches BiGRU de 2×320 paramètres par couche, soit un total de $8,3$M de paramètres. La couche de sortie utilise la fonction d'activation log-softmax.

5.1.3 Les systèmes mis en oeuvre

Nous proposons deux types de système durant cette étude : 1) Un système RAP CTC avec un décodage GreedySearch (qui correspond à la partie "Lexicon-Free"), 2) Un système RAP CTC-WFST avec un décodage WFST ("Weigthed Finite State Transducer") à base de graphe permettant de proposer un système à base de modélisation acoustique CTC graphème et un décodage WSFT. Nous nous basons sur l'outil Eesen proposée par Miao et al. (2015).

Phase d'apprentissage Un Dropout de 10% (Srivastava et al., 2014) est appliqué sur la sortie de chaque couche BiGRU. Le réseau est appris avec 100 itérations et une taille de batch de 32, avec la fonction d'objectif CTC, et l'optimiseur ADAM Kingma and Ba (2014) avec un taux d'apprentissage de 4e-5 et une

valeur de 0.9 pour le momentum. La technique qui consiste à utiliser des paires de filter-bank (Krishna et al. (2018)), appelée "Time Reduction (TR)", accélère l'entraînement, en utilisant une Nvidia GTX-1080 TI. L'apprentissage prend 34 heures au lieu de 58 heures sans la réduction temporelle (TR) pour 100 itérations sur l'ensemble d'apprentissage de WSJ (ce qui fait un total de 34 minutes par itération sur WSJ).

Inférence Pour l'inférence, nous utilisons un décodage "Greedy Search" sans modèle de langage, et un décodage WFST ("Weighted Finite State Transducer") tel qu'implémenté dans EESEN(Miao et al., 2015), avec un dictionnaire contenant les mots et leurs séquences de caractères correspondantes. Le modèle de langue, disponible dans le corpus WSJ, est un trigramme avec un vocabulaire de 20 000 mots. Tous les modèles sont implémentés avec l'outil PyTorch (Paszke et al. (2017)) et le décodage WFST est fait avec l'outil EESEN ; un effort d'intégration des deux outils est le fruit de ce travail et le produit final est distribué à l'équipe SAMOVA pour de futurs travaux.

5.1.4 Evaluation des systèmes CTC-GRU

Le tableau 5.1 présente les taux d'erreurs au niveau caractères et mots (CER, WER) obtenus avec nos modèles CTC-GRU caractère, en utilisant les algorithmes de décodage "Greedy Search" et WFST. Dans la partie supérieure du tableau, nous présentons les résultats de systèmes de l'état de l'art, opérant sur les mêmes données et usant de modèles ayant une architecture similaire à la nôtre, avec quatre couches et environ 8 à 9 millions de paramètres d'apprentissage.

TABLE 5.1 – Résultats de notre système CTC-GRU mono-tâche caractère sur l'ensemble d'évaluation Eval92 en termes de CER et WER. L'additif TR2 signifie qu'on applique la technique de "Time-Reduction" expliqué dans la section 5.1.2.

Model	Greedy Search			WFST-based	
	CVER	CER	WER	CER	WER
TDNN-Phone Miao et al. (2015)	_	_	_	_	7.1
CTC-LSTM Miao et al. (2015)	_	_	_	_	7,3
CTC-LSTM Graves and Jaitly (2014)	_	9.2	30.1	_	8.7
TDNN-Phone+TR2	_	_	_	_	6.0
CTC-GRU	_	8.4	29.6	2.8	7.4
CTC-GRU+TR2	_	8.0	28.8	2.4	6.8

Notre système de référence CTC-GRU caractère en mono-tâche avec un décodage WFST atteint un WER de 7,4%, comparable à celui des systèmes CTC-LSTM caractère présentés par (Graves and Jaitly (2014); Miao et al. (2015)) ($8,7\%$ et $7,3\%$, respectivement). A titre de comparaison, nous rapportons également la performance de 7,1% en terme de WER, obtenue par le système TDNN/HMM phonétique(Miao et al., 2015), ce système obtient de meilleurs résultats que notre système CTC-GRU avec une différence en WER absolu de 0.3%.

Avec un décodage sans modèle de langage avec un "Greedy Search", notre système CTC-GRU mono-tâche obtient de meilleurs résultats avec un CER / WER de $8,4\%$ / 29.6% comparé aux resultats du système CTC-LSTM (Graves and Jaitly (2014)) de $9,2\%$ / $30,1\%$. On remarque que, pour la même approche, l'architecture GRU permet de surpasser l'architecture LSTM d'environ $0,5\%$ en WER. Comme prévu, le décodage WFST en incorporant le modèle de langage et le lexique donne de bien meilleurs résultats que le décodage "Greedy Search". Un gain de $21,8\%$ en terme de WER est observé et confirme la nécessité d'utiliser des modèles hybrides pour des données en faible ressource.

Dans un deuxième temps, nous comparons le rajout de la technique de combinaison des trames pour la réduction du temps de calcul et la prise de contexte plus large. Le modèle CTC-GRU+TR2 permet d'obtenir des améliorations de 0,4% et 0,6% absolues en CER / WER, l'impact du TR est très intéressant au point de

vue du temps d'apprentissage, d'inférence et d'amélioration des performances du système ; nous l'utiliserons pour la suite de nos expérimentations. Nous obtenons une performance de 6.0 % en WER en utilisant la technique de TR à deux trames pour notre système hybride TDNN/HMM phonétique, cette technique nous permet de dépasser largement les performances du modèle hybride TDNN/HMM rapporté dans (Miao et al. (2015)) à $7,1\%$ de WER pour la même modélisation (nombre de paramètres équivalent).

5.2 Les systèmes CTC-MTL

Afin de rendre plus robuste les sorties du système CTC "graphème", nous tentons d'y adjoindre une information complémentaire par fusion ; l'approche CTC-MTL semble la bonne option.

5.2.1 Quelques systèmes CTC-multi-tâches antérieurs

Ce travail s'inspire des études de Jimenez et al. (2018) dans le domaine de la détection d'événements sonores (SED). Cette approche consiste à utiliser le concept d'ontologie. Jimenez *et al.* ont proposé une modélisation neuronale qui produit des sorties hiérarchiques et des prédictions à plusieurs niveaux. L'aspect multi-tâche de cette approche consiste à utiliser une couche ontologique simple définie avec des classes de "haut niveau" telles que "Musique", "Humain", "Animal", pour catégoriser les évènements sonores "de bas niveau" (tâche principale) telle que les classes "violon, piano", "manger, respirer", et "chat, chien". Le passage entre tâche principale et tâche auxiliaire se fait à l'aide d'une matrice binaire M fixe qui permet de convertir les prédictions sonores de bas niveau en prédictions de classes de haut niveau en sommant les probabilités des catégories de bas niveau qui partagent une catégorie de haut niveau commune. Ces travaux ont montré que la tâche de reconnaissance de niveau supérieur améliore considérablement l'exécution de la tâche principale.

En parole, il a été montré que les systèmes RAP End-to-End apprennent implicitement des représentations qui ont un sens du point de vue linguistique, dans les couches intermédiaires (Krishna et al., 2018; Belinkov and Glass, 2017). Les approches d'apprentissage multi-tâche pour les systèmes RAP End-to-End ont pris de l'ampleur au cours des dernières années (Rao and Sak, 2017; Kim et al., 2017b). Des travaux récents ont introduit l'utilisation du MTL hiérarchique dans la RAP avec des modèles hiérarchiques basés sur du CTC (Krishna et al., 2018; Sanabria and Metze, 2018). Krishna *et al.*, par exemple, ont proposé une modélisation RAP CTC hiérarchique pour la prédiction de sous-mots avec une tâche auxiliaire CTC au niveau du phonème appliquée à une couche intermédiaire d'un réseau neuronal. Cet apprentissage MTL hiérarchique a permis de dépasser, en terme de performance WER les approches MTL standard à deux têtes en sortie du réseau.

5.2.2 Trois nouvelles propositions de système CTC-MLT

Dans ce travail, je compare différentes modélisations CTC pour la prédiction de "caractères" avec des variantes MTL qui impliquent une tâche auxiliaire de reconnaissance de deux catégories spécifiques, notés par 'V' et 'C', représentant les voyelles et les consonnes. Pour la tâche auxiliaire, j'utilise cinq unités : "C" pour les consonnes, "V" pour les voyelles, l'apostrophe, le symbole ϵ (blanc) et l'espace ; les trois dernières unités sont nécessaires au bon fonctionnement de la fonction CTC, comme dit dans la section précédente. En liaison avec la tâche de reconnaissance des "caractères", le symbole "V" est choisi pour représenter les voyelles "a", "e", "i", "o", "u" et "y" tandis que le symbole "C" représente toutes les autres lettres. Dans cette configuration, la semi-voyelle 'w' est considérée comme une consonne.

Nous présentons dans la figure 5.2 les trois variantes MTL étudiées : "char CTC" et "CV CTC" indiquent l'emplacement des deux sorties utilisées respectivement pour les tâches de reconnaissance "caractère" et "CV". Les termes "char dense" et "CV dense" désignent des couches de neurones entièrement connectées avec un nombre d'unités neuronales égal au nombre de classes de caractères (26+3) et de CV (2+3).

La fonction Log-softmax est utilisé comme fonction d'activation des sorties. Dans toutes les variantes, nous utilisons la fonction de perte MTL définie dans Eq. (5.1). C'est la combinaison convexe des deux

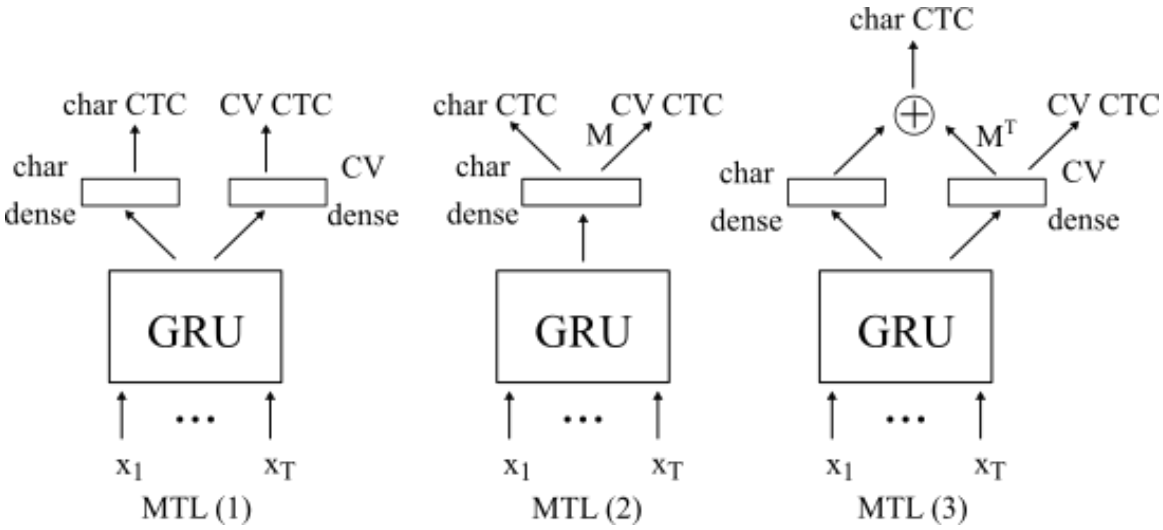


FIGURE 5.2 – Trois architectures pour l'utilisation de l'apprentissage MTL est proposée dans notre travail. MTL (1) : Architecture MTL standard, MTL (2) : approche hiérarchique caractère vers la catégorie CV Jimenez et al. (2018). MTL (3) : Char+CV-CTC MTL.

fonctions de perte CTC : $\mathcal{L}_1$, la fonction CTC liée à la sortie "caractère" correspondant à $\mathbf{p}(\hat{y}_{\text{char}}|x)$, et $\mathcal{L}_2$, la fonction CTC liée à la sortie "CV" correspondant à $\mathbf{p}(\hat{y}_{\text{cv}}|x)$, où $\hat{y}_{\text{char}}$ et $\hat{y}_{\text{cv}}$ indiquent respectivement la séquence de "caractères" et la séquence de catégories CV pour une observation donnée. Ici, λ est un paramètre réel appartenant à $[0, 1]$, à apprendre pour déterminer le poids de chaque tâche. Lorsque $\lambda = 1$, la tâche est réduite à la mono-tâche de reconnaissance de "caractères".

$$\mathcal{L} = \lambda\mathcal{L}_{\text{char}} + (1 - \lambda)\mathcal{L}_{\text{cv}} \tag{5.1}$$

Approche CTC-MTL standard : MTL (1) La première variante, appelée MTL (1) dans la Fig. 5.2, est l'approche CTC-MTL standard avec deux têtes de sortie : une pour la tâche "caractère" (char) et une pour la tâche auxiliaire (CV).

Approche CTC-MTL hiérarchique : MTL(2) La deuxième variante, appelée MTL (2), gère les deux tâches de manière hiérarchique où l'on considère que les caractères peuvent être catégorisés en CV comme proposé dans l'article (Jimenez et al., 2018) dans le domaine des évènements sonores et ontologies. La prédiction du réseau est réalisée sur les sorties "caractère" au niveau de la couche "char dense" représentée sur la Fig. 5.2. Ensuite, les poids des caractères correspondant aux voyelles et aux consonnes sont respectivement accumulés pour obtenir des probabilités pour 'C' et 'V'. La relation entre les caractères et le niveau CV est réalisée par une simple multiplication matricielle :

$$\hat{\mathbf{z}}_{\text{cv}} = M \cdot \hat{\mathbf{z}}_{\text{char}} \tag{5.2}$$

où $\hat{\mathbf{z}}_{\text{char}}$ et $\hat{\mathbf{z}}_{\text{cv}}$ correspondent aux sorties caractère et CV, avec M une matrice binaire permettant la liaison entre les caractères et les catégories CV. Cette matrice M de dimension $N_{\text{cv}} \times N_{\text{char}}$ est fixe et n'est pas mise à jour pendant l'apprentissage.

Les probabilités Char et CV sont ensuite obtenues en appliquant la fonction softmax. Pour être plus précis, l'implémentation de la fonction CTC utilisée dans ce travail utilise les log-probabilités en appliquant la fonction d'activation log-softmax.

Les catégories CV sont considérées comme une super classe des caractères. Cette variante MLT n'est pas basée sur une ontologie car il n'y a pas de sémantique ontologique entre les caractères et la catégorie CV.

Approche MTL Char+CV-CTC : MTL (3) La troisième variante est la principale contribution novatrice de ce travail. Nous combinons les sorties "caractères" et CV en sommant les sorties des deux tâches. Un Log-softmax est ensuite appliqué après avoir utilisé la transposée de la matrice binaire déterministe M décrite dans la variante (2) comme suit :

$$\hat{\mathbf{z}}'_{\text{char}} = \hat{\mathbf{z}}_{\text{char}} + M^T \cdot \hat{\mathbf{z}}_{\text{cv}} \tag{5.3}$$

La variante (3) est conçue à partir de l'idée que la détection de la tâche auxiliaire CV a un taux élevé de bonne prédiction, et que si cette tâche auxiliaire prédit que le caractère courant est une voyelle, alors nous devons accorder plus de poids sur tous les caractères qui sont liés à la catégorie des voyelles (classe 'V') ; il en est de même pour les consonnes (classe 'C').

5.3 Évaluations, résultats et analyse des systèmes CTC-MLT

Pour cette section, nous nous intéressons à évaluer mes travaux sur la contribution principale de ce chapitre qui réside en l'analyse des différents architectures MTL et juger de l'intérêt de combiner les unités caractères et d'autres unités telles que les catégories consonnes / voyelles.

Nous comparons notre système initial mono-tâche CTC-GRU+TR2 aux trois systèmes MTL décrits dans la section 5.2.2. La constante d'interpolation λ a été optimisée sur l'ensemble de développement Dev93, la meilleure valeur sélectionnée est de $\lambda = 0.8$, nous observons que la catégorisation "caractère" est prépondérante.

TABLE 5.2 – Résultats de nos systèmes CTC-GRU avec les architectures MTL (1), (2) et (3) sur l'ensemble d'évaluation Eval92 en terme de CER et WER. Le terme TR2 signifie qu'on applique la technique de "Time-Reduction" expliqué dans la section 5.1.2.

