



**HAL**  
open science

# Prédiction de performances des systèmes de Reconnaissance Automatique de la Parole

Zied Elloumi

► **To cite this version:**

Zied Elloumi. Prédiction de performances des systèmes de Reconnaissance Automatique de la Parole. Automatique. Université Grenoble Alpes, 2019. Français. NNT : 2019GREAM005 . tel-02173343

**HAL Id: tel-02173343**

**<https://theses.hal.science/tel-02173343>**

Submitted on 4 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## THÈSE

Pour obtenir le grade de

**DOCTEUR DE LA COMMUNAUTE UNIVERSITE  
GRENOBLE ALPES**

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

**ZIED ELLOUMI**

Thèse dirigée par **Laurent Besacier** et co-dirigée par **Olivier Galibert** et **Benjamin Lecouteux**

préparée au sein du **Laboratoire national de métrologie et d'essais (LNE)** et **Laboratoire d'informatique de Grenoble (LIG)**

dans l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

# **Prédiction de performances des systèmes de Reconnaissance Automatique de la Parole**

Thèse soutenue publiquement le **18 Mars 2019**  
devant le jury composé de :

**M, Laurent, Besacier**

Professeur, Université Grenoble Alpes, Directeur de thèse

**M, Jean-Francois, Bonastre**

Professeur, Université d'Avignon, Président

**M, Denis, Jovet**

Professeur, Université de Lorraine, Rapporteur

**M, Julien, Pinquier**

Maître de Conférences HDR, IRIT, Rapporteur

**M, Olivier, Galibert**

Ingénieur de Recherche, LNE, Co-encadrant de thèse

**M, Benjamin, Lecouteux**

Maître de Conférences, Université Grenoble Alpes, Co-encadrant de thèse



# Résumé

Nous abordons dans cette thèse la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole (SRAP). Il s'agit d'une tâche utile pour mesurer la fiabilité d'hypothèses de transcription issues d'une nouvelle collection de données, lorsque la transcription de référence est indisponible et que le SRAP utilisé est inconnu (boîte noire).

Notre contribution porte sur plusieurs axes : d'abord, nous proposons un corpus français hétérogène pour apprendre et évaluer des systèmes de prédiction de performances ainsi que des SRAP. Nous comparons par la suite deux approches de prédiction : une approche à l'état de l'art basée sur l'extraction explicite de traits et une nouvelle approche basée sur des caractéristiques entraînées implicitement à l'aide des réseaux neuronaux convolutifs (CNN). L'utilisation jointe de traits textuels et acoustiques n'apporte pas de gains avec l'approche état de l'art, tandis qu'elle permet d'obtenir de meilleures prédictions en utilisant les CNNs. Nous montrons également que les CNNs prédisent clairement la distribution des taux d'erreur sur une collection d'enregistrements, contrairement à l'approche état de l'art qui génère une distribution éloignée de la réalité.

Ensuite, nous analysons des facteurs impactant les deux approches de prédiction. Nous évaluons également l'impact de la quantité d'apprentissage des systèmes de prédiction ainsi que la robustesse des systèmes appris avec les sorties d'un SRAP particulier et utilisés pour prédire la performance sur une nouvelle collection de données. Nos résultats expérimentaux montrent que les deux approches de prédiction sont robustes et que la tâche de prédiction est plus difficile sur des tours de parole courts ainsi que sur les tours de parole ayant un style de parole spontané.

Enfin, nous essayons de comprendre quelles informations sont capturées par notre modèle neuronal et leurs liens avec différents facteurs. Nos expériences montrent que les représentations intermédiaires dans le réseau encodent implicitement des informations sur le style de la parole, l'accent du locuteur ainsi que le type d'émission. Pour tirer profit de cette analyse, nous proposons un système multi-tâche qui se montre légèrement plus efficace sur la tâche de prédiction de performance.

# Abstract

In this thesis, we focus on performance prediction of automatic speech recognition (ASR) systems. This is a very useful task to measure the reliability of transcription hypotheses for a new data collection, when the reference transcription is unavailable and the ASR system used is unknown (black box).

Our contribution focuses on several areas : first, we propose a heterogeneous French corpus to learn and evaluate ASR prediction systems. We then compare two prediction approaches : a state-of-the-art performance prediction based on engineered features and a new strategy based on learnt features using convolutional neural networks (CNNs). While the joint use of textual and signal features did not work for the state-of-the-art system, the combination of inputs for CNNs leads to the best word error rate prediction performance. We also show that our CNN prediction remarkably predicts the shape of the word error rate distribution on a collection of speech recordings.

Then, we analyze factors impacting both prediction approaches. We also assess the impact of the training size of prediction systems as well as the robustness of systems learned with the outputs of a particular ASR system and used to predict performance on a new data collection. Our experimental results show that both prediction approaches are robust and that the prediction task is more difficult on short speech turns as well as spontaneous speech style.

Finally, we try to understand which information is captured by our neural model and its relation with different factors. Our experiences show that intermediate representations in the network automatically encode information on the speech style, the speaker's accent as well as the broadcast program type. To take advantage of this analysis, we propose a multi-task system that is slightly more effective on the performance prediction task.



## ACRONYMES

- TAL** Traitement Automatique du Langage
- TALN** Traitement Automatique du Langage Naturel
- PP** Prédiction de Performance
- MC** Mesures de Confiance
- RAP** Reconnaissance Automatique de la Parole
- SRAP** Système de Reconnaissance Automatique de la Parole
- HMM** Hidden Markov Model
- GMM** Gaussian Mixture Model
- SGMM** Subspace Gaussian Mixture Model
- DNN** Deep neural network
- CNN** Convolutional Neural Network
- ANN** Artificial Neural Network
- ML** Modèle de Langage
- MA** Modèle Acoustique
- PLP** Perceptual Linear Prediction
- t-SNE** t-Distributed Stochastic Neighbor Embedding
- WER** Word Error Rate
- TP** Tour de Parole



# TABLE DES MATIÈRES

<b>Introduction générale</b>	<b>2</b>
<b>I Contexte de travail et état de l'art</b>	<b>7</b>
<b>1 La reconnaissance automatique de la parole</b>	<b>9</b>
1.1 Principe . . . . .	10
1.2 Extraction des paramètres . . . . .	11
1.3 Modélisation acoustique . . . . .	11
1.3.1 Modèles de Markov Cachés . . . . .	12
1.3.2 Les modèles à mélange de gaussiennes . . . . .	13
1.3.3 Les sous-espaces de modèle à mélange de gaussiens . . . . .	13
1.3.4 Réseaux de neurones profonds . . . . .	14
1.4 Modélisation linguistique . . . . .	15
1.5 Dictionnaire de prononciation . . . . .	17
1.6 Kaldi . . . . .	18
1.7 Évaluation des SRAP . . . . .	18
1.8 Conclusion . . . . .	19
<b>2 La prédiction de performances</b>	<b>21</b>
2.1 Introduction . . . . .	22
2.2 L'estimation des mesures de confiance . . . . .	23
2.3 La prédiction de performances . . . . .	25