Model	Greedy Search			WFST-based	
	CVER	CER	WER	CER	WER
TDNN-Phone+TR2	_	_	_	_	6.0
CTC-GRU+TR2	_	8.0	28.8	2.4	6.8
+ MTL (1)	5.2	7.8	27.3	2.3	6.6
+ MTL (2)	5.2	8.3	28.2	2.8	7.5
+ MTL (3)	4.5	7.7	27.5	2.2	6.1

On remarque du tableau 5.2 que le modèle système CTC-multi-tâche standard MTL (1) fonctionne légèrement mieux que le modèle mono-tâche CTC-GRU+TR2. En revanche, la variante hiérarchique MTL (2) inspirée des recherches de (Jimenez et al., 2018) est nettement moins performante avec un écart de 0.8 % en terme de WER avec un décodage WFST. Il semble que placer la tâche auxiliaire directement au-dessus de la tâche principale de prédiction de caractère réduit la performance de la tâche principale, contrairement aux résultats des réseaux basés sur l'ontologie pour la détection des événements sonores rapportés dans Jimenez et al. (2018). Cela est probablement dû au choix de la tâche auxiliaire, il semble que la catégorisation "sémantique" des caractères en CV ne permette pas d'aider l'apprentissage via cette structure hiérachique du MTL.

Les meilleurs résultats sont obtenus avec le système MTL (3) avec 2,2% CER et 6,1% WER. Cela correspond à un gain absolu de 0,7% de WER (10% relatif) par rapport à l'approche mono-tâche CTC-GRU+TR2. La combinaison des deux tâches en sommant leurs sorties "logit" s'est avérée fructueuse. L'ajout de poids entre tache auxiliaire et principale en donnant plus d'importance aux caractères de type voyelle (respectivement consonne) lorsque le symbole de la voyelle (consonne)'V' (respectivement'C') est prédit, est a posteriori pertinent.

5.4 Conclusion

Dans ce travail, nous avons présenté une expérience d'apprentissage multi-tâche (MLT) pour la reconnaissance RAP à base de système CTC "caractères", en utilisant une une tâche auxiliaire pour la reconnaissance des consonnes (C) et des voyelles (V). Trois architectures CTC-MLT sont testées : un modèle standard à deux têtes, un modèle hiérarchique "caractère" et CV, et notre nouvelle approche Char+CV-CTC.

Des expériences d'évaluation menées sur le corpus WSJ avec une modélisation de base de type BiGRU, témoignent de l'intérêt de la technique Char+CV-CTC ; elle obtient les meilleurs résultats avec un CER de 2,2% et un WER de 6,1%. Cette approche a permis de surpasser les performances du système mono-tâche de 0,7% absolu en WER. De manière intéressante, nos expérimentations montrent que l'approche Char+CV-CTC permet d'atteindre les mêmes performances que notre système hybride HMM-DNN phonétique. L'ensemble de ce travail a fait l'objet d'une publication à Interspeech 2019 Heba et al. (2019).

Il serait intéressant de confirmer ces résultats sur des corpus plus grands comme Librispeech. Il serait également intéressant d'explorer d'autres façons de découpage de l'ensemble des caractères, y compris aléatoires, afin d'expliquer l'amélioration obtenue dans le présent travail.

Chapitre 6

Approche End-to-End : unités et modélisation CRDNN

Introduction

Comme indiqué à la section 2.3.4, la contrainte d'alignement monotone entre les séquences d'entrée et de sortie et/ou la production de labels de manière synchrone avec la dimension temporelle sont des propriétés très intéressantes à prendre en compte dans la conception d'un système de RAP. Sous cet angle, l'architecture E2E RNN-Transducer présente un grand avantage par rapport à l'architecture CTC classique et les systèmes basés sur cette approche sont prometteurs. C'est pourquoi j'ai adjoint dans mes travaux, l'étude d'un tel système dont j'ai exploré plusieurs aspects.

— J'ai étudié la modélisation "Convolutional Recurrent Deep Neural Network "CRDNN" et je l'ai introduite dans différentes architectures de systèmes (CTC, RNN-T, seq2seq).
— J'ai proposé l'initialisation de l'encodeur/décodeur du RNN-T via un pré-apprentissage ou via un apprentissage multi-tâche.
— J'ai visité plusieurs unités de prédiction des sorties des systèmes E2E (caractères, bouts de mot, mots, ou ensembles de mots) et j'ai adapté la modélisation CRDNN en conséquence (granularité).

La mise en oeuvre de ces systèmes inclut un travail important en termes d'implémentation. Dans ce cadre, j'ai développé :

— une implémentation open-source “Numba-Transducer” pour le calcul de la fonction de perte sous “Python”.
— la parallélisation GPU du calcul de la fonction de perte du RNN-T pour accélérer l'apprentissage de ce modèle gourmand en mémoire et en calcul.

J'ai réalisé ces travaux et les expérimentations décrites dans ce chapitre au cours de mon stage au Mila. J'ai fait partie de l'équipe de développement du futur outil SpeechBrain [1], fondé entièrement sur PyTorch. Les expérimentations ci-dessous ont été réalisées sur les corpus TIMIT et LibriSpeech. L'ensemble sera reproductible et disponible courant 2021 sur le github de SpeechBrain [2].

Pour décrire l'ensemble de ces contributions, j'ai choisi l'organisation suivante : la 1ère partie rassemble ma démarche pour proposer une modélisation de type CRDNN et la comparaison des différents modèles et architectures E2E sur le corpus TIMIT et LibriSpeech 100 heures. Dans un deuxième temps, je présente mes contributions plus spécifiques sur le système RNN-T, à savoir, l'implémentation parallélisée du calcul de la fonction de perte du RNN-T et l'évaluation des méthodes d'initialisations des systèmes RNN-T. La troisième partie concerne l'exploration de la granularité des unités de prédiction avec notre modélisation CRDNN. La

1. WebSite SpeechBrain : `https://speechbrain.github.io/`
2. Code : `https://github.com/speechbrain/speechbrain`

dernière partie décrit le travail de groupe visant à mettre en place un système RAP E2E sur LibriSpeech 1000 heures.

6.1 La modélisation "Convolutional Recurrent Deep Neural Network "CRDNN"

Les approches RAP E2E avec des réseaux récurrents ont montré des résultats prometteurs ces cinq dernières années (Amodei et al. (2016); Chan et al. (2016); Battenberg et al. (2017)). La modélisation CRDNN pour *Convolutional Recurrent Deep Neural Networks* permet de bénéficier des avantages des deux types de couches :

— convolutionnelles pour l'apprentissage de représentations discriminantes pour la RAP.
— récurrentes pour la modélisation temporelle et l'apprentissage du contexte.

Devenu incontournable, je me suis attaché à construire pas à pas un tel modèle en m'attachant aux performances obtenues à chaque étape, sur le corpus TIMIT. Ce corpus propose une annotation phonétique des prononciations, c'est pourquoi nous avons dans un premier temps étudier des systèmes de reconnaissance phonétique, avant de les étendre à la reconnaissance de parole grand vocabulaire.

Avant de décrire plus précisément les différentes variantes de CRDNN, les architectures impactées et les expériences de validation, il convient d'expliquer le traitement et les augmentations, appliqués aux données audio brut en entrée, pour l'entraînement de tous les modèles présentés ici.

6.1.1 Augmentation de données à la volée sur GPU

Nous proposons dans la mise en place de notre système RAP, une phrase de pré-traitement des données à la volée en entrée du système à chaque passage de 'batch' avant la transformation spectrale ainsi que des perturbations de bruits pour permettre une meilleure robustesse acoustique.

Dans le cadre du projet SpeechBrain, le pré-traitement, à chaque itération durant la phase d'apprentissage, consiste à :

— transférer l'audio brut sur GPU ;
— réaliser une perturbation de vitesse avec un coefficient de (0.9, 1.0, ou 1.1) de manière aléatoire.
— réaliser une approximation de la technique SpecAugment **?**, non pas sur le spectrogramme, mais directement sur le signal audio.

Cette technique, appelée 'Time Domain SpecAugment' (TDSA) [3], a été proposée par mon collègue Peter Plantinga du Mila. Cela se fait de la manière suivante :

— pour la partie fréquentielle :
  — Sélectionner aléatoirement des fréquences et construire le filtre temporel correspondant ;
  — Faire la convolution du filtre avec le signal audio.
— Pour la partie temporelle : des portions du signal sont aléatoirement choisies et enlevées du signal.

Cet ensemble de traitement a été encapsulé dans une couche neuronale et rajouté en amont du modèle CRDNN. De ce fait, les transformations sont réalisées à la volée sur GPU. De la même manière, l'extraction des paramètres acoustiques à partir du signal audio est réalisée à la volée et permet la parallélisation de l'extraction sous GPU pour chaque batch.

Les grand avantages de cette méthode sont de permettre de :

— Réaliser toute la phase d'apprentissage ou de décodage sous GPU ;
— Disposer pour chaque itération, d'une modification différente de chaque observation et ainsi permettre à l'encodeur d'apprendre des représentations plus génériques et robustes.

6.1.2 Les modèles CRDNN dans le cadre d'une architecture CTC

Les modèles CRDNN que j'ai développés utilisent trois types de blocs :

3. TDSA : `https://github.com/speechbrain/speechbrain/blob/master/speechbrain/lobes/augment/tdsa.py`

— Bloc CNN : ce bloc est constitué d'une séquence de couches convolutionnelles construites comme suit : Conv2D avec des kernels de taille (3,3) → LayerNorm → LeakyReLU → MaxPooling sur (dimension temporelle).
— Bloc RNN : ce bloc est constitué d'un réseau récurrent de type Li-GRU (Ravanelli et al. (2018)). Ce type d'architecture remplace la fonction non-linaire Tanh des portes de la cellule GRU par une normalisation et une fonction ReLU. Ce qui permet un meilleur apprentissage comparé au GRU et LSTM.
— Bloc DNN : ce bloc est constitué d'une séquence de couches linéaires construit comme suit : Linear → BatchNorm1D → LeakyReLU → Dropout.

Ces blocs seront disponibles en open-source à la publication officielle du projet SpeechBrain en 2021.

Leur construction s'est faite progressivement, au regard des performances :
— le système CRDNN-CTC B1, considéré comme baseline, intègre un encodeur de 4 couches GRU bidirectionnelles à 1024 neurones par couche ;
— dans la version B2, sont utilisées des cellules Li-GRU proposées par Ravanelli et al. (2018) à la place des GRU : normalisation et activation ReLU se substituent à l'activation standard Tanh des GRU ; il est attendu une meilleure rétropropagation du gradient.
— avec le système CRDNN-CTC B3 apparaissent deux blocs MLP de 512 neurones par couche, avec BatchNorm et activation ReLU.
— Deux blocs CNN sont rajoutés en entrée du système CRDNN-CTC B3 avec un kernel de (3,3) similaire à l'architecture VGG en utilisant 128 filtres pour la première couche CNN et 256 pour la deuxième. Un max pooling 1D de taille 2 est appliqué sur la dimension des paramètres acoustiques ; comme nous le verrons ci apres, ce système CRDNN-CTC B4 s'avère le plus performant ;
— le système CRDNN-CTC B4 prend aux alentours de 3 min 30s de calcul pour chaque epoch. Pour réduire le temps de calcul, nous ajoutons une réduction de dimension temporelle par 2 en appliquant un max pooling sur la dimension de temps à la sortie du bloc CNN. Ce système CRDNN-CTC B5 permet de réduire le temps de calcul à 2 min 40 s par epoch.

Une vue générale de l'architecture complète est donnée en figure 6.1.

6.1.3 Évaluation de l'impact des modèles CRDNN selon l'architecture E2E CTC, RNN-T et seq2seq

Nous proposons d'évaluer l'apport des modèles CRDNN au sein de trois architectures différentes, à savoir CTC, RNN-T et Seq2seq avec mécanisme d'attention sur le corpus TIMIT (voir section 3.2.2). Comme dit précédemment, ce corpus propose une annotation phonétique des prononciations, c'est pourquoi nous développons dans un premier temps des systèmes de reconnaissance phonétique. La métrique de taux d'erreurs de phonèmes (*Phone Error Rate*, PER) est utilisée pour évaluer les systèmes.

Les processus d'extraction de paramètres du signal de la parole, d'augmentation de données, et la perturbation de ces paramètres se font à la volée sur GPU comme expliqué à la section 6.1.1. L'extraction des paramètres acoustiques Filter-Bank à 40 coefficients se fait chaque 20 ms avec un recouvrement de 10 ms, une normalisation CMVN est réalisée en amont.

J'ai utilisé dans tous les cas un optimiseur Adadelta et un clusteur Nvidia DGX de huit V100 à 32Gb (mis à disposition dans le cadre de mon stage au MILA).

Performances sur le corpus TIMIT des systèmes CRDNN-CTC

L'évaluation des systèmes CRDNN-CTC est rapportée dans le tableau 6.1. Dans la deuxième section de ce tableau, sont rapportées les évaluations des systèmes CRDNN-CTC. Le système CRDNN-CTC B1 est loin d'être performant, avec 22,6% de PER, et est grandement amélioré par l'utilisation de cellules Li-GRU dans le système CRDNN-CTC B2. Cette modification permet un gain de 3% en PER, confirme nos attentes et permet de se rapprocher des travaux de Graves tout comme le système CRDNN-CTC B3 qui affiche 18,3%

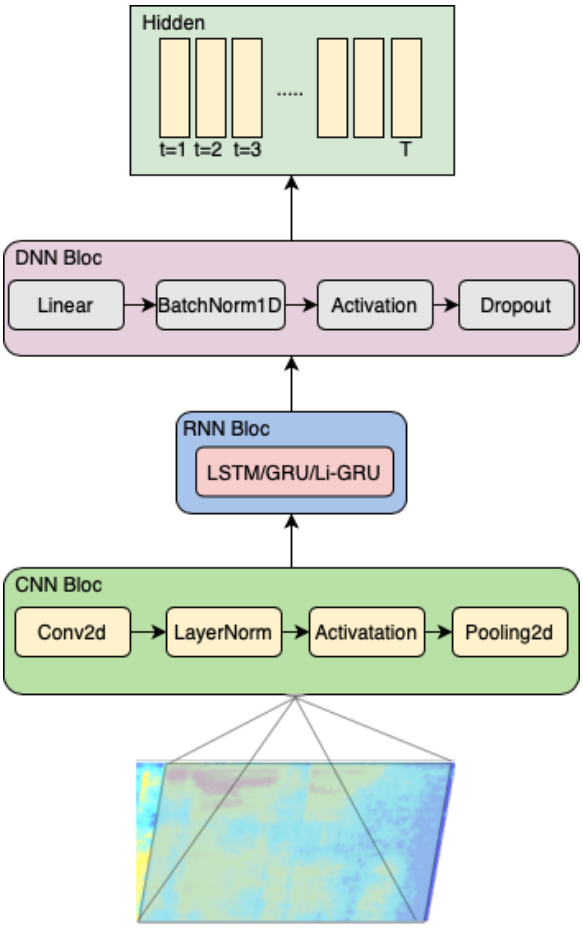


FIGURE 6.1 – Structure des modèles de type CRDNN.

en PER (nouveau gain de 1.5%).

Le système CRDNN-CTC B4 avec 9.2 M de paramètres permet de dépasser l'état de l'art E2E actuel avec une performance de 16.4% PER. Nous avons réalisé plusieurs expérimentations pour réduire le temps de calcul : le système CRDNN-CTC B4 prend aux alentours de 3 min 30 s de calcul pour chaque epoch. En ajoutant la réduction de dimension temporelle par 2 en appliquant un max pooling sur la dimension de temps à la sortie du bloc CNN, le temps de calcul est réduit à 2 min 40 s par epoch pour le système CRDNN-CTC B5 tout en obtenant une meilleure reconnaissance phonétique à 16.1% de PER.

Trois augmentations de données ont été réalisées sur ce dernier système : le *Time Domain SpecAugment* ou TDSpecAugment, comme décrit dans la section précédente. La combinaison du TDSpecAugment avec la perturbation du rythme et le rajout du bruit de fond "babble" (avec un SNR choisi au hasard entre 0 et 15 dB par batch)permet de disposer d'un système CRDNN CTC à l'état de l'art avec une performance de 14.2% de PER similaire au système HMM-DNN phonétique proposée par (Ravanelli et al., 2019).

Modèle	PER	
	Dev	Test
HMM-DNN Pytorch-Kaldi (Ravanelli et al., 2019)	-	14.2
HMM-DNN Li-GRU + fMLLR features (Ravanelli et al., 2019)	-	14.9
Bi-RNN + Attention (Chorowski et al., 2015)	-	17.6
Bi-LSTM + CTC (Graves et al., 2013)	-	17.7
Bi-LSTM + RNN-T (Graves, 2012)	-	23.2
Self-supervised Wav2vec 2.0 (Baevski et al., 2020)	**7.4**	**8.3**
CTC		
B1 : 4l-BiGRU	21.3	22.6
B2 : 4l-Bi-LiGRU	18.1	19.6
B3 : B2 + 2-MLP	16.8	18.3
B4 : CRDNN	14.9	16.4
+ Réduction Temporelle x2	14.5	16.1
+ TDSpecAugment	13.6	15.1
+ Speed perturb	12.9	14.7
+ RIRS perturb	12.5	14.2
RNN-T		
Enc CRDNN + Dec 1l-GRU + Joint 128Dim	12.9	14.6
Enc CRDNN + Dec 1l-Li-GRU (batch norm) + Joint 128Dim	**12.5**	**14.0**
Enc CRDNN + Dec 1l-Li-GRU (layer norm) + Joint 128Dim	**12.3**	**13.8**
Attention + CTC (MTL Attn : 0.8 CTC : 0.2)		
Enc CRDNN + Dec 1l-GRU	12.8	14.3
Enc CRDNN + Dec 1l-LiGRU (layer norm)	12.8	14.5

TABLE 6.1 – Taux d'erreur de phonèmes (PER) avec les techniques CTC, RNN-T, et Attention intégrant des modèles CRDNN. La première section du tableau reprend les meilleurs systèmes à l'état de l'art. Le décodage de tous les modèles est réalisé avec l'algorithme *Greedy Search*.

Performances sur le corpus TIMIT des systèmes "état de l'art"

À titre de comparaison, sont rapportés dans la première section du tableau 6.1 les meilleurs résultats obtenus par des systèmes de reconnaissance phonétique, en termes de PER sur le corpus TIMIT, durant les cinq dernières années. On remarque que seule l'architecture HMM-DNN de l'état de l'art arrive à un taux PER très faible, de 14.2 %, comparée aux approches E2E dont les PER sont aux alentours de 17%.

Cependant, l'architecture proposée par (Baevski et al., 2020) consiste à utiliser les vecteurs de représentation appris avec la technique "self-supervised" (auto-supervisé) sur le corpus de parole LibriSpeech 1000 heures. Ces représentations sont utilisées à la place des MFCC pour entraîner un réseau de neurones de 7 couches CNN avec 512 kernels avec une dimension de filtre de 10, 3, 3, 3, 3, 2, 2 pour les couches 1 à 7 (avec l'outil wav2letter++ Pratap et al. (2018)). Un nouvel état de l'art pour la reconnaissance phonétique à 8.3% de PER sur TIMIT a été très récemment proposé avec cette technique.

Intégration du CRDNN dans des architectures RNN-T et seq2seq avec attention+CTC

J'ai repris le modèle CRDNN proposé avec l'architecture CTC pour définir l'encodeur de systèmes RNN-T.

Le premier décodeur évalué comporte une seule couche GRU à 128 neurones, et le composant de jointure est une somme des deux sorties avec une fonction d'activation LeakyReLu. Le résultat de ce RNN-T est nettement supérieur à celui du CTC avec 14.6% de PER.

Un deuxième décodeur avec une architecture Li-GRU avec une normalisation Batch norm, permet un meilleur apprentissage du composant de décodage. Il permet d'augmenter la performance avec un PER 14.0%.

Une analyse et un suivi de l'évolution du décodage greedy search montrent que seules quelques hypothèses des phrases dans le batch sont étendues à un moment t donné, avec un nouveau label prédit, lors de la phase d'apprentissage. Le RNN-T avec batch-norm fonctionne bien, par contre, la normalisation n'a pas de sens dans la phase de décodage où l'on veut avoir différentes extensions de chaque phrase à chaque moment t. Nous proposons d'utiliser une "Layer-norm" à la place de la batch-norm dans le Li-GRU. Une layer-norm normalise les activations à chaque pas de temps. Cette modification nous permet de disposer d'un système RNN-T à 13.8% de PER qui correspond à un nouveau état de l'art pour le corpus TIMIT sans self-supervised training.

Nous mettons à titre de comparaison le résultat du système seq2seq avec attention+CTC développé par mon collègue Ju-Chieh Chou au sein du Mila. Ce système que nous appelons par la suite le système "Attention+CTC", contient un encodeur CRDNN avec une sortie CTC pour aider l'encodeur à apprendre l'alignement monotone. Le mécanisme d'attention utilisé est une attention locale à 256 neurones. Le multi-tâche attention+CTC est activé pour les 10 premières epochs pour permettre une bonne initialisation de l'encodeur, puis le CTC est désactivé pour laisser uniquement une optimisation entropie croisée CE. On remarque qu'un décodeur GRU ou Li-GRU à une couche de 256 neurones ne permet pas d'obtenir une performance intéressante par rapport aux systèmes CTC et RNN-T. Cette mauvaise performance peut probablement expliquée par la trop petite quantité de données du corpus TIMIT (3 heures).

6.1.4 Evaluation de l'approche CRDNN sur le corpus LibriSpeech 100 heures

J'aborde dans cette section une vue globale de nos expérimentations sur le corpus Librispeech train-clean-100 avec des modélisations CRDNN-CTC et RNN-T développées par moi même et les résultats des systèmes Attention+CTC réalisés par mon collègue Ju-Chieh Chou (avec ma contribution sur les unités subword BPE). J'utilise un décodage greedy search pour la modélisation CTC et RNN-T et un décodage beam search sur les systèmes Attention+CTC ; l'algorithme "beam search" sera prochainement implémenté pour le système RNN-T.

Modèle	Décodage	WER	
		Dev	Test
LAS (Irie et al., 2019)	BS	11.6	12.0
CTC	GS	16.2	16.8
RNN-T	GS	12.6	12.9
Attention+CTC	GS	11.8	12.1
	BS	**11.3**	**11.8**

TABLE 6.2 – Comparaison des différentes systèmes E2E caractères entraînées sur le corpus Librispeech train-clean 100 heures et évaluées sur le corpus test-clean. Le décodage avec les algorithmes greedy search ou beam search (noté respectivement par GS et BS) est précisé.

En observant le tableau 6.2, on remarque que le système RAP CTC est très loin en terme de performance avec 16.8% de WER, comparé aux systèmes RNN-T et Attention+CTC . Cela explique la difficulté qu'a le modèle CRDNN à apprendre implicitement la modélisation du langage. Les architectures RNN-T et Attention+CTC permettent un apprentissage explicite du langage, et cet apprentissage se traduit par un gap d'environ 4 % en termes de WER entre l'approche CTC et les approches modélisant la dépendance linguistique (RNN-T et seq2seq). Le RNN-T avec son décodeur auto-regressif permet d'obtenir un taux d'erreur de 12.9% WER sur l'ensemble de test clean avec un décodage GS. Le système Attention+CTC proposé par mon collègue Ju-Chieh Chou au Mila obtient la meilleure performance de 11.8% de WER, avec

un décodage BS et un beam de 40. Ce résultat est très proche de celui obtenu par le système Listen, Attend and Spell, développé par les collègues d'Aachen ((Irie et al., 2019)).

6.2 Choix des unités de prédiction

L'un des verrous dans la conception des systèmes RAP E2E réside dans le choix de l'unité de modélisation. Le choix de ces unités ainsi que la façon avec laquelle le système RAP combine ces unités pour représenter les mots est un axe important à explorer. Dans les systèmes discutés ci-dessus, sont privilégiées les unités de type "caractères" ou "phonèmes" car pratiques et faciles à mettre en oeuvre.

Néanmoins, différentes unités de modélisation peuvent s'appliquer selon les cas d'usage, tel que les systèmes large vocabulaire, les systèmes à données réduites (en anglais *low ressource setting*), les systèmes multilingues, etc. Plusieurs unités existent, comme par exemple, le phonème, le bi-phone, le tri-phone, la syllabe, le caractère, le "bout de mot" et le mot. Chacune d'elles a ses avantages et inconvénients. Les critères suivants sont essentiels pour la choisir :

— L'unité doit être représentée par une réalisation acoustique dans différents contextes.
— L'unité doit être disponible en nombre important dans le corpus d'apprentissage pour disposer d'une bonne estimation statistique de ses caractéristiques.
— L'unité doit être générique pour pouvoir former tout mot.

En se fondant sur cet ensemble de critères, j'ai décidé d'explorer trois types de tokenisation : 1) caractère, 2) bout de mot dont nous gardons l'appellation anglaise *subword units*, 3) les unités bout de mot composées de bout de deux mots successifs dont nous gardons l'appellation anglaise *crossword units* ; par exemple, *kind_of, sort_of* sont considérés des unités.

6.2.1 Génération des unités Subword/Crossword avec l'algorithme Byte Pair Encoding

Pour la création des unités subword/crossword, nous nous basons sur un algorithme de tokenisation non-supervisé appelé *Byte-Pair Encoding* (BPE) (Shibata et al. (1999)), qui est très utilisé dans le domaine de la traduction automatique (Sennrich et al. (2015)). L'outil SentencePiece [4] de Google propose une implémentation de cet algorithme.

L'algorithme BPE est un algorithme de tokenisation itératif non-supervisé qui permet à partir d'un corpus textuel de compresser itérativement les mots peu fréquents par un découpage en bouts de mot les plus fréquemment utilisés à l'itération précédente.

La création d'unités Subword est réalisée avec l'algorithme BPE en initialisant l'ensemble des unités par tous les mots séparés par un espace. Itérativement, l'algorithme BPE permet de découper les mots moins fréquents en bout de mot, pour réduire l'ensemble d'unités à N unités fixé au départ. Il est tout à fait possible, avec cet algorithme de trouver un mot complet parmi les unités si sa fréquence dans le corpus textuel est élevée.

La création d'unité Crossword consiste à appliquer l'algorithme BPE sur le corpus textuel en ne considérant plus l'espace comme étant un séparateur entre les mots. De ce fait l'initialisation de l'algorithme BPE considère que chaque phrase est une unité possible. Et itérativement, l'algorithme du BPE découpe ces unités en maximisant la fréquence d'apparition des bouts de mots, mots, ou composition de deux à trois mots. Ce type d'unité peut proposer par exemple la suite "il est" comme étant une unité crossword.

La différence entre une génération d'unités subword versus crossword consiste à prendre en compte la séparation par un espace ou non des mots du corpus textuel fourni à l'algorithme BPE. L'option *split_by_whitespace* disponible dans l'outil SentencePiece permet cette génération.

À des fins de comparaison, le tableau 6.3 présente un exemple de séquences de sortie du corpus LibriSpeech avec une décomposition subword et crossword de 1000 unités.

4. SentencePiece Github : `https://github.com/google/sentencepiece`

Tokenisation	Phrase
Original	IT WAS ONE OF THE MASTERLY AND CHARMING STORIES OF DUMAS THE ELDER
Caractère	I T _ W A S _ O N E _ O F _ T H E _ M A S T E R L Y _ A N D _ C H A R M I N G _ S T O R I E S _ O F _ D U M A S _ T H E _ E L D E R
Subword 1k	_IT _WAS _ONE _OF _THE _MASTER LY _AND _CHA R M ING _ST OR IES _OF _DU MA S _THE _E LD ER
Crossword 1k	_IT_WAS _ONE _OF_THE _MASTER LY _AND _CHA R M ING _STOR IES _OF _DU MA S _THE _EL D ER

TABLE 6.3 – Exemples d'une phrase issue du corpus Librispeech découpée en caractères, BPE subword et BPE crossword à 1000 unités. Le caractère '_' représente un espace et est inclus dans un bout de mot.

Des améliorations de l'algorithme BPE sont proposées dans la littérature notamment par (Kudo (2018); Provilkov et al. (2019)) qui utilise des techniques de régularisation telles que la prise en compte de multiples candidats de découpage : par exemple, la suite de mots *Hello world* peut s'écrire : {*_Hell/o/_world*}, {*_H/ello/_world*}, {*_He/llo/_world*}. D'autres types de régularisation telles que l'utilisation d'un modèle de langage unigramme ou l'application de dropout permettent d'améliorer grandement les systèmes de traduction automatique. Nous utilisons dans nos expérimentations une régularisation utilisant un modèle de langage unigramme.

6.2.2 Évaluation comparée des unités de prédiction : caractère, subword, crossword avec le système CRDNN Attention+CTC

Je présente une évaluation des unités de sortie sur le corpus LibriSpeech. Nous utilisons dans cette évaluation des systèmes encodeur-décodeur avec mécanisme d'attention et un décodage avec l'algorithme beam search. Ce choix a été réalisé à des fins de comparaison avec les systèmes avec mécanisme d'attention existant dans l'état de l'art et c'est l'architecture qui donne les meilleurs résultats jusqu'à présent.

Nos évaluations sont réalisées avec le sous-ensemble de 100 heures du corpus LibriSpeech pour la partie apprentissage, l'ensemble de développement (dev clean) est utilisé pour régler les hyper-paramètres du modèle. Le corpus de test (test clean) est utilisé pour la comparaison des performances en terme de WER. Le choix d'utiliser l'ensemble de 100 heures est dû au fait que la puissance de calcul des GPU est limitée, un apprentissage sur l'ensemble des 1000 heures prend environ 24 jours de calcul sur 1 GPU Nvidia V100 32Gb. Nous utilisons pour notre système l'augmentation de données à la volée (voir section 6.1.1), 40 coefficients F-BANK extraits chaque 20 ms avec un recouvrement de 10 ms. Nous utilisons l'outil SentencePiece [5] de Google pour la génération des sorties Subword et Crossword, avec une régularisation BPE et un modèle de langage unigramme ;nous imposons une longueur maximale de 5 caractères par unité.

Première série d'expériences avec les unités "subwords" Ce premier volet expérimental a pour but d'identifier la meilleure architecture d'un système CRDNN Attention+CTC, pour la prédiction de 1000 graphèmes (que ce soit les Subword ou Crossword). Notre architecture contient un encodeur avec 2 blocs convolutionnels de (128,256) filtres de (3x3), 4 couches récurrentes de 1024 neurones et 2 blocs MLP de 512 neurones. Le décodeur est modélisé par 1 ou 2 couches GRU de 1024 neurones et une attention locale avec un vecteur de 1024 neurones est utilisée.

5. SentencePiece github : `https://github.com/google/sentencepiece`

Le tableau 6.4 représente une évaluation du choix de la modélisation récurrente (RNN, GRU, LSTM, Li-GRU) de notre encodeur CRDNN. Nous avons fait pareil pour le décodeur, aucune différence significative n'a été observée en modifiant l'architecture du décodeur. De ce fait, nous présentons uniquement la différence de performance lié au nombre de couches utilisées pour le décodeur.

Modèle	WER	
	Dev	Test
Attention + CTC (MTL Attn : 0.8 CTC : 0.2)		
Enc CRDNN (RNN) + Dec 1l-GRU	15.8	16.4
Enc CRDNN (GRU) + Dec 1l-GRU	15.8	15.6
Enc CRDNN (Li-GRU) + Dec 1l-GRU	13.8	14.4
Enc CRDNN (LSTM) + Dec 1l-GRU	**12.6**	**13.1**
Enc CRDNN (LSTM) + Dec 2l-GRU	13.2	13.9

TABLE 6.4 – Évaluation du choix du type de modélisation récurrente (RNN, GRU, Li-GRU, LSTM) de l'encodeur CRDNN pour la reconnaissance d'unités de type subwords de 1000 unités pour un système CRDNN Attention+CTC entrainé sur Librispeech 100 heures.

Aucune des architectures récursives RNN, GRU, Li-GRU utilisées pour notre encodeur CRDNN n'a permis d'obtenir une bonne performance en utilisant les unités subword comme sortie. l'architecture LSTM quant à elle avec sa modélisation à 4 portes semble nettement mieux fonctionner, la porte dite 'output gate' semble avoir un grand impact sur l'apprentissage du lien entre une séquence de sortie courte (à base de subword) et la longue séquence d'observation. Nous remarquons aussi que l'utilisation d'un décodeur à 1 couche est plus performant qu'un décodeur à 2 couches.

Pour la suite, nous utiliserons le modèle ayant la meilleure performance, à savoir : un système Attention+CTC avec un encodeur CRDNN (LSTM) et un décodeur GRU à une couche.

Influence du choix des unités pour un système CRDNN Attention+CTC . Le tableau 6.5 rassemble les performances de nos systèmes Attention+CTC avec une modélisation CRDNN LSTM et celles des approches utilisant l'architecture LAS (Lüscher et al., 2019; Irie et al., 2019) ayant des sorties caractères ou bouts de mots (Subword). On retrouve les résultats déjà présentés dans le paragraphe précédent correspondant aux sorties de type caractère (tableau 6.2).