2.3.1	Principe . . . . .	27
2.3.2	Granularité . . . . .	28
2.3.3	Évaluation . . . . .	29
2.3.4	Travaux connexes . . . . .	29
2.4	Conclusion . . . . .	31
<b>3</b>	<b>Les réseaux de neurones convolutifs en traitement automatique des langues</b>	<b>33</b>
3.1	Introduction . . . . .	34
3.2	Un neurone formel . . . . .	36
3.3	Extraction et apprentissage des traits . . . . .	37
3.3.1	L'entrée du réseau . . . . .	37
3.3.2	La convolution . . . . .	40
3.3.3	Le <i>pooling</i> . . . . .	43
3.4	Modélisation et prédiction . . . . .	45
3.5	Apprentissage d'un réseau de neurones . . . . .	46
3.6	Conclusion . . . . .	49
<b>II</b>	<b>Contributions</b>	<b>51</b>
<b>4</b>	<b>Cadre expérimental</b>	<b>53</b>
4.1	Scénario envisagé . . . . .	54
4.2	Corpus . . . . .	56
4.3	Métriques d'évaluation . . . . .	57
4.4	Systèmes de reconnaissance de la parole construits . . . . .	58
4.4.1	Processus de pré-traitement . . . . .	58
4.4.2	Modèles acoustiques . . . . .	59
4.4.3	Modèles de langage . . . . .	60
4.4.4	Dictionnaire de prononciation . . . . .	62
4.4.5	Évaluation des systèmes . . . . .	64
4.5	Conclusion . . . . .	65
<b>5</b>	<b>Implémentation des systèmes de prédiction de performances</b>	<b>69</b>
5.1	Prédiction basée sur des traits explicites ( <i>baseline</i> ) . . . . .	70
5.2	Prédiction par les réseaux neuronaux convolutifs (CNNs) . . . . .	72

5.2.1	Architecture . . . . .	72
5.2.2	Expériences . . . . .	74
5.2.3	Résultats . . . . .	76
5.2.4	Analyse des taux d'erreur de mots prédits . . . . .	78
5.3	Conclusion . . . . .	79
<b>6</b>	<b>Analyse des facteurs impactant nos systèmes de prédiction de performances</b>	<b>81</b>
6.1	Effet de la durée et des styles de parole sur la qualité des SPPs . . . . .	82
6.1.1	Analyse par durée des tours de parole . . . . .	82
6.1.2	Évaluation de l'impact du style de parole sur la qualité des SPPs . . . . .	83
6.2	Évaluation de la robustesse des systèmes de prédiction de performances . . . . .	85
6.2.1	Impact de la taille du corpus d'apprentissage sur la qualité des SPPs . . . . .	87
6.2.2	Effet de la qualité du SRAP ayant généré les données d'apprentissage sur l'apprentissage des SPPs . . . . .	89
6.3	Conclusion . . . . .	91
<b>7</b>	<b>Évaluation des représentations apprises par le système de prédiction neuronal</b>	<b>93</b>
7.1	Travaux existants . . . . .	94
7.2	Méthodologie . . . . .	95
7.3	Analyse par classification . . . . .	96
7.3.1	Classifieur peu profond pour l'analyse . . . . .	96
7.3.2	Données . . . . .	97
7.3.3	Résultats . . . . .	98
7.4	Analyse par visualisation . . . . .	101
7.5	Apprentissage multi-tâche . . . . .	101
7.6	Conclusion . . . . .	104
<b>8</b>	<b>Conclusion et perspectives</b>	<b>107</b>
<b>A</b>	<b>Annexes</b>	<b>125</b>



## LISTE DES TABLEAUX

2.1	Les principales différences entre l'estimation des mesures de confiance et la prédiction de performances . . . . .	26
4.1	Distribution de nos corpus entre les styles de parole non spontanés (NS) et spontanés (S) . . . . .	57
4.2	Exemple de tour de parole avant et après le processus de pré-traitement . . . . .	59
4.3	Description des modèles acoustiques produits en termes de méthode d'apprentissage et script utilisé . . . . .	60
4.4	Description des données monolingues utilisés pour construire les modèles de langage pour le système RAP-LIG . . . . .	61
4.5	Performances des deux modèles de langage filtrés LM-3G et LM-5G évalués sur différents corpus en termes de perplexité . . . . .	62
4.6	Liste des variantes de phonèmes supplémentaires pour BDLEX . . . . .	63
4.7	Les règles suivies pour corriger les phonétisations . . . . .	63
4.8	Description des 2 systèmes de reconnaissance automatique de la parole produits et leurs performances WER évalués sur nos corpus $\text{Train}_{Pred}$ et $\text{Test}_{Pred}$ . . . . .	64
4.9	Performance sur le corpus $\text{Test}_{Pred}$ en termes de WER (Taux d'Erreur Mots) . . . . .	66
4.10	Performance sur le corpus $\text{Train}_{Pred}$ en termes de WER (Taux d'Erreur Mots) . . . . .	66

5.1	Liste des phonèmes de notre dictionnaire de phonétisation avec leurs types . . . . .	71
5.2	TranscRater <i>vs</i> $CNN_{Softmax}$ <i>vs</i> $CNN_{ReLU}$ évaluées au niveau des tours de parole avec la métrique MAE ou Kendall sur le corpus $Test_{pred}$ . . . . .	77
5.3	TranscRater (POS+LEX+LM+SIG) <i>vs</i> $CNN_{Softmax}$ (EMBED+RAW-SIG) des WER prédits (moyennés sur toutes les phrases) par type de parole (NS/S) sur le corpus $Test_{pred}$ . . . . .	78
6.1	Performances des systèmes TR et CNN évalué sur le corpus $Test_{Pred}$ en terme de MAE selon les durées des tours de parole . . . . .	83
6.2	Performances des systèmes de prédiction TR et CNN évalués sur les deux sous ensembles NS et S et la totalité du corpus $Test_{Pred}$ (NS+S) en termes de MAE. . . . .	83
6.3	Description des 4 systèmes de reconnaissance automatique de la parole produits et leurs performances évalués sur nos corpus pour tâche prédiction de performance $Train_i$ et $Test_i$ en termes de WER- $i$ l'id du SRAP utilisé pour la transcription automatique . . . . .	86
6.4	Évaluation du système $SRAP_1$ sur des sous-échantillons corpus d'apprentissage des systèmes de prédiction en terme de WER . . . . .	87
6.5	Évaluation des nouveaux systèmes <b>TR</b> sur 4 corpus d'évaluation $Test_i$ ( $ASR_i$ ) en termes de MAE . . . . .	88
6.6	Évaluation des nouveaux systèmes <b>CNN</b> sur 4 corpus d'évaluation $Test_i$ ( $ASR_i$ ) en termes de MAE . . . . .	88
6.7	Effet de la qualité des sorties des systèmes de RAP sur la performance des systèmes TR en termes de MAE . . . . .	90
6.8	Effet de la qualité des sorties des systèmes de RAP sur la performance des systèmes CNN en termes de MAE . . . . .	90
7.1	Distribution des tours de parole entre les styles non spontanés et spontanés et accents natifs/non natifs . . . . .	97
7.2	Nombre des tours de parole pour chaque émission . . . . .	97
7.3	Description de notre ensemble de données équilibré pour chaque catégorie . . . . .	98

7.4	Performances des systèmes de classification Émission/Style/Accent en termes de taux de bonne classification en utilisant les représentations apprises durant l'apprentissage de notre système de prédiction	99
7.5	Évaluation de la prédiction de performance du SRAP avec des modèles multi-tâche ( $DEV  TEST$ ) en terme de MAE et Kendall - en termes de taux de bonne classification pour les tâches de classification secondaires	103
7.6	Évaluation des tâches de classification secondaires des modèles multi-tâche ( $DEV  TEST$ ) en termes de taux de bonne classification	103
A.1	Évaluation des modèles TranscRater sur le corpus de TEST en termes de MAE	126
A.2	Évaluation des 10 modèles de prédiction $CNN_{softmax}$ EMBED+RAW-SIG sur les corpus DEV et TEST en termes de MAE	127
A.3	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	127
A.4	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	129
A.5	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	129
A.6	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	130
A.7	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	130
A.8	Exemple de transcription automatique (obtenue par $SRAP_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et $CNN_{softmax}$	131
A.9	Exemples de WER prédits au niveau des tours de parole par les meilleurs systèmes de prédiction : TranscRater <i>vs</i> $CNN_{softmax}$ EMBED + RAW-SIG	132

A.10 Exemples de WER prédits au niveau des tours de parole par les meilleurs systèmes de prédiction : TranscRater <i>vs</i> CNN <sub>Softmax</sub> EMBED+RAW- SIG . . . . .	133
---	-----

## TABLE DES FIGURES

1	Boucle vertueuse pour créer ou adapter des systèmes de RAP selon des besoins identifiés . . . . .	2
1.1	Architecture d'un système de reconnaissance automatique de la parole statistique . . . . .	11
1.2	Exemple d'un HMM à 3 états émetteurs. . . . .	13
1.3	Architecture d'un modèle hybride HMM/DNN . . . . .	15
1.4	Architecture de la boîte à outils Kaldi [Povey et al., 2011b] . . . . .	19
2.1	Processus d'apprentissage des systèmes de prédiction de performances	28
2.2	Processus de prédiction de performances d'une nouvelle collection de données . . . . .	28
3.1	Les principaux blocs de construction d'une architecture utilisant un réseau de neurones convolutif simple . . . . .	35
3.2	Architecture d'un neurone formel . . . . .	36
3.3	Processus de transformation d'une séquence de mots en une représentation matricielle par la couche <i>EMBED</i> . . . . .	39
3.4	Exemple de d'utilisations d'un signal acoustique à l'entrée des réseaux de neurones convolutifs . . . . .	40
3.5	Exemple des différentes dimensions des opérations de convolution .	42
3.6	Exemple d'application d'une convolution 1D sur une grille de mots <i>A</i>	43
3.7	Fonctionnement d'une opération de <i>Max-pooling</i> appliquée sur une carte de caractéristiques . . . . .	45

3.8	Architecture du bloc de modélisation et prédiction . . . . .	47
4.1	Protocole d'évaluation pour la tâche de prédiction de performance	55
4.2	Apprentissage à l'état de l'art d'un système de prédiction de performances . . . . .	56
4.3	Apprentissage de bout-en-bout d'un système de prédiction profond en utilisant les réseaux de neurones . . . . .	56
4.4	Performance (WER) du système SRAP <sub>1</sub> comparé aux performances d'autres systèmes de la compagnie d'évaluation REPERE sur des données identiques. Chaque fichier en abscisse représente une instance d'un type d'émission. . . . .	65
5.1	Architecture de nos CNN à partir d'entrées texte (vert) et signal (rouge). Les couches avec des pointillés correspondent à l'utilisation conjointe texte+signal . . . . .	75
5.2	Distribution des tours de parole en fonction de leurs WER : (a) Référence (b) Prédit par le meilleur système TranscRater (c) Prédit par le meilleur système CNN . . . . .	79
6.1	Évaluation des systèmes TR et CNN en terme de $\Delta_{MAE}$ (CNN est meilleur lorsque $\Delta_{MAE} > 0$ ) sur le corpus Test <sub>1</sub> (transcrit par SRAP <sub>1</sub> ) au niveau des instances d'émission pour les styles de parole NS (vert) et S (rouge) . . . . .	84
6.2	Évaluation des systèmes de prédiction sur le corpus Test <sub>1</sub> (transcrit avec SRAP <sub>1</sub> ) en termes de MAE au niveau du type d'émission . . . .	86
7.1	Architecture de notre CNN - en jaune les couches de représentations qui vont être analysées . . . . .	96
7.2	Visualisation des représentations des tours de parole de la couche C2 pour les différents styles de parole (Spontanée/Non spontanée). (a) des tours de parole ayant une durée de 4 à 5 s et (b) de 5 à 6 s .	100
7.3	Matrice de confusion de la classification EMISSION en utilisant les représentations de la couche C2 (EMBED+RAW-SIG) comme entrée - évaluée sur le corpus DEV . . . . .	102

A.1	Distribution des tours de parole du corpus $\text{Test}_{pred}$ en fonction de leurs WER . . . . .	125
A.2	Distribution des tours de parole du corpus TEST en fonction de leurs WER . . . . .	126
A.3	Distribution des tours de parole du corpus TRAIN en fonction de leurs durées (en seconde) . . . . .	128
A.4	Distribution des tours de parole du corpus TEST en fonction de leurs durées (en seconde) . . . . .	128

# Introduction

## Motivation et objectifs

L'évaluation des systèmes de reconnaissance automatique de la parole (SRAP) est importante pour la communauté scientifique. Il s'agit d'une opération fondamentale et indispensable (comme illustré dans la figure 1) pour mesurer la fiabilité d'un système appris, et sa capacité de réaliser une tâche bien précise déterminée par des besoins théoriques et applicatifs. Plusieurs campagnes d'évaluation ont été organisées en reconnaissance automatique de la parole, tel que : NIST [Martin and Greenberg, 2009], QUAERO [Galibert et al., 2011], REPERE [Kahn et al., 2012], ESTER [Galliano et al., 2005], etc.