La première section du tableau 6.5 correspond aux travaux des chercheurs de Aachen sur les systèmes encodeur-décodeur avec attention. L'approche LAS différente de l'approche classique d'encodeur-décodeur a été proposée par (Irie et al., 2019) ; la différence est dans l'encodeur pyramidal avec une réduction de la séquence temporelle par deux à chaque couche récurrente de l'encodeur. Elle permet d'obtenir un meilleur score avec les mêmes sorties subword 16K (disponible sur l'outil RETURN [6] de Aachen). Dans les deux papiers de référence (Lüscher et al., 2019) et (Irie et al., 2019), est proposée à la fois l'évaluation sur l'ensemble des 100 heures ainsi que sur l'ensemble total des 1000 heures ; ce dernier est à l'état de l'art des systèmes RAP actuels. Seuls les résultats avec les 100 heures sont rapportés dans le tableau ; cette performance observée de 12.9% WER est le meilleur score qu'on peut retrouver dans la littérature pour cet ensemble de 100 heures d'apprentissage. Cette différence en performance est due à une meilleure prise en compte du contexte par l'encodeur du LAS.

Après analyse des différences entre notre système Attention+CTC avec un encodeur standard et l'architecture de l'encodeur du système LAS, nous avons ajouté à notre système une réduction temporelle d'un facteur 2 et 4 à la sortie du bloc CNN (pooling "aggressif") et ainsi permis de prendre en compte un plus large contexte par l'architecture LSTM pour la prédiction des unités Subword/Crossword, unités représentant un grand ensemble de caractères à prédire à la fois.

Le système CRDNN Attention+CTC **caractère** avec une réduction temporelle par deux permet de dépasser l'état de l'art actuel avec une performance de 11.8% WER. Cependant, les performances sont

6. RETURNN github : `https://github.com/rwth-i6/returnn`

Modèle	Unité		WER	
	Type	#	Dev	Test
E2E Attention (Lüscher et al., 2019)	Subword	16K	14.7	14.7
E2E LAS Attention (Irie et al., 2019)	Subword	16K	12.7	12.9
	Phoneme	-	13.8	14.3
	Char	-	11.6	12.0
CRDNN Attn+CTC (Réduc. Temp x2) (MTL Attn : 0.8, CTC : 0.2)	Char	-	11.3	11.8
	Subword	1K	12.6	13.1
	Crossword	1K	13.2	13.8
CRDNN Attn+CTC (Réduc. Temp x4) (MTL Attn : 0.8, CTC : 0.2)	char	-	11.8	12.2
	Subword	1K	**9.1**	**9.7**
		2k	9.8	10.3
		5K	11.2	11.9
	Crossword	1K	9.8	10.3
		2K	10.3	10.8
		5K	12.7	13.1

TABLE 6.5 – Résultats en termes de WER des systèmes encoder-décodeur avec attention entraînés sur le corpus LibriSpeech 100 h et évalués sur le corpus LibriSpeech test-clean.

dégradées si l'on applique une réduction plus agressive (facteur 4) sur la dimension de temporelle (prédiction caractère). Cela peut s'expliquer par le fait de vouloir modéliser une petite unité avec un très grand contexte acoustique.

Notre système CRDNN Attention+CTC avec réduction temporelle facteur 4, donne de meilleures performances avec un ensemble de 1000 unités de sortie de type subword et crossword que les systèmes avec caractères. La réduction temporelle forte semble donc adéquate pour la granularité subword/crossword. Une performance de 9.7% de WER est obtenue avec le système appris sur des subwords à 1000 sorties et un décodage beam search avec un seuil de beam de 80. Cette performance représente un nouvel état de l'art sur ces données. On observe cependant une dégradation des performances si l'on augmente trop le nombre d'unités ; ici sont testés 2000 et 5000 unités de prédiction avec les types subword/crossword.

6.3 Modélisation RNN-T : parallélisation des calculs et initialisation du modèle

Le modèle RNN-T permet de prendre en considération la dépendance acoustique et linguistique et permet de combiner ces deux informations pour la prédiction de la séquence de sortie. Au cours de mon stage au Mila, j'ai été chargé d'implémenter l'architecture RNN-T, en vue de l'intégrer au projet SpeechBrain. J'ai alors pris conscience de trois grands verrous liés à l'apprentissage de ce modèle :

— la réduction de la consommation mémoire des RNN-T.
— L'optimisation du calcul de la fonction de perte des RNN-T.
— L'initialisation des RNN-T par un modèle CTC pour l'encodeur et par un modèle de langage pour le décodeur.

J'ai choisi lors du développement du modèle RNN-T de focaliser mon travail sur les deux derniers points : 1) la parallélisation GPU du calcul de la fonction de perte (le calcul de l'algorithme forward-backward et du gradient associé n'est pas faisable sans parallélisation), 2) l'évaluation de plusieurs techniques d'initialisation des RNN-T pour un apprentissage stable et efficace.

6.3.1 Parallélisation GPU du calcul de la fonction de perte du RNN-T

La modélisation RNN-T bénéficie d'un composant dédié à l'apprentissage acoustique (l'encodeur) appelé aussi *Transcription network* et d'un deuxième composant dédié à l'apprentissage linguistique (le décodeur) appelé *Prediction network* (voir section 2.3.4). Ces composants sont par la suite fusionnés en gardant la dimension temporelle de l'information acoustique et linguistique pour prédire la séquence de sortie.

Le calcul de la fonction de perte se fait par le biais de l'algorithme forward-backward qui permet de calculer tous les chemins possibles pour la réalisation de la séquence de sortie. Le gradient est ainsi calculé à partir des matrices α et β pour estimer les paramètres du modèle RNN-T. L'ensemble des équations et calculs ont été donnés dans la section 2.3.4, voir en particulier les équations 2.26 et 2.29).

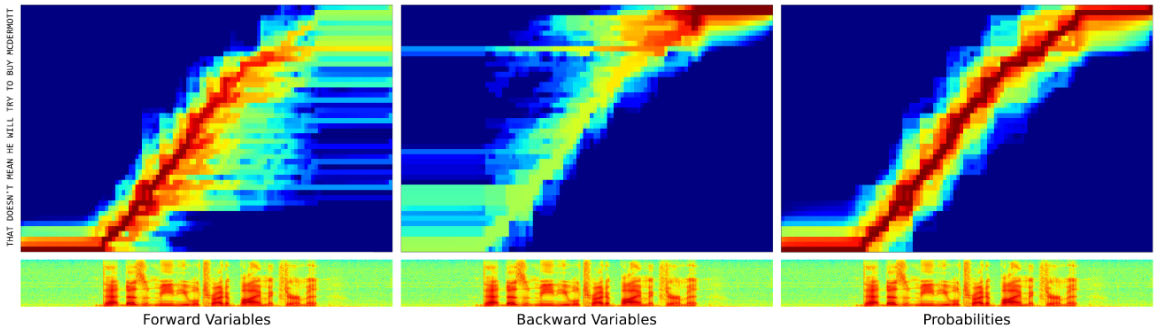


FIGURE 6.2 – Représentation des sorties de l'algorithme forward-backward durant la phase d'apprentissage pour une même séquence audio. Les images du bas représentent le spectrogramme de la phrase (l'axe abcisse correspond à l'échelle de temps signal). La phrase est transcrite en caractères, repris à gauche des images (l'axe vertical est l'échelle caractère). Les images de gauche à droite représentent les matrices α, β du calcul de l'algorithme forward-backward et leur produit.

La figure 6.2 reprise de (Graves, 2012) représente (de gauche à droite) les matrices α, β et leur combinaison sur la prononciation : "THAT DOESN'T MEAN HE WILL TRY TO BUY MCDERMOTT". On remarque que les chemins les plus pertinents (couleurs chaudes sur la diagonale) sont bien appris par le réseau de neurones via l'algorithme du forward-backward.

J'ai proposé dans le cadre du projet SpeechBrain d'implémenter une version parallélisée de cet algorithme sous GPU en Python avec la librairie Numba [7] (librairie qui permet la compilation 'Just In Time (JIT) [8]' et la parallélisation GPU des matrices Numpy et Pytorch). À ce jour, aucun des outils de deeplearning (PyTorch, Tensorflow) ne permettent d'offrir une implémentation optimisée (sous Cuda GPU) de la fonction de perte du RNN-T et de son gradient, et très peu d'implémentations existent en open-source. 'Warp-Transducer' [9] disponible en open-source et développé en C++ offre une implémentation parallélisée avec une liaison Python mais l'intégration des librairies C++ peut être très difficile à maintenir. Notre implémentation de la fonction de perte parallélisée sous GPU sur Python et compatible avec PyTorch permet à la communauté SpeechBrain de disposer d'une implémentation "Python friendly", open-source et facile à intégrer.

La figure 6.3 représente l'évolution de l'exécution parallélisée (sur la dimension verticale) du calcul de la matrice α de l'algorithme forward-backward. Pour ce faire, un système de verrou et de synchronisation des threads GPU a été mis en place pour le calcul de la matrice α. Les éléments en vert représentent les éléments déjà calculés, les éléments en rose représentent les éléments qui peuvent être calculés en parallèle. La même parallélisation a été effectuée sur la matrice β et le calcul du gradient associé. Le pseudo-code 2 des équations 2.29 parallélisées est disponible ci-dessous.

7. Numba : `http://numba.pydata.org/`
8. JIT : `https://www.freecodecamp.org/news/just-in-time-compilation-explained/`
9. Warp-Transducer : `https://github.com/HawkAaron/warp-transducer`

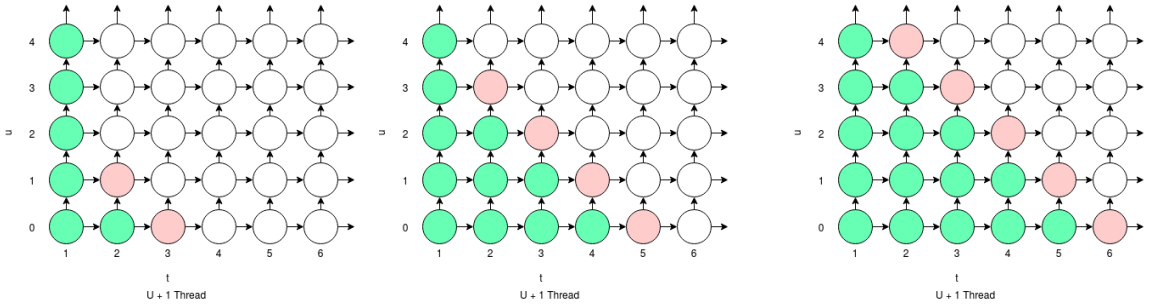


FIGURE 6.3 – Schéma de l'évolution du calcul de la matrice α de l'algorithme forward-backward en parallèle sur GPU avec $(U+1)$ threads (U étant la longueur de la séquence du décodeur). Chaque thread u est responsable du calcul de la dépendance de chaque élément du vecteur $\alpha[:,u]$. La dépendance temporelle est assurée à l'intérieur d'un même thread sur toute la séquence temporelle. La dépendance entre deux threads $u+1$ et u est maintenue avec un système de verrou défini dans le pseudo code 2 et est représentée par l'évolution du calcul de la matrice α de gauche à droite.

L'outil Numba permet d'accéder aux blocs et threads disponibles de la GPU (la notion de bloc et thread sur GPU est utilisée pour découper les processus en deux types de threads : ceux qui partagent une mémoire locale (les threads au sein d'un même bloc), et ceux qui partagent la mémoire globale (les processus bloc) ; nous utilisons les annotations $cuda.blockIdx.x$ et $cuda.threadIdx.x$ pour représenter respectivement le ième exemple du batch et la position u de la séquence de sortie dans la matrice $alpha$.

L'algorithme 2 ci-dessous présente la parallélisation sous GPU de la matrice α de l'algorithme forward backward. Si B correspond au nombre de prononciations par batch, et U correspondant à la longueur de la séquence de décodage, $B+(U+1)$ processus permettent de paralléliser le calcul pour chaque prononciation au sein du batch et pour chaque label au sein de la séquence de sortie. Un mécanisme de synchronisation 'lock' est utilisé pour assurer la dépendance du calcul des éléments par rapport aux valeurs précédentes.

Supposons que B, max_T, max_U, K représentent respectivement le nombre de phrases traitées par batch, la longueur la plus longue des séquences d'entrée, la longueur $U+1$ de la séquence de sortie la plus longue, et la taille de l'ensemble des prédictions possible. supposons que l'on dispose d'une matrice de vraisemblance $P(k|b,t,u)$ (sorties du RNN-T) à 4 dimensions (B, max_T, max_U, K), et d'une matrice $labels$ à 2 dimensions (B, max_U). Le calcul parallélisé sous GPU de la matrice α de dimension (B, max_T, max_U) se fait avec $(B * max\_U)$ processus en parallèle et se base sur un vecteur entier (lock) de dimension (B, max_U) fonctionnant comme un verrou sur la matrice α. Le fonctionnement de ce verrou est représenté dans la figure 6.3 et permet au thread u d'incrémenter par -1 la case $lock[b, u+1]$ à chacune de ses exécutions sur la dimension temporelle. Cette technique de verrou permet de paralléliser le calcul de la matrice α tout en respectant la double dépendance du calcul des éléments $\alpha[b,t,u]$ par rapport à $\alpha[b,t-1,u]$ et $\alpha[b,t,u-1]$.

Algorithm 2 Calcul parallélisé des valeurs de la variable forward de l'algorithme forward-backward avec le système de verrou représenté dans la figure 6.3.

$b \leftarrow$ cuda.blockIdx.x
$u \leftarrow$ cuda.threadIdx.x
$t \leftarrow 0$
$lock[b, u] \leftarrow 0$
while $t < T[b]$ **do**
  if $u = 0$ **then**
    if $t > 0$ **then**
      $alpha[b, t, 0] \leftarrow alpha[b, t-1, 0] + p(b, t-1, 0, blank)$

      $lock[b, u+1] \leftarrow -1$ # autoriser le processus (b,u+1) à s'executer au temps t
      $t \leftarrow t + 1$
    end if
  else
    if $lock[b, u] < 0$ **then** # si la dépendance temporelle est satisfaite entre (b,u) et (b,u+1)
      if $t = 0$ **then**
        $alpha[b, 0, u] \leftarrow alpha[b, 0, u-1] + p[b, 0, u-1, labels[b, u-1]]$
      else
        $emit \leftarrow alpha[b, t, u-1] + p[b, t, u-1, labels[b, u-1]$
        $no\_emit \leftarrow alpha[b, t-1, u] + p[b, t-1, u, blank]$
        $alpha[b, t, u] =$ log_sum_exp(emit, no_emit)
      end if
      if $u < U[b]$ **then**
        $lock[b, u+1] \leftarrow lock[b, u+1] - 1$ # autoriser le processus (b,u+1) à s'executer au temps t
      end if
      $lock[b, u] \leftarrow lock[b, u] + 1$ # verrouiller le lock du thread (b, u) au temps t pour assurer la dépendance avec $(u-1)$
      $t \leftarrow t + 1$
    end if
  end if
end while

Comparons maintenant le temps de calcul de mon implémentation et celle de l'outil *Warp-Transducer*. Une première évaluation du temps de calcul de la fonction de perte du RNN-T porte sur différentes tailles de batch, $N = [1, 16, 32]$, avec des exemples de phrases d'une durée moyenne de 6 secondes (560 fenêtres de 20 ms avec un recouvrement de 10 ms).

Le tableau 6.6 présente cette comparaison avec les implémentations suivantes : 1) l'implémentation en python natif, 2) la version open-source C++ *Warp-Transducer*, 3) mon implémentation *Numba-Transducer* en Python.

On observe que mon implémentation Numba-transducer est 30% plus rapide que l'implémentation C++ Warp-transducer. La librairie Numba avec son compilateur 'Just-In-Time' permet de joindre efficacement la compilation de l'algorithme parallélisé sous CUDA avec les matrices (tensors) de l'outil Pytorch.

Nous nous intéressons par la suite à la comparaison en termes de temps de calcul et consommation maximale de la mémoire GPU par *epoch* de mon implémentation par rapport aux architectures standard E2E et de fonction de perte utilisée (CTC et encodeur-décodeur avec attention).

Nous pouvons observer dans le tableau 6.7 que mon implémentation est nettement plus rapide que les fonctions de perte CTC et entropie croisée, disponibles sur CuDNN. Mon code sera disponible en open-source

Matrice $P(k\|t,u)$	Python	Warp-transducer	Numba-transducer
T=560, U=65, A=40			
N = 1	10 sec	1.15 ms	0.84 ms
N = 16	2 min 32 sec	2.81 ms	2.09 ms
N = 32	5 min 20 sec	4.97 ms	2.74 ms

TABLE 6.6 – Comparaison des temps de calculs de la fonction de perte RNN-T avec l'implémentation C++ de Warp-transducer et mon implémentation Numba-transducer en python avec une carte graphique Nvidia V100 32Gb.

sur le github du projet SpeechBrain courant 2021 [10].