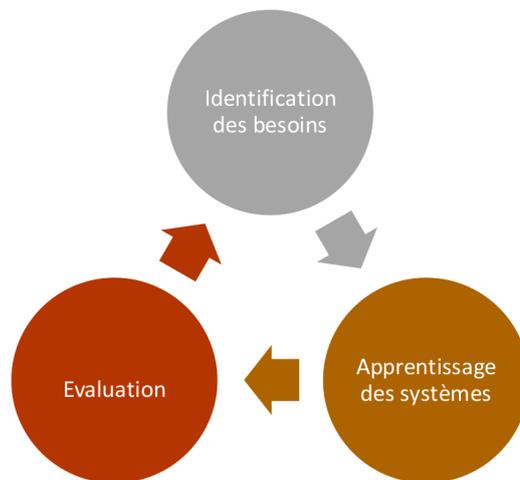


FIGURE 1 – Boucle vertueuse pour créer ou adapter des systèmes de RAP selon des besoins identifiés

L'évaluation d'un SRAP implique une transcription de référence (produite par des experts), une hypothèse de transcription (automatique) et une métrique d'évaluation spécifique à la tâche telle que le taux d'erreur de mots. Étant donné que la production d'une transcription de référence est très coûteuse (en temps et ressources), l'estimation de qualité peut être une tâche très utile pour mesurer la

fiabilité d'une transcription automatique, notamment lorsque la référence est indisponible.

Dans ce cas, un module « estimation de qualité » peut être intégré dans la boucle vertueuse de la figure 1 en remplaçant le module « Évaluation automatique ». Ainsi, le nouveau module doit être capable d'estimer correctement la qualité d'un système particulier pour déterminer sa capacité à répondre aux besoins théoriques et applicatifs identifiés par un expert et estimer le coût d'adaptation nécessaire afin de résoudre une nouvelle tâche et/ou atteindre les objectifs identifiés.

Pour estimer la qualité d'un SRAP, de nombreux travaux ont proposé d'estimer des mesures de confiance (MC) afin de détecter les erreurs dans les sorties d'un SRAP. Ces mesures consistent à étiqueter chaque mot en entrée comme « correct » ou « incorrect ». Cette opération est fondée sur des informations issues d'un système de reconnaissance automatique de la parole particulier. Toutefois, si le SRAP change ou que son développement interne est inaccessible, l'estimation des mesures de confiance est beaucoup plus difficile voire impossible. Dans cette situation, la prédiction de performances est la tâche adéquate, puisqu'elle ne se concentre pas sur un système de reconnaissance automatique de la parole particulier (ni sur des treillis ou des N-meilleures hypothèses). C'est une nouvelle tâche qui consiste à prédire le taux d'erreur de mots d'une nouvelle collection de signaux (jamais rencontrée auparavant) où le système de reconnaissance automatique de la parole est inconnu (boîte noire). La prédiction de performances s'effectue à une granularité plus large que le mot, telle que : en tour de parole (*utterance*) ou document. C'est une tâche très utile pour déterminer la difficulté de transcription d'une nouvelle collection de données et estimer le coût de création des nouveaux systèmes de RAP ainsi que le coût d'adaptation un système de RAP à une nouvelle tâche (langue, dialecte ou style de parole).

## Contributions

Dans ce manuscrit de thèse, nous nous intéressons à la tâche de prédiction de performances des systèmes de RAP, où le système de RAP est considéré comme une boîte noire. Étant donné que les approches de l'état de l'art se fondent sur l'extraction de traits pré-définis, nous proposons dans ce travail une approche *end-*

*to-end* fondée sur des traits entraînés à l'aide des réseaux de neurones convolutifs. Nos systèmes de prédiction sont appris sur des entrées acoustiques et textuelles pour prédire un taux d'erreur de mots au niveau de chaque tour de parole (*utterance*). Ainsi, le réseau de neurones doit être capable d'extraire, d'apprendre et d'adapter des caractéristiques spécifiques à la tâche principale pour bien prédire les performances d'un ou plusieurs systèmes de RAP inconnus. Dans ce travail, nous présentons les contributions suivantes :

1. Proposer un protocole expérimental spécifique à la tâche de prédiction de performances lorsque les références de transcription sont indisponibles et le système de RAP est inconnu (une boîte noire) ;
2. Proposer un corpus de parole français hétérogène (difficile à transcrire) spécifique à la tâche de prédiction de performances ;
3. Proposer une méthode flexible (ne dépendant pas de la langue) qui se fonde sur des représentations apprises au cours de l'apprentissage du système à l'aide des réseaux de neurones ;
4. Comparer la méthode proposée à une méthode état de l'art ;
5. Analyser les facteurs impactant la qualité des systèmes de prédiction ;
6. Évaluer la robustesse des systèmes de prédiction appris et/ou évalués sur plusieurs systèmes de RAP ;
7. Analyser les représentations intermédiaires apprises par le meilleur système neuronal afin de comprendre quelles informations sont capturées au moment de l'apprentissage du réseau.

## Structure du document

Ce manuscrit de thèse est organisé en deux grandes parties : état de l'art et contributions. Nous introduisons dans la première partie la reconnaissance automatique de la parole, la prédiction de performances ainsi que les réseaux de neurones convolutifs. Dans la deuxième partie, nous présentons nos travaux expérimentaux et nos contributions sur la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole.

Ainsi, les chapitres de ce manuscrit de thèse sont organisés comme suit :

**Chapitre 1** : nous présentons le principe de fonctionnement d'un système de reconnaissance automatique de la parole et les composants fondamentaux pour construire une système de RAP à l'état de l'art.

**Chapitre 2** : nous introduisons la tâche de prédiction de performances en la comparant à une tâche classique d'estimation de qualité des systèmes de RAP qui est l'estimation des mesures de confiance.

**Chapitre 3** : nous détaillons le fonctionnement de base d'un réseau de neurones convolutif en décrivant ses différentes caractéristiques pour le traitement des données textuelles et acoustiques.

**Chapitre 4** : nous proposons un protocole expérimental spécifique pour la tâche de prédiction de performances. Nous présentons le scénario envisagé, un corpus hétérogène spécifique à la tâche de prédiction, les métriques d'évaluation ainsi que les systèmes de reconnaissance automatique de la parole créés pour obtenir les transcriptions automatiques.

**Chapitre 5** : nous proposons une nouvelle approche pour la tâche de prédiction de performances fondée sur des caractéristiques apprises à l'aide des réseaux de neurones convolutifs. Nous comparons également les performances obtenues à un système à l'état de l'art fondé sur des traits pré-définis.

**Chapitre 6** : nous proposons une analyse profonde en étudiant des facteurs impactant la qualité des systèmes de prédiction. Nous étudions ensuite la robustesse tout en évaluant l'impact de la taille des corpus d'apprentissage et l'effet de la qualité des systèmes de RAP sur la qualité des systèmes de prédiction de performances.

**Chapitre 7** nous évaluons les représentations intermédiaires apprises par notre meilleur système de prédiction CNN par rapport à différents facteurs en utilisant deux approches d'évaluation : une évaluation par classification et une évaluation par visualisation. Nous essayons ensuite de tirer profit de cette analyse en proposant un apprentissage multi-tâche qui prend implicitement les informations détectées.

**Chapitre 8** nous concluons ce manuscrit de thèse par un résumé de nos travaux ainsi que nos perspectives.

---

## Contexte de la thèse

Cette thèse s'est réalisée dans le cadre d'une convention industrielle de formation par la recherche (CIFRE) entre le laboratoire national de métrologie et d'essais (LNE) et le laboratoire d'informatique de Grenoble (LIG) au sein de l'équipe GETALP.

Le LNE est un Établissement Public à Caractère Industriel et Commercial (EPIC). Il couvre plusieurs domaines d'activités en tant que laboratoire d'essais, expert-technique, laboratoire national de métrologie, organisme de recherche et organisme de certification. Le LNE organise des campagnes d'évaluation depuis plus de 8 ans comme les évaluations QUAERO [Galibert et al., 2011], REPERE [Kahn et al., 2012] ou MAURDOR [Brunessaux et al., 2014].

L'équipe GETALP (Groupe d'Étude en Traduction Automatique/Traitement Automatisé des Langues et de la Parole) est née en 2007 lors de la création du LIG. Issue de l'union vertueuse de chercheurs en traitement de l'écrit et de la parole, c'est une équipe pluridisciplinaire (informaticiens, linguistes, phonéticiens, traducteurs et traiteurs de signaux) dont l'objectif est d'aborder tous les aspects théoriques, méthodologiques et pratiques de la communication et du traitement de l'information multilingue (écrite ou orale).

## Première partie

### Contexte de travail et état de l'art



# CHAPITRE 1

## LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

### Sommaire

---

<b>1.1 Principe</b> . . . . .	<b>10</b>
<b>1.2 Extraction des paramètres</b> . . . . .	<b>11</b>
<b>1.3 Modélisation acoustique</b> . . . . .	<b>11</b>
1.3.1 Modèles de Markov Cachés . . . . .	12
1.3.2 Les modèles à mélange de gaussiennes . . . . .	13
1.3.3 Les sous-espaces de modèle à mélange de gaussiens . . . . .	13
1.3.4 Réseaux de neurones profonds . . . . .	14
<b>1.4 Modélisation linguistique</b> . . . . .	<b>15</b>
<b>1.5 Dictionnaire de prononciation</b> . . . . .	<b>17</b>
<b>1.6 Kaldi</b> . . . . .	<b>18</b>
<b>1.7 Évaluation des SRAP</b> . . . . .	<b>18</b>
<b>1.8 Conclusion</b> . . . . .	<b>19</b>

---

**D**ans ce chapitre, nous nous intéressons à la tâche de reconnaissance automatique de la parole.

Ce chapitre est organisé comme suit : nous présentons dans la section 1.1 les principes de base ainsi que les principaux composants d’un système de reconnaissance automatique de la parole. Nous détaillons ensuite dans la section 1.2, le processus d’extraction des paramètres acoustiques et les différentes techniques de transformation d’un signal de parole brut en une représentation exploitable par le système de reconnaissance automatique de la parole. Puis, nous présentons les différents types de modélisation acoustique et linguistique dans les sections 1.3 et 1.4. Dans la section 1.6, nous présentons la boîte à outils Kaldi et les configurations utilisées pour créer nos systèmes de RAP. Enfin, nous concluons ce chapitre en présentant la métrique utilisée pour nos évaluations dans la section 1.7.

## 1.1 Principe

Les systèmes modernes de reconnaissance automatique de la parole ont été introduits par Jelinek [1976]. Il s’agit de transcrire un signal de parole prononcé  $X$  en une séquence de mots  $\hat{W}$  la plus probable.

Comme décrit dans la figure 1.1, le système de reconnaissance automatique de la parole statistique s’appuie sur l’extraction des caractéristiques acoustiques  $X = x_1, x_2, \dots, x_y$  et sur deux modèles probabilistes (un modèle de langage et un modèle acoustique) afin d’appliquer la fonction *argmax* et trouver la séquence de mots hypothèse  $\hat{W}$  la plus probable. Le modèle de langage renvoie la probabilité de  $W$  et le modèle acoustique estime la probabilité de  $P(X|W)$ . La qualité du système dépend essentiellement de la qualité de ces deux modèles probabilistes. La séquence de mots hypothèse  $\hat{W}$  est obtenue comme suit :

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X) \tag{1.1}$$

En appliquant la formule de Bayes, l’équation devient :

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)} \tag{1.2}$$

Comme, les paramètres acoustiques  $X$  sont fixes, l'équation peut se simplifier en :

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(X|W)P(W) \quad (1.3)$$

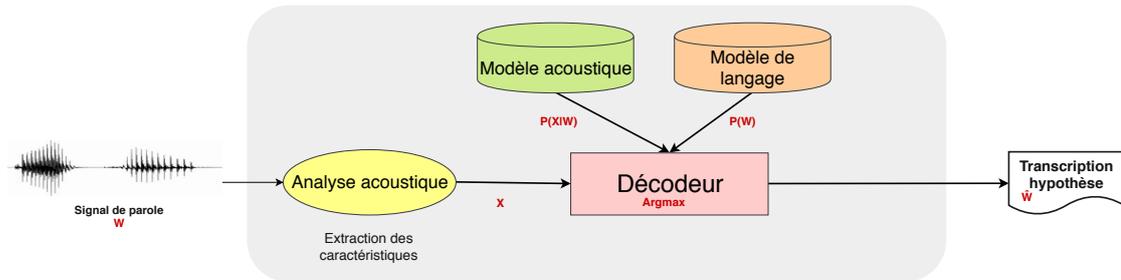


FIGURE 1.1 – Architecture d'un système de reconnaissance automatique de la parole statistique

## 1.2 Extraction des paramètres

Étant donné que le signal acoustique contient des informations autres que les mots prononcés par un locuteur (bruit, musique, etc.), l'étape d'extraction des paramètres consiste à caractériser le signal de parole en entrée afin de trouver les informations pertinentes pour les exploiter dans le système de reconnaissance et produire la séquence de mots  $\hat{W}$ . L'objectif principal est de produire des représentations vectorielles qui caractérisent le signal de parole via des fenêtres glissantes de 10 à 30 ms. Différentes techniques sont utilisées dans la littérature pour extraire les paramètres acoustiques à partir d'un signal de parole brut qui peuvent être enrichis avec leurs dérivées premières  $\Delta$  et secondes  $\Delta\Delta$  : *Mel-Frequency Cepstral Coefficients* (MFCC) [Davis and Mermelstein, 1990], *Perceptual Linear Prediction* (PLP) [Hermansky and Cox Jr, 1991] et *Linear Prediction Cepstral Coefficients* (LPCC) [Markel and Gray, 2013] [Markel et Gray, 1982].

## 1.3 Modélisation acoustique

Le modèle acoustique est un composant fondamental pour la tâche de reconnaissance de la parole. Son objectif est d'estimer la probabilité  $P(X|W)$ . Plusieurs approches de modélisation acoustique ont été proposées dans la littérature, mais

les principales sont actuellement les modèles de Markov cachés (HMM - Hidden Markov Models), les réseaux de neurones profonds (DNN - Deep Neural Network) et les modèles hybrides HMM-DNN.