Modèle	Loss Implem.	# params (M)	GPU-Ram (Gb)	time / epoch (sec)
CTC	CUDNN	9.3	6	170
Attn + CTC	CUDNN	9,9	10	214
RNN-T	Numba-Transducer	9.8	13	137

TABLE 6.7 – Comparaison des temps de calculs des modèles CTC, Attention + CTC, RNN-T sur le corpus TIMIT. Pour chaque modèle, est renseignée l'implémentation de la fonction de perte (pure CUDA ou autre) correspondante, ainsi que le nombre de paramètres des modèles. La comparaison a été réalisée sur une carte graphique Nvidia V100 32Gb.

6.3.2 Initialisation du modèle RNN-T

L'initialisation du RNN-T est une étape cruciale pour que l'apprentissage se passe bien. Le composant de jointure du RNN-T entre les sorties de l'encodeur et du décodeur est une partie clé du RNN-T, la rétro-propagation du gradient sur le réseau RNN-T via ce composant de jointure est essentielle pour un apprentissage de l'alignement des deux séquences (entre encodeur et décodeur). Il est parfois très difficile de trouver une modélisation de l'encodeur et du décodeur qui permettent d'avoir durant les premières itérations de l'apprentissage un bon alignement des données d'apprentissage. Par exemple, un encodeur avec des réductions de dimension (*pooling* ou sous- échantillonnage) agressives complexifie grandement la tâche. Ce problème a déjà été rencontré avec les modèles seq2seq avec attention, où le vecteur d'attention est initialisé aléatoirement par exemple. L'une des solutions apportées par la communauté pour ces modèles a été d'ajouter un apprentissage avec alignement monotone tel que le CTC en sortie de l'encodeur et ainsi de faire un apprentissage multitâche CTC et Attention (Kim et al., 2017b). Généralement, les systèmes RNN-T sont donc initialisés à partir de :

— Un encodeur pré-entrainé avec la fonction CTC.
— Un décodeur pré-entrainé comme un modèle de langage.

Cette initialisation du RNN-T peut être coûteuse en temps et nous proposons dans nos expérimentations d'explorer l'initialisation du RNN-T par :

1. Initialisation des modèles (encodeur et décodeur) avec le pré-apprentissage.
2. Utilisation des approches multi-taches pour une meilleure initialisation des composants (encodeur et décodeur).

L'outil PyTorch nous permet de disposer de compilation du modèle neuronal en mode dynamique dans le temps ; il est tout à fait possible d'activer un apprentissage multi-tâche : 1) RNN-T + CTC pour l'encodeur, 2)

10. Numba-transducer : `https://github.com/speechbrain/speechbrain/blob/master/speechbrain/nnet/transducer/transducer_loss.py`

RNN-T + apprentissage du décodeur par entropie croisée CE (*Cross Entropy*), 3) ou même un apprentissage RNN-T + CTC pour l'encodeur + CE pour le décodeur comme représenté dans la figure 6.4 durant les premières itérations de l'apprentissage, puis de désactiver les tâches auxiliaires (à savoir le CTC et le CE).

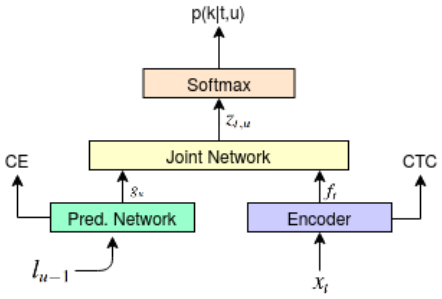


FIGURE 6.4 – Initialisation du modèle RNN-T avec une approche multi-tâche, trois tâches sont apprises en simultané : 1) un apprentissage CTC pour l'encodeur, 2) un apprentissage d'entropie croisée (CE) pour le décodeur, 3) un apprentissage RNN-T du joint-network.

Je présente maintenant une évaluation des différentes manières d'initialiser un modèle RNN-T. J'utilise pour cette évaluation un encodeur CRDNN et un décodeur Li-GRU similaire à celui utilisé dans ma comparaison des différentss architectures sur le corpus LibriSpeech 100 heures 6.2.

Le tableau 6.8 présente une évaluation de l'initialisation du modèle RNN-T avec pré-apprentissage, et de l'intérêt d'utiliser un apprentissage multi-tâche (multi-objectif). Nous pouvons remarquer que l'initialisation du modèle RNN-T par un encodeur pré-entraîné avec la fonction CTC nous permet d'avoir une amélioration globale de l'ordre de 0.3% en terme de WER. De plus, si j'ajoute une initialisation du décodeur avec un modèle de langage entraîné avec la fonction de perte CE sur les mêmes données (100 heures d'apprentissage), une amélioration de 0.4% WER est observée.

J'ai expérimenté par la suite la possibilité de renforcer l'apprentissage du modèle RNN-T avec une approche d'apprentissage multi-tâche. Pour ce faire, j'ai utilisé la simplicité des graphes dynamiques que PyTorch offre pour activer l'apprentissage CTC + RNN-T (CTC pour l'encodeur) ou CTC + CE + RNN-T (CTC pour l'encodeur, CE pour le décodeur) et RNN-T pour l'ensemble des composants durant les 10 premières *epochs* pour disposer d'un bon alignement ; puis je désactive les tâches auxiliaires, à savoir CTC et CE. On remarque qu'on obtient des performances similaires aux méthodes d'initialisation utilisant un modèle pré-entraîné.

Modèle	# Epoch	WER	
		Dev	Test
RNN-T	50	11.1	11.6
+ Pre-trained encoder (CTC)	20	10.7	11.3
+ Pre-trained decoder (CE)	20	10.5	11.2
+ MTL CTC on encoder (10 epoch, $\lambda_{ctc} = 0.2$)	20	10.9	11.4
+ MTL CE on decoder (10 epoch, $\lambda_{ctc} = 0.3$,$\lambda_{ce} = 0.2$)	20	10.6	11.3

TABLE 6.8 – Comparaison des performances en termes de WER du système RNN-T CRDNN (1000 unités subwords), avec des modèles pré-entrainés versus un apprentissage multi-tâche (MTL) sur le corpus LibriSpeech 100 heures. Différents types d'initialisation des poids du RNN-T sont utilisés ainsi que le décodage Greedy Search.

L'avantage principal d'un apprentissage multi-tâche est de pouvoir directement entraîner un système RNN-T sans passer au préalable par une phase d'apprentissage d'un modèle CTC ou d'un modèle de langage.

Il est clair que la fonction de jointure et la rétro-propagation du gradient est un composant sensible à l'initialisation des poids de l'encodeur/décodeur et que cela reste délicat en pratique à mettre en øeuvre.

6.4 Résultats du projet SpeechBrain sur LibriSpeech 1000 heures

La mise en place d'un système RAP E2E sur LibriSpeech 1000 heures avec une performance à moins de 3% de WER est un challenge scientifique et technique qui nécessite l'intervention de plusieurs composants :

— une exploration intensive de la modélisation E2E CRDNN et en particulier l'introduction des *Transformers*,
— un modèle de langage neuronal (récurrent et/ou transformer),
— un décodeur beam search permettant la fusion des deux modélisations acoustique et langue.

Dans le courant du mois de décembre 2020, l'équipe ASR E2E du projet SpeechBrain (composée de 6 doctorants/chercheurs, dont je fais partie) s'est engagée dans la réalisation graduelle d'un ensemble de contributions qui nous a permis d'atteindre un WER de 2.6% sur LibriSpeech 1000 heures. Je décris ici les différentes contributions réalisées de façon générale en attendant la publication de ces travaux.

6.4.1 Évaluation du modèle de langage sur le système E2E RAP de 100 heures LibriSpeech

Dans un premier temps, nous nous sommes intéressés à l'ajout d'un modèle de langage à la modélisation acoustique apprise sur LibriSpeech 100 heures. Mon collègue Jerry a préparé et utilisé les 800 Millions de mots issus du corpus LibriSpeech, ainsi que le texte issu des 960 heures du corpus. Nous avons utilisé une modélisation LSTM avec deux couches à 2048 neurones avec la tokénisation subword de 1000 unités que j'ai proposé dans mes évaluations antérieures. Il est à noter que l'apprentissage du modèle a duré 20 jours avec une Tesla V100 32Gb. Ce modèle a été entraîné sur 5 epochs, choix fait à partir des courbes d'apprentissage de ce modèle de langage.

Mes contributions sur les RNN-T ne permettaient pas jusqu'à présent d'utiliser un décodage beam search, encore moins avec un modèle de langage. De ce fait, j'ai proposé une implémentation open-source des travaux de (Jain et al., 2019) sur une version améliorée de l'algorithme beam search sur les RNN-T. Cette implémentation nous permet de réaliser un décodage beam search pour les RNN-T avec notre modèle de langage.

Le tableau ci-dessous 6.9 reprend les résultats obtenus par l'équipe avec un RNN-T et un système Attention+CTC multitâche avec ($L = \lambda_{ctc}L_{ctc} + (1 - \lambda_{attn})L_{attn}$) entraîné sur 100 heures de LibriSpeech, avec un décodage beam search et un modèle de langage BPE à 1000 unités. Dans les deux cas, l'encodeur est le CRDNN précédemment étudié.

Modèle E2E	Architecture	Beam Search	WER sur test clean	
			w/o LM	w LM
CRDNN	RNN-T	beam = 4 (best $\lambda_{lm} = 0.3$)	11.0	11.2
		beam = 4 (best $\lambda_{lm} = 0.4$) + gestion du label	11.0	**9.6**
	Attention + CTC ($\lambda_{ctc} = 0.2$)	beam = 80 (best $\lambda_{lm} = 0.45$)	**9.8**	**6.0**

TABLE 6.9 – Performances de différentes architectures E2E, entraînées sur le corpus Librispeech 100 heures, évaluées sur le corpus test-clean. Le décodage est de type beam search et le modèle de langage LSTM est entraîné sur 800 Millions de mots et les 1000 heures librispeech.

On remarque dans le tableau 6.9 que le décodage beam search avec le modèle de langage LSTM et un poids $\lambda_{lm} = 0.45$ permet d'améliorer significativement le résultat. On observe un gain de 3.8% en WER avec

le système E2E Attention+CTC en utilisant un seuil de beamsearch de 80 et le modèle de langage.

Cependant, pour le système RNN-T, on remarque une perte de performance entre un décodage beam search avec ou sans modèle de langage (LM). Après analyse des sorties RNN-T et des sorties du modèle de langage lors de la combinaison des deux informations par le beam search, il apparaît que le label du RNN-T est identifié comme étant le label “Begin of Sentence (BOS)” pour la modélisation du LM. De ce fait, le modèle de langage et le système RNN-T ont une incompatibilité de représentation. Nous proposons par la suite une petite modification dans ce cas d'extension d'une hypothèse avec le symbole qui consiste à ne pas utiliser le LM pour la prédiction du RNN-T. Avec cette modification, un gain de 1.4% de WER avec un beam de 4 a été obtenu. On remarque une plus petite amélioration avec les systèmes RNN-T qu'avec un système seq2seq avec attention. Nous nous limitons à un décodage beam search de 4 avec la modélisation RNN-T ; la raison est liée au choix de l'outil SpeechBrain de proposer un traitement des données à 100% sur GPU et la limitation GPU de 32Gb de la carte Tesla V100 ne permet pas un décodage beam search supérieur à 4.

6.4.2 Modélisation SpeechBrain sur LibriSpeech 1000 heures

Dans un deuxième temps, nous avons fait le choix collectif au sein de l'équipe E2E SpeechBrain d'avancer avec des systèmes seq2seq avec attention. Plusieurs verrous techniques et scientifiques pour la mise en place d'un système RAP à l'état de l'art sur ce type d'architecture bloquaient notre avancement :

— l'outil SpeechBrain ne proposait pas encore le calcul parallélisé et distribué,
— la modélisation neuronale à base de modèle récurrent était trop lente à entraîner. L'apprentissage du modèle de langage prenait plus de 20 jours. L'apprentissage de la modélisation CRDNN E2E prenait aux alentours de 10 jours. Les nouvelles modélisations à base de *Transformers* (Vaswani et al. (2017)) nous ont paru être une piste très intéressante pour l'apprentissage du contexte acoustique mais aussi pour l'apprentissage du contexte linguistique du modèle de langage.

Je me suis concentré dans un premier temps avec mon collègue Peter Plantiga à proposer une solution pour l'utilisation de l'outil SpeechBrain en distribué. Nous avons pris un mois de travail pour offrir une architecture de l'outil SpeechBrain qui peut etre utilisée par une ou plusieurs machines et plusieurs GPU.

Nous avons pu ainsi réduire le temps d'apprentissage des modèles CRDNN sur le corpus LibriSpeech à 7 heures par epoch au lieu de 12 heures (avec 2 GPU V100 32Gb). Cela a permis d'accélérer nos travaux de recherches et d'expérimentation.

Nous avons par la suite exploité notre nouvelle solution distribuée pour entraîner deux systèmes RAP E2E sur LibriSpeech :

— Le premier est uns système seq2seq et implique une modélisation CRDNN avec une sortie de 1000 unités subwords. Cette modélisation CRDNN de l'encodeur comprend 2 blocs CNN avec 128 et 256 filtres de kernel (3x3), un max pooling temporelle de 4 (réduction x4) est appliqué. La partie "blocs récurrents" utilise une modélisation LSTM avec 4 couches à 1024 unités. Enfin 2 blocs MLP de 512 neurones sont utilisés en sortie de l'encodeur. Pour la partie décodeur, nous utilisons 1 couche LSTM à 1024 unités avec un vecteur attention de 1024 unités. Nous réalisons un entraînement de ce système seq2seq avec 15 epochs et un batch de 8 sur les 960 heures de LibriSpeech.
— Le second est un système seq2seq "ContextNet+Transformers" avec une sortie de 5000 unités subwords. Cette modélisation comprend un encodeur avec 2 composants : 1) le premier composant est basé sur l'architecture ContexteNet proposée par Han et al. (2020) et il est composé de 9 blocs CNN avec 256 filtres pour les 4 premiers blocs et 512 filtres pour les 5 derniers ; 2) un deuxième composant Transformers (Vaswani et al. (2017)) de 12 à 8 attentions multi-tête. Le décodeur Transformers comprend 9 couches. Enfin une couche MLP de 2048 unités est utilisée à la sortie du réseaux pour prédire les 5000 unités subwords. Nous réalisons un entraînement de 60 epochs avec un batch de 8.

L'implémentation de la partie Transformers a été réalisée par mon collègue Jianyuan Zhong. L'ensemble des composants de notre système seg2seg CRDNN a été mise en place par Ju-Chieh Chou, Sung-Lin Yeh, Mirco Ravanelli, Titouan percollet, et moi-même.

Pour la partie modèle de langage, nous avons proposé deux modèles de langage entraînés sur les 800

millions de mots du projet LibriSpeech et le texte des 1000 heures de l'ensemble d'apprentissage de LibriSpeech :

1. le modèle de langage LSTM décrit plus haut avec 2 couches LSTM de 2048 unités et une sortie de 1000 subwords,
2. le modèle de langage Transformers avec 16 couches à 12 multi-têtes d'attention, avec une couche MLP de 3072 neurons après le bloc Transformers et une sortie de 5000 unités subwords .

Je limite ma description des modèles à ce niveau de détail, car nous sommes entrain d'écrire une article de revue sur ces recherches. Les paramètres d'optimisation des architectures proposées dans ce chapitre et les modèles pré-entrainées seront disponibles lorsque le projet SpeechBrain sera mis à disposition de la communauté.

En conclusion, l'ensemble des contributions ci-dessous nous ont permis de proposer un système RAP à l'état de l'art sur LibriSpeech :

— l'exploration des différentes granularités des unités de prédiction, réalisée par moi même.
— l'exploration des différentes modélisations neuronales réalisées par l'ensemble de l'équipe et Jianyuan Zhong en particulier pour les transformers.
— l'exploration de l'algorithme beam search et de sa paramétrisation (pénalisation de la longueur, smoothing des sorties E2E, exploration du *lambda* du modèle E2E et modèle de langage) réalisée par Sung-Lin Yeh.
— la mise à jour de l'outil SpeechBrain pour le support des protocoles d'apprentissage distribué, réalisé par Peter Plantiga et moi-même.

Le tableau 6.10 présente les performances des systèmes de l'équipe SpeechBrain sur le corpus LibriSpeech. Les modèles utilisent un apprentissage multi-tâche CTC+attention, et un décodage beam search joignant les sorties CTC et seq2seq. On remarque de ce fait que nos deux systèmes "CRDNN CTC+Attention" et "ContexteNet + Transformers" obtiennent respectivement un WER de 4.4% et de 4.2% sans modèle de langage.

À titre de comparaison, parmi les systèmes E2E à base de modélisations récurrentes de type LSTM, le système LAS de Aachen (Irie et al., 2019) avec un modèle de langage LSTM à 16K subwords obtient un WER de 3.2% (respectivement de 9.9%) sur l'ensemble test clean (respectivement sur l'ensemble test others). Notre "CRDNN CTC+Attention" obtient des performances comparables (voire meilleures) avec un WER de 3.1% sur l'ensemble test clean et de 9.81% sur l'ensemble test others.

Modèle E2E	LM	WER		
		Test clean		Test others
		w/o LM	w LM	
Aachen E2E LAS (BPE 16K) (Irie et al., 2019)	LSTM	4.8	3.2	9.9
	Transformers		2.8	9.3
ESPnet Transformers (BPE 5K) (Karita et al., 2019)		5.7	2.6	5.6
Wav2vec 2.0 (Char) (Baevski et al., 2020)		—	**1.8**	**3.3**
CRDNN CTC+attn (1K BPE)	LSTM ($\lambda_{lm} = 0.5$)	4.40	3.1	9.81
	Transformers ($\lambda_{lm} = 0.6$)		2.93	8.7
ContextNet + Transformers (5K BPE) CTC + Attn		**4.19**	**2.68**	**6.10**

TABLE 6.10 – Modèles de l'outil SpeechBrain entraînés sur 1000 heures de LibriSpeech et évalués sur le corpus test-clean et test-others.

Á ce jour, le modèle Wav2vec 2.0 proposé par (Baevski et al., 2020), utilisant des transformers, a obtenu un WER de 1.8% sur l'ensemble test clean et de 3.3 % sur l'ensemble test others. Dans cette approche,

un transofmer est entraîné sur 60 000 heures de LibriSpeech avec une technique de *self-supervised*, puis est adapté *fine-tuned*) avec une tâche supervisée de prédiction de caractères sur les données LibriSpeech 100 heures. Ce résultat représente actuellement la meilleure performance obtenue sur l'évaluation du corpus LibriSpeech.

Dans la catégorie des systèmes E2E à base de Transformers, le système ESPnet Transformers (Karita et al., 2019) permet d'obtenir sur l'ensemble test clean et l'ensemble test others, respectivement, un WER de 2.6% et 5.6%. On observe que notre système ContextNet + Transformers obtient un WER similaire avec un WER de 2.68 % et 6.10 %, légèrement supérieur.

L'équipe SpeechBrain ASR E2E poursuivra l'exploration des transformers et étudie l'introduction de nouvelles modélisations telles que le Conformer (Gulati et al., 2020) pour une architecture RNN-T et le Linformer (Wang et al., 2020).

6.5 Conclusion

J'ai présenté dans ce chapitre mes contributions sur les architectures RAP E2E et leurs modélisations internes.

La modélisation CRDNN proposée durant ma thèse en collaboration avec le MILA permet de dépasser l'état de l'art actuel sur le corpus TIMIT avec une performance de 13.8% en PER et une performance de 9.8% en WER sur Librispeech 100 heures.

Une étude approfondie sur la granularité des unités de prédiction des modèles RAP E2E met en évidence l'intérêt des unités subwords, avec l'avantage d'être plus proche de l'unité mot que les caractères.

Une contribution sur les systèmes RNN-T avec la parallélisation de la fonction de perte et son intégration à PyTorch (via l'outil SpeechBrain) est au coeur de mon travail. Je propose l'initialisation des RNN-T avec un apprentissage multi-tâche dans les premières itérations avec des fonctions CTC et CE pour l'encodeur et le décodeur par rapport à une initialisation à partir de modèles pré-entrainés.

L'architecture RNN-T avec son alignement monotone et son décodeur auto-regressif permet d'obtenir une bonne performance comparé à l'état de l'art des systèmes seq2seq avec mécanisme d'attention. Ceci dit, ces deniers obtiennent les meilleures performances dans nos évaluations. Il serait intéressant dans un futur proche, d'explorer l'avantage des mécanismes d'attentions sur les composants de l'architecture RNN-T.

Le travail de groupe proposé à la fin de ce chapitre résume les 6 mois de développement intensif de 6 chercheurs/doctorants pour la mise en place d'un système de référence E2E RAP et la construction graduelle de l'outil SpeechBrain. Les performances à l'état de l'art nous confortent dans le souhait de créer une communauté derrière un tel outil et poursuivre le travail pour améliorer les systèmes RAP E2E.

Chapitre 7

Conclusion

Les résultats rapportés dans cette thèse résument les principaux efforts entrepris au cours des quatre dernières années. Durant cette période, la communauté scientifique dans le domaine de la parole a connu une **révolution** grâce à la popularisation et à la maturation des techniques d'apprentissage profond (Deep Learning). Lorsque ce doctorat a débuté en novembre 2016, l'impact réel de cette technique était considéré comme prometteur et un grand effort pour le développement d'outils open-source dans ce sens était amorcé. Aujourd'hui, après seulement quelques années de recherche, les modélisations neuronales que ce soit hybrides ou end-to-end sont considérées comme une percée majeur dans le domaine de la parole. qui peut être comparée à la vague des HMMs durant les 40 dernières années.

Cette thèse s'intéresse à la problématique de la modélisation acoustique pour les sytèmes RAP dans le but d'identifier les facteurs d'amélioration des performances de reconnaissance.

L'état de l'art existant que nous avons présenté dans le chapitre 2 a montré que la plupart des travaux de modélisations pour la RAP sont découpés en deux grandes familles : les approches hybrides DNN-HMM et les approches neuronales de bout en bout "End-to-End". Le choix d'une modélisation parmi d'autres implique une identification des besoins et des objectifs et une étude des différents techniques afin de se positionner et choisir la ou les modélisation(s) cohérente(s) pour nos objectifs.

Ce chapitre clôture le document en donnant une synthèse des contributions apportées par cette thèse et en proposant des pistes pour étendre mes travaux. La synthèse des contributions est présentée dans la section 1. Les perspectives sont détaillées par la suite dans la section 2.

7.1 Synthèse des contributions

Depuis le début de ma thèse, nous étions totalement convaincus du grand potentiel qu'avait le "Deep Learning" pour mon travail sur la modélisation acoustique, les anciennes modélisations GMM-HMM étaient très loin de satisfaire le besoin de robustesse au niveau acoustique. Nous avons donc focalisé nos efforts sur les facteurs de succès des modélisations DNN rapportés dans la littérature, et nous avons graduellement contribué à la mise à disposition d'une modélisation DNN hybride industrialisée pour la langue française dans une première partie, et nous avons exploré les facteurs de succès de la modélisation E2E sur les approches CTC et RNN-T dans une deuxième partie.

Dans notre ***première contribution***, nous nous somme donné comme objectif **la réalisation de système RAP DNN-HMM hybride pour la langue française** ; ce choix a été décidé dans le cadre du projet de recherche OpenPaas :NG et répond aux besoins de mise en place de plateforme RAP industrialisée et modulaire pour des adaptations à posteriori des briques acoustiques, lexicales et linguistiques aux contextes métiers. Pour ce faire, nous avons découpé ce travail en 3 phases dont la première est la **collecte, sélection et normalisation de grandes masses de données parole et texte pour la langue française** développées dans le chapitre 3 qui est un facteur essentiel pour la modélisation DNN. La deuxième phase a consisté à mettre en place **un système hybride DNN-HMM** : nous avons effectué une étude

approfondie sur les architectures TDNN dans le chapitre 4 et nous avons proposé à la fois **l'utilisation des architectures TDNN-Factorized** qui avait un grand impact sur la rapidité de l'inférence acoustique et **une couche de self-attention** pour renforcer la prédiction acoustique. Les résultats ont montré que notre modèle TDNN-F + self-attention permet d'établir un nouveau état de l'art sur le corpus ESTER, nous avons aussi obtenu nos meilleures performances sur tous les corpus connus pour la langue Française et sur les 3 types de parole (lue, préparée et spontanée) pour avoir au final une nouvelle baseline pour un système RAP en français. Comme troisième phase de notre contribution nous avons mis en place **une plateforme industrielle appelée "LinSTT Model Factory"** à destination d'ingénieur IA pour l'adaptation des modèles hybrides en RAP, au contexte métier : adaptation acoustique, lexical ou linguistique. Cette plateforme a fait ses preuves lors de 3 réalisations industrielles et a été reprise par le projet de recherche Linto pour de future extensions.

Plusieurs de mes contributions concernent les systèmes RAP End-to-End et ont été effectuées lors de ***mon séjour scientifique au Mila*** ; elles sont présentées dans le chapitre 6. J'ai choisi dans un premier temps de proposer à la communauté Open-Source via le projet SpeechBrain une implémentation de la modélisation RNN-T et de la parallélisation de sa fonction de perte avec mon implémentation **"Numba-Transducer"** ; cette accélération GPU nous a permis d'exploiter la modélisation RNN-T et de proposer un nouvel état de l'art sur le corpus TIMIT avec notre modélisation CRDNN à base de réseaux neuronales CNN et LiGRU.

Dans le cadre des systèmes E2E, j'ai étudié la granularité de prédiction des modèles RAP E2E et j'ai montré que **les unités "Subword"** par rapport aux unités "Crossword" ou "Caractères" permettent d'obtenir de meilleure représentation pour la RAP. Notre modélisation CRDNN proposée sur les "Subword" se base sur des architectures récurrentes "LSTM" à la place de la modélisation "Li-GRU" précédemment utilisée sur la modélisation phonétique de TIMIT ; il apparaît que la 4ème porte du "LSTM" a un grand impact lorsque des réductions de dimension temporelle (4x) sont appliquées à notre modèle CRDNN. Nous avons évalué nos systèmes RAP : la performance de 9.8% de WER dépasse l'état de l'art actuel sur le sous ensemble de 100H de LibriSpeech avec notre modélisation CRDNN (LSTM) seq2seq avec attention et une performance de 10.6% de WER est atteinte avec notre système RNN-T avec des prédictions "Subword". L'architecture RNN-T avec son alignement monotone et son décodeur auto-regressive permet d'obtenir une bonne performance comparée à l'état de l'art des systèmes seq2seq à base de mécanisme d'attention ; ceci dit, l'approche seq2seq avec attention permet d'obtenir les meilleures performances dans nos évaluations. Il serait intéressant dans un future proche, d'explorer l'avantage des mécanismes d'attention sur les composants de l'architecture RNN-T.

Afin de fournir un système baseline avec un WER inférieur à 3% sur le corpus LibriSpeech, mes 5 collègues (Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Mirco Ravanelli, Titouan Percollet) et moi même de l'équipe RAP E2E du Mila, avons travaillé pendant plus de 2 mois sur **une modélisation CRDNN seq2seq et une modélisation ContextNet + Transformers** ; les performances obtenus sont respectivement de 3.1% et 2.6% de WER sur l'ensemble test clean, et de 6.1% et 9.8% de WER sur l'ensemble test other.

Inspiré par les travaux de Jimenez et al. (2018) sur la détection d'environnements sonores et la hiérarchisation des représentations, une ***troisième contribution*** a eu pour cadre les approches multi-tâche (MTL) pour la RAP à base de modélisation CTC caractères, la tâche auxiliaire étant la reconnaissance des consonnes (C) et des voyelles (V). Nous avons proposé une étude sur les différentes techniques de combinaison des sorties MTL ; nous avons proposé **l'approche Char+CV-CTC** qui consiste pour la prédiction de la séquence de sortie caractère à combiner les représentations de la tâche primaire et auxiliaire avec un apprentissage CTC. D'autres architectures MLT ont été testées : un modèle standard à deux têtes, un modèle hiérarchique ontologique caractère et CV, et comparées à notre nouvelle approche Char+CV-CTC. Les évaluations des différentes techniques ont été menées sur le corpus WSJ pour permettre d'évaluer notre approche sur un ensemble limité de données (80 heures dans notre cas). L'approche de combinaison des deux représentations Char+CV-CTC a permis de surpasser les performances de tous les modèles techniques MTL présentés et une différence absolue de 0,7% de WER avec la modélisation mono-tâche caractère. Autre résultat intéressant, nous avons montré que l'approche Char+CV-CTC permet d'atteindre la même performance qu'un modèle hybride HMM-DNN phonétique.

En parallèle à mon sujet de recherche, je me suis intéressé dans le cadre du projet de recherche Open-Paas :NG à **la détection et la mesure de l'emphase au niveau mot pour la langue française**. Cette contribution a pour but de proposer une métrique de l'analyse de l'information non verbale, à savoir l'accentuation des mots dans notre cas, pour améliorer la détection des mots clés au niveau NLP ; ce travail a été réalisé en collaboration avec l'école polytechnique et n'est pas rapporté dans ce document.

Nous avons utilisé dans ce travail le corpus SIWIS qui fournit des prononciations de parole lue en français avec une annotation manuelle des emphases. Nous avons utilisé notre système RAP pour la langue en français pour fournir les alignements des accentuations. Par la suite, nous avons proposé une modélisation CNN sur un contexte acoustique fixe de 1,5 secondes encadrant chaque mot pour réaliser notre système de détection de l'emphase. L'évaluation proposée montre que notre système établit un nouvel état de l'art à une performance de 93% de détection de l'emphase sur le corpus SIWIS.

7.2 Perspective

Les travaux rapportés dans cette thèse pourraient être étendus selon plusieurs directions. Dans cette section, nous discutons de certaines d'entre elles :

7.2.1 Systèmes hybrides LVCSR pour la langue française

Plusieurs pistes ont déjà été identifiées en conclusion de mes travaux :

1. **À propos de la modélisation LM** : La partie LM proposée dans mon travail se basait sur une modélisation simple à base de n-gram. Il serait intéressant d'explorer la partie RNNLM et de proposer une représentation linguistique plus robuste que le comptage d'occurences.
2. **À propos des unités caractères pour la prédiction acoustique des systèmes hybrides** : La partie lexicale représente un composant clé de la modélisation hybride. Le dictionnaire de prononciation utilisée dans mon travail contenait 125K mots annotés manuellement et a été étendu à 250K de mots avec un modèle graphème-phonème (G2P). De mon analyse, cette extension G2P automatique comprend un grand facteur d'introduction d'erreur de génération de prononciation. Une piste d'amélioration identifiée serait de proposer une modélisation hybride caractères où le dictionnaire contiendrait la séquence des graphèmes associés à chaque mot. Cette proposition permettrait de : 1/ d'automatiser l'extension de vocabulaire sans introduire d'erreur dans l'intégration de la brique acoustique et linguistique ; 2/ d'apprendre des représentations plus intéressantes au niveau acoustique. Il serait aussi préférable dans ce cas de combiner l'architecture TDNN-F et LTSM pour notre modèle acoustique afin de pallier au problème de l'apprentissage de représentation linguistique par le modèle acoustique neuronale.
3. **À propos de paramètres acoustiques plus représentatifs pour notre modélisation hybride** : Nous avons dans nos travaux utilisé une représentation acoustique à base de MFCC et avons validé des approches d'augmentation et de perturbation de données à partir de perturbation du rythme et de la réverbération. Il serait intéressant dans un futur proche d'exploiter les approches self-supervised telles que l'approche PASE+ proposée par Ravanelli et al. (2020) ou l'approche Wav2Vec proposée par Baevski et al. (2020) pour améliorer la robustesse de la représentation acoustique.

7.2.2 Modélisation RAP E2E

J'ai choisi de continuer mes travaux de recherche après ma thèse sur la modélisation RNN-T et j'ai pu identifier les axes suivants :

1. à court terme :
   — d'une manière générale pour la modélisation RAP E2E : **pouvoir explorer les unités "Sub-word" en prenant compte leur représentation phonétique** ; mes analyses montrent que le découpage non supervisé avec la technique BPE directement à partir du texte ne permet pas de

prendre en compte la dépendance du contexte phonétique et induit beaucoup d'erreurs dues au mauvais choix des unités BPE sub-optimales pour la RAP. Il serait intéressant d'explorer dans un futur proche des BPE prenant en compte la dépendance phonétique pour améliorer la sélection.
   - **rajouter la possibilité d'entraîner notre système RAP E2E RNN-T avec la fonction de perte "Minimum Word Error Rate"** (Prabhavalkar et al., 2018).

2. à long terme :
   - **réduire l'utilisation mémoire pour le RNN-T** : le calcul de tous les alignements possibles entre l'encodeur et le décodeur dans un système RNN-T requière la concaténation des représentations avec des dimensions T et U. Cette combinaison est très gourmande en place mémoire et réduit notre capacité de modélisation et d'utilisation des ressources GPU. Or, il est clair que le calcul de tous les alignements possibles ne sont pas pertinents et nécessaires pour l'apprentissage de notre modélisation RNN-T, les alignements les plus intéressants entre la représentation acoustique et linguistique se situent généralement en diagonale de la jointure des deux représentations. Il serait intéressant d'explorer la réduction du nombre possible d'alignements et de réaliser un sous échantillonnage (down sampling) pour la réduction de la matrice de jointure à une dimension de type $(T * U,\ N * (T + U))$ avec N un nombre fixe d'alignements à calculer.