### 1.3.1 Modèles de Markov Cachés

Les systèmes de reconnaissance automatique de la parole statistiques sont souvent basés sur des modèles de Markov cachés (HMM - *Hidden Markov Model*) [Jelinek, 1976, Rabiner, 1989]. Les HMM sont des automates probabilistes à états finis utilisés pour calculer la probabilité d'émission d'une séquence d'observations. Les observations sont les paramètres acoustiques extraits à partir d'un signal de parole (voir la section 1.2) qui peuvent être des MFCC, PLP, LPCC, etc.

Comme illustré dans la figure 1.2, un modèle de Markov caché est caractérisé par :

- Le nombre d'états ( $N$ )
- Un ensemble d'états d'émission  $S$  ainsi que des états de début *Départ* et de fin *Fin* qui ne sont pas liés aux observations.
- Une matrice de probabilité de transition d'état  $A$ , où chaque élément de la matrice  $a_{ij}$  est la probabilité de transition de l'état  $i$  à  $j$  pour  $i, j \in \{1, \dots, K\}$ .
- Des fonctions de densité de probabilité pour estimer la probabilité d'émettre une observation  $x_t$  à partir d'un état  $i$  à l'instant  $t$ ,  $b_i(x_t) = p(x_t | S_t = i)$ .
- Ensemble de probabilités d'état initial  $\pi = \{\pi_i = P(s_0 = i)\}$ .

Le processus d'apprentissage d'un HMM est souvent effectué à l'aide de l'algorithme de *Baum-Welch* [Baum, 1972], qui consiste essentiellement à ajuster les paramètres de la matrice de probabilité de transition d'états HMM  $A$  et les paramètres des densités de probabilité  $B$  afin de maximiser la probabilité d'observer  $X$ .

Plusieurs méthodes ont été proposées pour estimer les densités de probabilité d'émission des états HMM ( $b_i(x_t)$ ) comme : les mélanges de gaussiennes GMM/SGMM et les réseaux de neurones profonds.

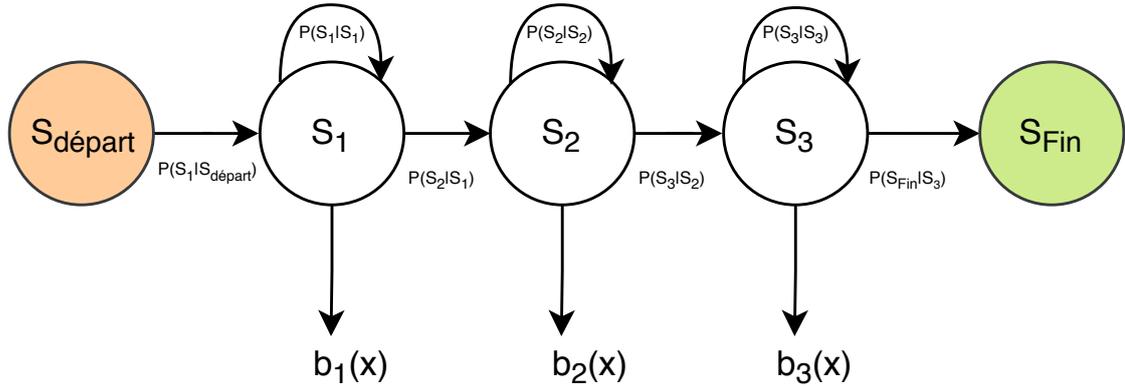


FIGURE 1.2 – Exemple d’un HMM à 3 états émetteurs.

### 1.3.2 Les modèles à mélange de gaussiennes

Un modèle à mélange de gaussiennes (GMM - *Gaussian Mixture Models*) est un modèle statistique, dont l’objectif est de représenter la densité de probabilité acoustique d’un état HMM avec une somme pondérée de  $M$  gaussiennes comme décrit dans l’équation 1.4.

$$p(x|j) = \sum_{i=1}^M w_i \mathcal{N}(x|\mu_i, \Sigma_i); \sum_{i=1}^M w_i = 1 \quad (1.4)$$

où  $x \in \mathbb{R}^D$  le vecteur des observations acoustiques de dimension  $D$  pour l’état HMM  $j$ ,  $i$  l’index de la gaussienne,  $w_i$  le poids de mélange (leur somme est égale à 1) associé à la gaussienne  $i$ ,  $\mu_i$  le vecteur moyen et  $\Sigma_i$  la matrice de covariance.  $\mathcal{N}(x|\mu_i, \Sigma_i)$  est une densité de probabilité gaussienne, définie comme suit :

$$\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right] \quad (1.5)$$

### 1.3.3 Les sous-espaces de modèle à mélange de gaussiens

Les sous-espaces de modèle à mélange de gaussiennes (SGMM - *Subspace Gaussian mixture model*) sont une modélisation acoustique introduite par Povey et al. [2011a]. Dans un modèle SGMM, chaque état HMM est représenté par un GMM mais certains paramètres sont partagés. Les probabilités des états HMM sont définies comme suit :

$$p(\mathbf{x}|j) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^I w_{jmi} \mathcal{N}(\mathbf{x}|\mu_{jmi}, \Sigma_i) \quad (1.6)$$

$$\mu_{jmi} = \mathbf{M}_i \mathbf{v}_{jm} \quad (1.7)$$

$$w_{jmi} = \frac{\exp \mathbf{w}_i^T \mathbf{v}_{jm}}{\sum_{i'=1}^I \exp \mathbf{w}_{i'}^T \mathbf{v}_{jm}} \quad (1.8)$$

Avec  $\mathbf{x} \in \mathbb{R}^D$  un vecteur des observations acoustiques pour l'état HMM  $j \in \{1..J\}$  de  $I$  gaussiennes. L'état  $j$  est modélisé par  $M_j$  sous-états avec des vecteurs de paramètres spécifique à l'état  $\mathbf{v}_{jm}$ , un mélange de poids  $c_{jm}$  (sachant que  $\sum_{m=1}^{M_j} c_{jm} = 1$ ) et la matrice de covariance  $\Sigma_i$ .

### 1.3.4 Réseaux de neurones profonds

En reconnaissance automatique de la parole statistique, les GMMs et les SGMMs sont souvent utilisés avec des HMMs pour produire la séquence de mots la plus probable. Cependant, des réseaux de neurones artificiels (ANN - Artificial Neural Networks) ont aussi été introduits pour la modélisation acoustique. L'objectif était d'estimer les états HMM à l'aide d'une seule couche cachée non-linéaire. L'apprentissage multi-couches n'était pas encore utilisé dans les algorithmes développés à cause de l'absence des machines de calcul puissantes. Les performances obtenues montraient que cette approche neuronale n'était pas efficace par rapport aux GMMs. Aujourd'hui, avec des machines plus performantes et la disponibilité des processeurs graphiques (calcul puissant), [Hinton et al. \[2012\]](#), [Seide et al. \[2011\]](#), [Dahl et al. \[2012\]](#) ont réussi à proposer des méthodes plus efficaces pour la modélisation acoustique en remplaçant les GMMs et les SGMMs par des réseaux de neurones profonds (DNN - Deep Neural Networks).

Comme illustré dans la figure 1.3, l'architecture d'un HMM/DNN est caractérisée par  $L$  couches : une couche d'entrée, une suite de couches cachées entièrement connectées ainsi qu'une couche de sortie permettant d'estimer une probabilité pour chaque état HMM pour une observation acoustique.

La sortie de chaque couche  $l \in \{1, \dots, L\}$  est définie comme suit :

$$Z_l = \varphi(W_l \cdot X + b_l) \quad (1.9)$$

Avec  $\varphi$  une fonction sigmoïde permettant de transférer les sorties d'une couche à une autre (de la première couche à la couche  $L - 1$ ),  $b_l$  un vecteur de bias,  $W_l$  une matrice de poids et  $X$  le vecteur d'entrée.

Pour la dernière couche  $L$ , La fonction  $\varphi$  correspond à une fonction *Softmax* permettant d'estimer une probabilité pour chaque état HMM  $j$  à l'instant  $t$  :

$$P(j|x_t) = \frac{e^{Z_L}}{\sum_{k=1}^N e^{Z_L}} ; j \in \{1, \dots, N\} \quad (1.10)$$

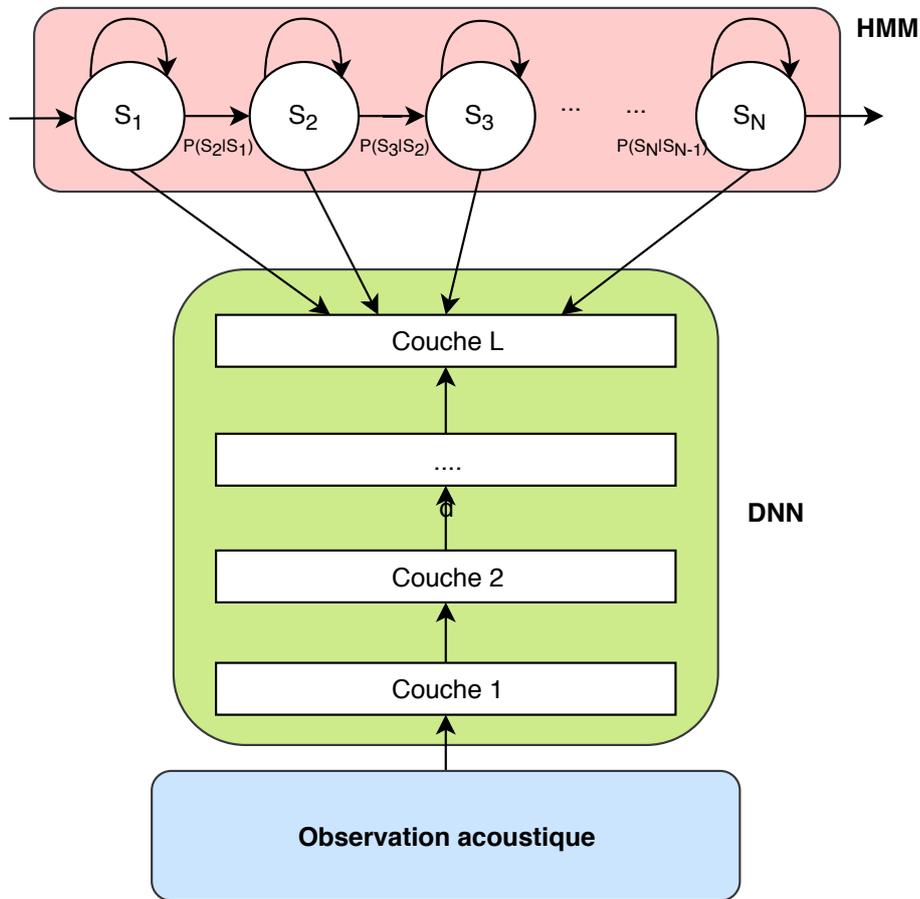


FIGURE 1.3 – Architecture d'un modèle hybride HMM/DNN

## 1.4 Modélisation linguistique

La modélisation linguistique est un élément fondamental pour les systèmes de reconnaissance automatique de la parole. Elle permet au système de RAP de

mieux décoder la séquence de mots la plus pertinente en estimant la probabilité  $P(W)$ . Plusieurs approches ont été proposées pour créer des modèles de langage comme : les modèles N-grammes, les réseaux de neurones, etc.

La modélisation linguistique n-grammes est fondée sur l'hypothèse de Markov où la probabilité d'un mot est estimée en fonction de son historique. L'historique d'un mot représente une suite de mots de taille  $K - 1$  apparus avant le mot cible. L'ordre  $N$  du modèle peut être défini en fonction des contraintes dans le corpus d'apprentissage, il est fixé généralement entre trois et cinq ( $3 \leq K \leq 5$ ). Soit la séquence de mot  $W = w_1, w_2, \dots, w_m$ , la probabilité  $P(W)$  est défini comme suit :

$$\begin{aligned} P(W) &= P(w_1, w_2, \dots, w_m) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_n|w_1, \dots, w_{m-1}) \\ &= P(w_1) \prod_{i=2}^k P(w_i|w_1 \dots w_{i-1}) \end{aligned} \tag{1.11}$$

où  $P(w_i)$  est la probabilité du mot  $w_i$  et  $P(w_i|w_1\dots w_{i-1})$  est la probabilité du mot  $w_i$  sachant son historique  $w_1\dots w_{i-1}$ .

Au moment du décodage d'une nouvelle collection de signaux, le système de reconnaissance automatique de la parole peut rencontrer des séquences de mots jamais observées dans le corpus d'apprentissage du modèle de langage. Pour éviter ce type de problème, plusieurs méthodes de lissage/repli ont été proposées dans la littérature afin d'attribuer une probabilité non nulle pour chaque séquence n-gramme non observée au moment de l'apprentissage du modèle de langage, telles que : Good-Turing [Good, 1953], Witten-Bell [Witten and Bell, 1991] and Kneser-Ney [Kneser and Ney, 1995].

Comme le modèle de langage a une forte influence sur la qualité des systèmes de RAP, il est indispensable d'évaluer sa qualité. La qualité d'un modèle de langage dépend de sa capacité à prédire les bonnes hypothèses selon le contexte lexical. Souvent, les modèles de langage sont évalués en termes de perplexité  $PPL$  [Jelinek et al., 1977]. La perplexité est calculée en mesurant l'entropie  $H$  d'un jeu de données  $S$  de  $n$  phrases (avec  $S = s_1, s_2, \dots, s_n$ ).

$$H = \frac{1}{n} \sum_{i=1}^n \log P(s_i) \tag{1.12}$$

La perplexité PPL est alors défini comme suit :

$$PPL = 2^H \tag{1.13}$$

La perplexité correspond à un indicateur de la capacité à produire le bon mot. Si elle est très élevée, le modèle de langage est incertain ; si elle est trop basse le modèle sera trop contraint. Généralement, des perplexités convenables pour le français ou l’anglais sont de l’ordre de 80 à 120.