À l'issue de cette thèse, j'ai pu mesurer l'étendue du problème du domaine de la RAP. La révolution avec l'arrivée du deep learning et l'apparition des outils open-source tels que "Pytorch" et "Tensorflow" a certes permis d'ouvrir plusieurs verrous relatifs à la modélisation neuronale "end-to-end". Cependant, le problème de la RAP n'est pas encore résolu ; l'avenir est très prometteur avec la mise à disposition de supports matériels audio open-source, d'outils open-source dédiés à l'analyse de la parole (Kaldi, SpeechBrain, Espnet), du faible coût des GPUs. Le groupements d'experts du domaine autour d'un même projet est aussi un facteur de réussite.

Je continue à travailler en recherche/développement dans ce domaine, sur les perspectives décrites précédemment qui me semblent très importantes. L'atout essentiel que j'ai pu constater personnellement, réside, comme dit ci-dessus, dans le travail d'équipe rassemblant un nombre de spécialistes relevant de différent axes traitant la problématique de la RAP et dans la qualité de leur collaboration : à titre d'exemple, au Mila, nous sommes actuellement 15 personnes à travailler sur la partie RAP "hybrides" et "end-to-end" et près de 40 sur tout le projet SpeechBrain.

Annexe A

Annexe : article sur la détection de l'emphase

Lexical Emphasis Detection in Spoken French using F-BANKs and Neural Networks

Abdelwahab Heba[1,2], Thomas Pellegrini[2], Tom Jorquera[1], Régine André-Obrecht[2], and Jean-Pierre Lorré[1]

[1] Linagora, Toulouse, France
{aheba,tjorquera,jplorre}@linagora.com
[2] IRIT, Université de Toulouse, Toulouse, France
{aheba,pellegri,obrecht}@irit.fr

Abstract. Expressiveness and non-verbal information in speech are active research topics in speech processing. In this work, we are interested in detecting emphasis at word-level as a mean to identify what are the focus words in a given utterance. We compare several machine learning techniques (Linear Discriminant Analysis, Support Vector Machines, Neural Networks) for this task carried out on SIWIS, a French speech synthesis database. Our approach consists first in aligning the spoken words to the speech signal and second to feed classifier with filter bank coefficients in order to take a binary decision at word-level: neutral / emphasized. Evaluation results show that a three-layer neural network performed best with a 93% accuracy.

Keywords: Emphasized content recognition, Non verbal information in speech, SIWIS French Speech Synthesis Database

1 Introduction

Speech in human communication is not only about the explicit message conveyed or the meaning of words, but also includes information, intentionally or not, which are expressed through nonverbal behaviors. Verbal and non-verbal information shape our interactions with others [4]. In [25], for instance, an appropriate use of emphasis was shown to improve the overall perception of synthesized speech.Word-level *emphasis* is considered as an important form of expressiveness in the speech synthesis field with the objective of drawing the listener attention on specific pieces of information.

A speech utterance may convey different meanings according to intonation. Such ambiguities can be clarified by emphasizing some words in different positions in a given utterance. Automatically detecting emphasized content may be useful in spoken language processing: localizing emphasized words may help speech understanding modules, in particular in semantic focus identification [15].

In speech production, various processes occur at word, sentence, or larger chunk levels. According to [6], the tonal variation, defined by pitch variation,

is considered as a type of pronunciation variation at the suprasegmental level. Generally, systems for automatic classification of accented words use prosody, and typically use combination of suprasegmental features such as duration, pitch, and intensity features [5,14,17,20,24]. Emphasis cues found in natural speech are more vague and heavily affected by suprasegmental features. Compared to the intensity, pitch and duration are more insensitive to the channel effects such as the distance between the speaker and the microphone. Furthermore, rather than intensity, changes in vocal loudness also affect features such as spectral balance, spectral emphasis or spectral tilt, which were explored in the detection of prominent words [3], focal accent [9], stressed and unstressed syllables [18,19,22]. Indeed, these measures were also generally found to be more reliable than intensity.

In this paper, a statistical approach that models and detects word-level emphasis patterns is investigated. Related works are dedicated to the detection of lexical stress and pitch accent detection, in particular for Computer-Assisted Language Learning [12,13,21,27,28]. The present study differs from these works from the fact that we target at detecting acoustic emphasis at lexical level and in native speech. We plan to detect emphasis at word-level as a first step for future applications we would like to address. In particular, we would like to study if keyword detection in speech transcripts could be improved using a measure of emphasis as an additional piece of information.

Our methods consists first in aligning the speech signal to the spoken words, second in classifying each word segment as emphasized or neutral using filter-bank coefficients (F-BANKs) as input to a classifier. These acoustic features measure the energy from a number of frequency bands and take time dynamics into account. Furthermore, our preliminary experiments showed that F-BANKs outperform the use of single pitch variations (F0). We compare several types of classifiers for this task. As will be reported in this paper, neural networks performed the best.

The present article is structured as follows. Section 2 describes our methodology for word-level emphasis detection, including feature extraction and model description. In Section 3, we present the SIWIS French speech synthesis database, then we report a comparison of approaches and analyze the classification results.

2 Method

Figure 1 illustrates the global system schema for an example sentence: "ce FICHIER facilitera principalement la recherche [...]" (*"this FILE will mainly ease the search for [...]"*). In this sentence, the word "FICHIER" (*FILE*) was emphasized by the speaker. As a first step, a word alignment is carried out, which automatically aligns the expected text to the audio speech signal. Then, low-level acoustic features described hereafter are extracted and fed to a binary classifier that takes decisions on the emphasized/neutral pronunciations at word-level.

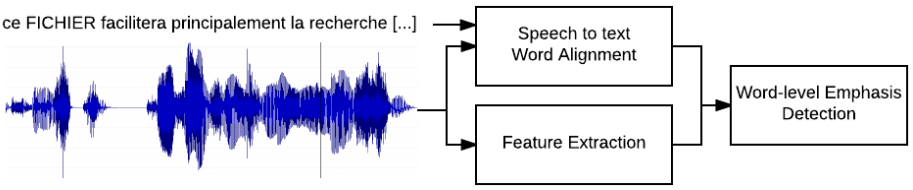


Fig. 1. Word-level emphasis detection

2.1 Word alignment

We adopt a standard approach used in speech recognition called the time alignment procedure. This procedure is accomplished using supervised phone-based recognition and produces phone-by-phone time markings, which are reduced to a word-by-word format involving the following steps [23]:

- create a word-level grammar from the orthographic transcription (read speech);
- extract acoustic features from the speech signal;
- associate a phone transcription to each word, either extracting it from our pronunciation lexicon or generating them automatically with a grapheme-to-phoneme model ; several pronunciations may be associated to a given word;
- perform the word alignment;
- extract the time markings from the aligned word segments.

We used in-house acoustical models trained with the Kaldi Speech Recognition Toolkit [16]. They were trained with the ESTER corpus [8] which consists of 90 hours of French broadcast news speech, each broadcast session contains from 20 to 40 minutes of spontaneous speech. Non-speech sounds, such as breath noises and laughter are indicated in the transcriptions and we explicitly modeled them.

We followed a standard Kaldi recipe to train the models to obtain triphone Gaussian Mixture Models / Hidden Markov Models, on 39 static, delta, and delta-delta Mel-frequency cepstral coefficients, with LDA-MLLT and Speaker Adaptive Training (SAT). Finally, we obtain triphone with about 150k Gaussian mixtures and 21.2k HMM states.

Regarding the pronunciation lexicon, we used the 105k entry CMU-Sphinx French dictionary. For out-of-vocabulary words, pronunciations were derived from a grapheme-to-phoneme tool trained over the CMU-Sphinx lexicon [2]. This concerned a set of 471 words over the 33,628 different word types contained in the SIWIS corpus used in this work.

2.2 Features

As input to the emphasis/neutral classifiers, we use 26 log filter-bank coefficients (F-BANKs) extracted on 25 ms duration frames with a hop size of 10 ms. Different

numbers of filter bands were tested, but the set of 26 static F-BANK features has shown to perform well.

For each word of duration N frames, a $N \times 26$ matrix is extracted. We compare the use of these length varying input matrices and the same matrices but in which some context is added: we add left and right frames to reach a $1s$ total duration, which gives a 108×26 matrix for each word.

These matrices are used in two ways, as an image (dimension: 108×26) fed to a Convolutional Neural Network (CNN), or as global statistics features (one value per filter-bank coefficient along time): the minimum, maximum, mean, median, standard deviation, skewness and kurtosis (dimension: 7×26), which are expected to characterize the behavior of each F-BANK coefficient to improve the temporal modeling.

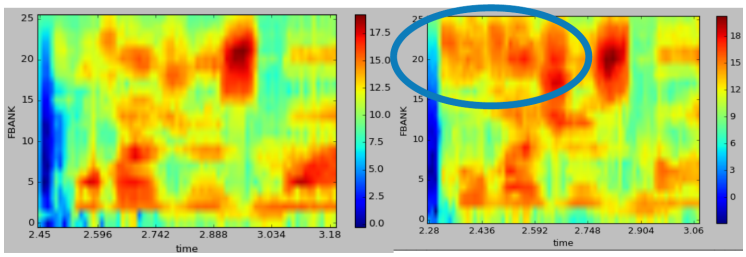


Fig. 2. The two figures represent the F-BANK coefficients for the word : "*courageusement*" (*courageously*) with and without focus emphasis on the right and on the left, respectively.

Figure 2 shows the 26 F-BANK coefficient images for both an emphasized and a neutral pronunciations of the word "courageusement" (*courageously*). We can notice that the high frequency region on the right figure (with emphasis) has higher energy values, particularly at the beginning of the word, as depicted with a blue ellipse over this area [7].

2.3 Models

Two types of neural networks were tested for our task: a neural network with fully-connected layers (FCNN), and a convolutional neural network (CNN).

In the case of FCNNs, the input layer is the concatenation of the global statistics on the 26 F-BANKs, i.e. a vector of size $7 \times 26 = 182$. The k_0 Softmax outputs estimate the emphasis of each trial. We use rectified linear (ReLU) units that have shown accurate performance in speech recognition tasks [26]. Furthermore, we experimented different number of hidden layers and different number of units. We report results with a single layer and three hidden layers each comprised of 200 units.

With CNNs, the input layer is composed of 108 frames of 26 log filter bank coefficient. Three convolution layers were respectively applied: the frequency

filtering 1×26, then dynamic time filtering 108×1 and, finally, 3×3 squared filters. Followed by 2×2 down-sampling (max-pooling) layers, and produce respectively 32, 16, and 8 activation maps that serve as input parameters for three 200-unit dense hidden layers with rectified linear unit (ReLU) activation function. Finally, the output dense layer comprises 2 units with a Softmax activation function to provide a probability.

The networks were trained with the Adam optimization [11] using a cross-entropy cost function. The regularization L^2 was used over all hidden layers. Those models are not very deep but appear to be sufficient to get insights on emphasis detection on a small database such as SIWIS. To carry out our work, Tensorflow was used to perform the experiments on a GPU TITAN 1080 device [1].

Support Vector Machines (SVM) with a Gaussian kernel and Linear Discriminant Analysis (LDA) were also used as a baseline of our experiments.

3 Experiments

3.1 Speech Material

The SIWIS French Speech Synthesis corpus contains read speech recorded from a single native female French speaker, who reads texts selected from three different written sources: books from French novels, parliament speeches and semantically unpredictable sentences. These three written sources were divided to six subsets and serve different purposes. In our study, we only use the sentences containing emphasized words (named "part 5") and their corresponding neutral sentences (contained in parts 1 to 4). The corpus contains $1575 * 2$ sentences equivalent to 3 h 35 min duration of audio, moreover, emphasized words can be seen at different positions in the sentences (begin, middle and end). The manual annotations of emphasized phones are available in the HTS label format. Indeed, SIWIS aims at building TTS systems, investigate multiple styles, and emphasis. For more information about the corpus, the reader may refer to [10].

Word alignment experiments. Since the manual annotation provided with the SIWIS database did not allow to get time markings at word-level easily, one needed to perform word alignment as a preprocessing step for emphasis detection. We have grouped the manual time markings of emphasized phones for each word to evaluate the root mean square difference between the manual and the automatic word boundaries, according to the following formula, in which t_m^i and t_a^i are the manual and automatic time markers for the i^{th} word, respectively, and N the total number of words to be aligned:

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_m^i - t_a^i)^2}$$

We worked on $1817 * 2$ words (emphasized and neutral pronunciations). The mean duration of these words is $0.372s$ ($\pm$ 19% std), the word "*monoparentales*"

(*uniparental*) has the longest duration of about $1s$, and the shortest word "*un*" (a) has a $0.03s$ duration.

The RMS obtained was 0.243. The smaller, the better the alignments. Nevertheless, we are aware that this value *per se* is difficult to interpret without any other reference value obtained on other speech data. Furthermore, the phone error rate is 8.7 %.

A set of 1695 emphasized words were found when grouping phones (*e.g.* "temps en temps" was considered as one word by the corpus annotators), but our manual re-checking of the 1575 sentences lead to 1817 emphasized words. This increase is due to the fact that we consider each word in contiguous emphasized word sequences as several emphasized words (we consider *"temps en temps"* as three different words).

3.2 Classification results

In this part, we evaluate the word-level emphasis detection method, which consists in applying the procedure shown in Figure 1. The dataset contains 1817*2 words (emphasized and neutral). The data was split into a training and a test subsets in 80%/20% proportions, respectively, and we performed a 5-fold cross-validation. We chose to keep pairs of the same words with emphasized and neutral pronunciations in the same subset, either in a training or a test fold.

In a first experiment, we focused on the global statistical features extracted over the F-BANKs. As explained previously, they were extracted with and without adding context:

- with context: all the feature matrices share the same 108×26 dimension,
- without context: the feature matrices have a variable time length according to each word: $N \times 26$.

With the different machine learning algorithms used for the classification task, we show in Table 1 that using a bit of context leads to better results in accuracy. The FCNN with 3 hidden layers with 200 units in each layer, using the ReLU activation function, obtained the best performance with a 93.4% accuracy. The variations in performance indicated in the table correspond to the variations according to the five folds used for cross-validation.

Table 1. Accuracy comparison between different classifier types.

with context	no	yes
FCNN (1 layer)	81.1 ±1.0%	89.9 ±1.7%
FCNN (3 layers)	81.3 ±5.9%	**93.4** ±3.3%
CNN	_	90.2 ±1.8%
SVM	81.5 ±1.0%	92.9 ±2.3%
LDA	76.8 ±2.4%	89.0 ±3.6%

In a second experiment, we tested the use of a CNN model. As we showed in Table 1, using context allowed better performance on this task. Consequently, we used the F-BANK images with context as input to a CNN (matrices of shape 108×26, which represent the extracted F-BANKs features over $1s$ of speech signal).

In order to train a CNN model, we needed a validation subset so that we used 70% for training, 10% for validation, and 20% for testing and always with 5 folds. During training, we noticed a clear overfitting of the model on the validation subset so that we used L^2 regularization to overcome this issue. The averaged performance on the five folds was 90.2 % ($\pm$ 1.8%), which is not as good as the SVM (92.9%) and the FCNN (93.4%).

3.3 Error Analysis

In this subsection, we analyze the errors made by the best classifier, the FCNN.

A first interesting cue concerns the influence of word duration on performance. The mean duration of the wrong predicted words is about $200ms$. The false positives (emphasized word predicted as neutral) predominantly concern short words such as *"moi"* (me), *"un"* (a), *"pas"* (not), *"cela"* (that), and their mean duration is about $140ms$. On the contrary, the false negatives are longest word mostly, such as *"historien"* (historian), *"constamment"* (constantly). Mean duration of the false negatives is around $400ms$.

By listening to some word utterances incorrectly predicted as neutral, we noticed that the relative focus on these words was as obvious as other emphasized realizations. Smaller intensity values can also be observed in their corresponding spectrograms.

We also explored if there were any relation between the word positions in sentences and the detection errors. Figure 3 shows histograms counting the errors (in black the false positives, in orange the false negatives) according to the word position: at the beginning, middle, or end of a sentence. No clear impact of word position can be observed. Nevertheless, it seems that more false negatives (emphasized words predicted as neutral) occur at the beginning and end of sentences.

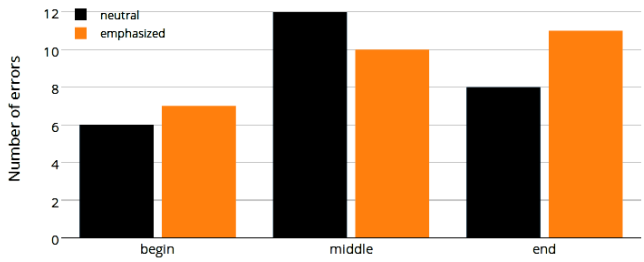


Fig. 3. Number of incorrect predictions according to the word position in sentences.

4 Conclusions

In this paper, we presented an approach to detect emphasis/neutral intonation at word level specific to the French language. As a first step, a word alignment is carried out, which automatically aligns the expected text to the audio speech signal. Then, F-BANK coefficients are extracted and fed to a binary classifier that takes decisions on the emphasized/neutral decision at word-level.

Evaluation was conducted on SIWIS, a publicly available speech database, that provides read speech material in French with a sub-part manually annotated in terms of emphasis.

Several types of classifiers were tested and the best performance was obtained with a neural network comprised of three fully-connected layers of 200 units each.

As future work, we plan to exploit this system to attempt to improve our keyword extraction module applied to speech transcripts in spoken French. Additionally, we would like to use sequence modelling approaches to carry out the detection of emphasized words through entire sentences.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
2. Bisani, M., Ney, H.: Joint-sequence models for grapheme-to-phoneme conversion. Speech communication 50(5), 434–451 (2008)
3. Campbell, N.: Loudness, spectral tilt, and perceived prominence in dialogues. In: Proceedings ICPhS. vol. 95, pp. 676–679 (1995)
4. Campbell, N.: On the use of nonverbal speech sounds in human communication. In: Verbal and nonverbal communication behaviours, pp. 117–128. Springer (2007)
5. Campbell, W.N.: Prosodic encoding of english speech. In: Second International Conference on Spoken Language Processing (1992)
6. Cohn, A.C., Fougeron, C., Huffman, M.K.: The Oxford handbook of laboratory phonology, Sect 6.2, 103-114. Oxford University Press (2012)
7. Cole, J., Mo, Y., Hasegawa-Johnson, M.: Signal-based and expectation-based factors in the perception of prosodic prominence. Laboratory Phonology 1(2), 425–452 (2010)
8. Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.F., Gravier, G.: The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In: Interspeech. pp. 1149–1152 (2005)
9. Heldner, M.: On the reliability of overall intensity and spectral emphasis as acoustic correlates of focal accents in swedish. Journal of Phonetics 31(1), 39–62 (2003)
10. Honnet, P.E., Lazaridis, A., Garner, P.N., Yamagishi, J.: The siwis french speech synthesis database? design and recording of a high quality french database for speech synthesis. Tech. rep., Idiap (2017)
11. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Li, K., Meng, H.: Automatic lexical stress and pitch accent detection for l2 english speech using multi-distribution deep neural networks. Speech Commun. (2016)

13. Li, K., Zhang, S., Li, M., Lo, W.K., Meng, H.M.: Prominence model for prosodic features in automatic lexical stress and pitch accent detection. In: INTERSPEECH. pp. 2009–2012 (2011)
14. Narupiyakul, L., Keselj, V., Cercone, N., Sirinaovakul, B.: Focus to emphasize tone analysis for prosodic generation. Computers & Mathematics with Applications 55(8), 1735–1753 (2008)
15. Noth, E., Batliner, A., Kießling, A., Kompe, R., Niemann, H.: Verbmobil: The use of prosody in the linguistic components of a speech understanding system. IEEE Transactions on Speech and Audio processing 8(5), 519–532 (2000)
16. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584, IEEE Signal Processing Society (2011)
17. Shriberg, E., Stolcke, A., Hakkani-Tür, D., Tür, G.: Prosody-based automatic segmentation of speech into sentences and topics. Speech communication 32(1), 127–154 (2000)
18. Sluijter, A.M., Shattuck-Hufnagel, S., Stevens, K.N., Van Heuven, V., et al.: Supralaryngeal resonance and glottal pulse shape as correlates of prosodic stress and accent in american english (1995)
19. Sluijter, A.M., Van Heuven, V.J.: Spectral balance as an acoustic correlate of linguistic stress. The Journal of the Acoustical society of America 100(4), 2471–2485 (1996)
20. Streefkerk, B.M., Pols, L.C., ten Bosch, L., et al.: Automatic detection of prominence (as defined by listeners' judgements) in read aloud dutch sentences. In: ICSLP (1998)
21. Tepperman, J., Narayanan, S.: Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on. vol. 1, pp. I–937. IEEE (2005)
22. Van Kuijk, D., Boves, L.: Acoustic characteristics of lexical stress in continuous telephone speech. Speech Communication 27(2), 95–111 (1999)
23. Wheatley, B., Doddington, G., Hemphill, C., Godfrey, J., Holliman, E., McDaniel, J., Fisher, D.: Robust automatic time alignment of orthographic transcriptions with unconstrained speech. In: Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on. vol. 1, pp. 533–536. IEEE (1992)
24. Wightman, C.W., Ostendorf, M.: Automatic labeling of prosodic patterns. IEEE Transactions on speech and audio processing 2(4), 469–481 (1994)
25. Yu, K., Mairesse, F., Young, S.: Word-level emphasis modelling in hmm-based speech synthesis. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. pp. 4238–4241. IEEE (2010)
26. Zeiler, M.D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q.V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., et al.: On rectified linear units for speech processing. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 3517–3521. IEEE (2013)
27. Zhao, J., Yuan, H., Liu, J., Xia, S.: Automatic lexical stress detection using acoustic features for computer assisted language learning. Proc. APSIPA ASC pp. 247–251 (2011)
28. Zhu, Y., Liu, J., Liu, R.: Automatic lexical stress detection for english learning. In: Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on. pp. 728–733. IEEE (2003)

Bibliographie

D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 2, pages 1137–1140. IEEE, 1996.

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice : A massively-multilingual speech corpus. *arXiv preprint arXiv :1912.06670*, 2019.

A. Baevski, Y. Zhou, A.-r. Mohamed, and M. Auli. wav2vec 2.0 : A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33, 2020.

D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE, 2016.

J. Baker. The dragon system–an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1) :24–29, 1975.

E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu. Exploring neural transducers for end-to-end speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 206–213. IEEE, 2017.

L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3(1) :1–8, 1972.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1) :164–171, 1970.

Y. Belinkov and J. Glass. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, pages 2441–2451, 2017.

Y. Bengio and P. Frasconi. Credit assignment through time : Alternatives to backpropagation. In *Advances in neural information processing systems*, pages 75–82, 1994.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2) :157–166, 1994.

M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Ninth European Conference on Speech Communication and Technology*, 2005.

F. Boyer and J.-L. Rouas. End-to-end speech recognition : A review for the french language. *arXiv preprint arXiv :1910.08502*, 2019.

J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell : A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

J.-C. Chappelier, M. Rajman, R. Aragüés, and A. Rozenknop. Lattice parsing for speech recognition. 1999.

C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*, 2014.

J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. End-to-end continuous speech recognition using attention-based recurrent nn : First results. *arXiv preprint arXiv :1412.1602*, 2014.

J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *Proc. ICML*, pages 2067–2075, Lille, 2015.

R. Collobert, C. Puhrsch, and G. Synnaeve. Wav2letter : an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv :1609.03193*, 2016.

K. H. Davis, R. Biddulph, and S. Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6) :637–642, 1952.

E. DELAIS-ROUSSARIE, E.-U. Langages, et al. Acsynt. 2013.

L. Deng and D. Yu. Deep learning : methods and applications. *Foundations and trends in signal processing*, 7(3–4) :197–387, 2014.

P. A. Devijver. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3(6) :369–373, 1985.

H. Dudley and S. Balashek. Automatic recognition of phonetic patterns in speech. *The Journal of the Acoustical Society of America*, 30(8) :721–732, 1958.

T. A. Edison. The perfected phonograph. *The North American Review*, 146(379) :641–650, 1888.

T. A. Edison. Phonograph or talking-machine., May 23 1916. US Patent 1,184,333.

L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The hearsay-ii speech-understanding system : Integrating knowledge to resolve uncertainty. In *Readings in artificial intelligence*, pages 349–389. Elsevier, 1981.

Y. Esteve, T. Bazillon, J.-Y. Antoine, F. Béchet, and J. Farinas. The epac corpus : Manual and automatic annotations of conversational speech in french broadcast news. In *LREC*. Citeseer, 2010.

A. Ferraresi, S. Bernardini, G. Picci, and M. Baroni. Web corpora for bilingual lexicography : A pilot study of english/french collocation extraction and translation. *Using Corpora in Contrastive and Translation Studies. Newcastle : Cambridge Scholars Publishing*, pages 337–362, 2010.

J. W. Forgie and C. D. Forgie. Results obtained from a vowel recognition computer program. *The Journal of the Acoustical Society of America*, 31(11) :1480–1489, 1959.

B. Gaiffe and K. Nehbi. Le corpus de l'est républicain. *Rapport technique, Laboratoire ATILF*, 2009.

M. Gales, S. Young, et al. The application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3) :195–304, 2008.

M. J. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2) :75–98, 1998.

M. J. Gales. Factored semi-tied covariance matrices. In *Advances In Neural Information Processing Systems*, pages 779–785, 2001.

M. J. Gales, P. Woodland, H. Y. Chan, D. Mrva, R. Sinha, S. E. Tranter, et al. Progress in the cu-htk broadcast news transcription system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14 (5) :1513–1525, 2006.

S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Ninth European Conference on Speech Communication and Technology*, 2005.

S. Galliano, G. Gravier, and L. Chaubard. The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

J.-L. Gauvain, L. F. Lamel, and M. Eskénazi. Design considerations and text selection for bref, a large french read-speech corpus. In *First International Conference on Spoken Language Processing*, 1990.

I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

A. Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv :1211.3711*, 2012.

A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al. Conformer : Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv :2005.08100*, 2020.

W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu. Contextnet : Improving convolutional neural networks for automatic speech recognition with global context. *arXiv preprint arXiv :2005.03191*, 2020.

A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deep speech : Scaling up end-to-end speech recognition. *arXiv preprint arXiv :1412.5567*, 2014.

A. Heba, T. Pellegrini, J.-P. Lorré, and R. André-Obrecht. Char+ cv-ctc : combining graphemes and consonant/vowel units for ctc-based asr using multitask learning. In *20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019)*, pages 1611–1615, 2019.

H. Hermansky, D. P. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1635–1638. IEEE, 2000.

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

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7) :1527–1554, 2006.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv :1207.0580*, 2012b.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen. On the choice of modeling unit for sequence-to-sequence speech recognition. *arXiv preprint arXiv :1902.01955*, 2019.

M. Jain, K. Schubert, J. Mahadeokar, C.-F. Yeh, K. Kalgaonkar, A. Sriram, C. Fuegen, and M. L. Seltzer. Rnn-t for latency controlled asr with improved beam search. *arXiv preprint arXiv :1911.01629*, 2019.

N. Jaitly and G. E. Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, 2013.

A. Jimenez, B. Elizalde, and B. Raj. Sound event classification using ontology-based neural networks. In *Proc. NeurIPS*, Montreal, 2018.

B.-H. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of markov chains. *AT&T technical journal*, 64(6) :1235–1249, 1985.

S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, et al. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE, 2019.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3) :400–401, 1987.

C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani. Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. 2017a.

S. Kim, T. Hori, and S. Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE, 2017b.

D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

T. Ko, V. Peddinti, D. Povey, and S. Khudanpur. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

P. Koehn. Europarl : A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

K. Krishna, S. Toshniwal, and K. Livescu. Hierarchical multitask learning for ctc-based speech recognition. *arXiv preprint arXiv :1807.06234*, 2018.

T. Kudo. Subword regularization : Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv :1804.10959*, 2018.

L. F. Lamel, J.-L. Gauvain, and M. Eskenazi. Bref, a large vocabulary spoken corpus for french. In *Second european conference on speech communication and technology*, 1991.

L. Lee and R. Rose. A frequency warping approach to speaker normalization. *IEEE Transactions on speech and audio processing*, 6(1) :49–60, 1998.

J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde. Jasper : An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv :1904.03288*, 2019a.

J. Li, R. Zhao, H. Hu, and Y. Gong. Improving rnn transducer modeling for end-to-end speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 114–121. IEEE, 2019b.

A. M. Liberman, K. S. Harris, H. S. Hoffman, and B. C. Griffith. The discrimination of speech sounds within and across phoneme boundaries. *Journal of experimental psychology*, 54(5) :358, 1957.

L. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Transactions on Information Theory*, 28(5) :729–734, 1982.

R. P. Lippmann. Speech recognition by machines and humans. *Speech communication*, 22(1) :1–15, 1997.

B. T. Lowerre. The harpy speech recognition system. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1976.

C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney. Rwth asr systems for librispeech : Hybrid vs attention–w/o data augmentation. *arXiv preprint arXiv :1905.03072*, 2019.

J. M. Maguire. The hearsay system : Around and through the thicket. *Vand. L. Rev.*, 14 :741, 1960.

Z. Meng, J. Li, Y. Gaur, and Y. Gong. Domain adaptation via teacher-student learning for end-to-end speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 268–275. IEEE, 2019.

Y. Miao, M. Gowayyed, and F. Metze. Eesen : End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174. IEEE, 2015.

A.-r. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1) :14–22, 2011.

R. K. Moore. A comparison of the data requirements of automatic speech recognition systems and human listeners. In *Eighth European Conference on Speech Communication and Technology*, 2003.

N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with hidden markov models. In *International conference on acoustics, speech, and signal processing*, pages 413–416. IEEE, 1990.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

P. R. Norvig and S. A. Intelligence. *A modern approach*. Prentice Hall, 2002.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech : an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

C. Parada, A. Sethy, M. Dredze, and F. Jelinek. A spoken term detection framework for recovering out-of-vocabulary words using the web. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Proc. NIPS*, Long Beach, 2017.

D. B. Paul and J. M. Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.

V. Peddinti, D. Povey, and S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

R. K. Potter et al. Visible speech. 1966.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747, 2018a.

D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur. A time-restricted self-attention layer for asr. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878. IEEE, 2018b.

R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly. A comparison of sequence-to-sequence models for speech recognition. In *Interspeech*, pages 939–943, 2017.

R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan. Minimum word error rate training for attention-based sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4839–4843. IEEE, 2018.

V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. wav2letter++ : The fastest open-source speech recognition system. *arXiv preprint arXiv :1812.07625*, 2018.

I. Provilkov, D. Emelianenko, and E. Voita. Bpe-dropout : Simple and effective subword regularization. *arXiv preprint arXiv :1910.13267*, 2019.

L. Qin. *Learning out-of-vocabulary words in automatic speech recognition*. PhD thesis, Citeseer, 2013.

L. Rabiner and B. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1) :4–16, 1986.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.

K. Rao and H. Sak. Multi-accent speech recognition with hierarchical grapheme based models. In *Proc. ICASSP*, pages 4815–4819, New Orleans, 2017.

K. Rao, H. Sak, and R. Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 193–199. IEEE, 2017.

M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE, 2018.

M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2) :92–102, 2018.

M. Ravanelli, T. Parcollet, and Y. Bengio. The pytorch-kaldi speech recognition toolkit. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6465–6469. IEEE, 2019.

M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio. Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 2020.

F. Richardson, M. Ostendorf, and J. R. Rohlicek. Lattice-based search strategies for large vocabulary speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 576–579. IEEE, 1995.

T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals. Wsjcamo : a british english speech corpus for large vocabulary continuous speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 81–84. IEEE, 1995.

R. Sanabria and F. Metze. Hierarchical Multitask Learning With CTC. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 485–490. IEEE, 2018.

G. Saon, H. Soltau, D. Nahamoo, and M. Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59. IEEE, 2013.

F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv :1508.07909*, 2015.

Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa. Byte pair encoding : A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1) :1929–1958, 2014.

Y. Su and C.-C. J. Kuo. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing*, 356 :151–161, 2019.

S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier, 2003.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30 :5998–6008, 2017.

W. Von Kempelen. *Mechanismus der menschlichen Sprache*. Degen, 1791.

S. Wang, B. Li, M. Khabsa, H. Fang, and H. Ma. Linformer : Self-attention with linear complexity. *arXiv preprint arXiv :2006.04768*, 2020.

G. Weiss, Y. Goldberg, and E. Yahav. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv :1805.04908*, 2018.

P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using htk. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–125. IEEE, 1994.

P. Zhan and A. Waibel. Vocal tract length normalization for large vocabulary continuous speech recognition. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1997.

Y. Zhang, W. Chan, and N. Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849. IEEE, 2017.