## 1.5 Dictionnaire de prononciation

Le dictionnaire de prononciation (nommé également dictionnaire de phonétisation) représente un point clef pour l’apprentissage des modèles acoustiques probabilistes. Il a pour objectif d’associer à chaque mot du vocabulaire, la liste des variantes de prononciation possibles sous forme d’unités sonores telles que : des syllabes, des graphèmes ou des phonèmes.

La qualité du dictionnaire a une forte influence sur la qualité du système de reconnaissance automatique de la parole, par exemple, si un mot est mal phonétisé ou absent dans le dictionnaire, le système de RAP peut générer des erreurs au niveau du mot courant qui se propageront aux mots voisins.

Plusieurs approches ont été proposées dans la littérature pour créer des dictionnaires de prononciation. L’approche de phonétisation manuelle par des spécialistes, reste toujours la méthode la plus efficace. Néanmoins, cette approche est coûteuse en temps et ressources et ne peut pas couvrir tout le vocabulaire. Des méthodes automatiques sont généralement utilisées comme une approche complémentaire.

En français, *BDLEX* [Perennou and Calmes, 1987] est le dictionnaire de prononciation le plus connu et le plus exploité pour l’apprentissage des modèles acoustiques. C’est une ressource payante produite par des experts, contenant 440k formes fléchies (générées à partir de 50k mots). Pour générer des variantes de prononciation automatiquement, Béchet [2001] propose un outil nommé *LIA\_PHON* fondé sur des règles de phonétisation française et transformant les graphèmes en phonèmes.

## 1.6 Kaldi

Kaldi [Povey et al., 2011b] est une boîte à outils destinée aux chercheurs en reconnaissance automatique de la parole, développée en *C++*, disponible en ligne sous la licence *Apache v2.0*. Kaldi a mis à disposition des utilisateurs un site web<sup>1</sup> contenant les descriptions détaillées de ses fonctionnalités ainsi qu'un forum pour les contributions scientifiques. De plus, Kaldi propose plusieurs outils, techniques et recettes permettant aux chercheurs d'entraîner différents modèles acoustiques (à l'état de l'art comme les GMM, les SGMM et les DNN, etc) et des décodeurs afin de créer rapidement des systèmes de reconnaissance automatique de la parole statistiques. Kaldi inclut également plusieurs méthodes d'adaptation des modèles acoustiques comme : *Maximum Likelihood Linear Regression* [Leggetter and Woodland, 1995], *Constrained Maximum Likelihood Linear Regression* [Digalakis and Neumeyer, 1996], *Maximum A Posteriori* [Gauvain and Lee, 1994] , *Speaker Adaptive Training* [Anastasakos et al., 1996], etc.

Comme illustré dans la figure 1.4, l'architecture de la boîte à outils Kaldi est composée de 4 principaux types de composants : la librairie Kaldi *C++* qui se base essentiellement sur des bibliothèques externes optimisées pour l'algèbre linéaire comme *BLAS/LAPACK* et la librairie *OpenFST* [Allauzen et al., 2007], des exécutables Kaldi *C++* et des scripts *Shell* permettent de pré-traiter les données, d'apprendre et d'évaluer des systèmes de RAP, de visualiser les graphes, etc. La librairie *OpenFST* permet à Kaldi d'exploiter les transducteurs à états finis (FST) afin de représenter partiellement les différents modèles acoustiques avec des opérations de graphe, le modèle de langage, le modèle de prononciation, etc. Les transducteurs à états finis font de la tâche de décodage un problème de recherche heuristique dans un graphe.

## 1.7 Évaluation des SRAP

Les systèmes de reconnaissance automatique de la parole sont souvent évalués en terme de taux d'erreur de mots (WER). Le WER est obtenu à l'aide d'un algorithme d'alignement dynamique qui permet d'aligner les hypothèses de transcriptions (sorties du SRAP) avec les transcriptions de référence (créées manuellement

---

1. <http://kaldi-asr.org/doc/>

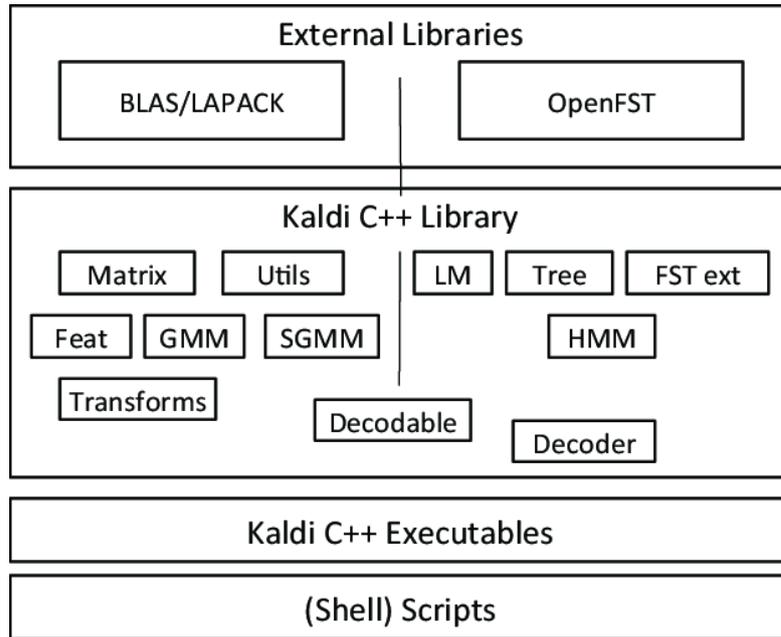


FIGURE 1.4 – Architecture de la boîte à outils Kaldi [Povey et al., 2011b]

par des humains) en dénombrant 3 types d’erreurs :

- **Suppression** (D) : le nombre de mots oubliés par le SRAP dans l’hypothèse.
- **Insertion** (I) : le nombre de mots insérés par erreur dans l’hypothèse.
- **Substitution** (S) : le nombre de mots reconnus à la place d’un mot de la transcription référence dans la transcription hypothèse.

Le WER est obtenu par la division du nombre d’erreurs ( $D + I + S$ ) par le nombre de mots dans la référence  $N$ , comme le décrit l’équation 1.14.

$$WER = \frac{D + I + S}{N} * 100 \quad (1.14)$$

## 1.8 Conclusion

Dans ce chapitre, nous avons abordé la tâche de reconnaissance automatique de la parole statistique. Nous avons commencé par présenter le fonctionnement de base d’un système de reconnaissance automatique de la parole en détaillant les principaux modules de fonctionnement. Nous avons décrit le module d’extraction des caractéristiques et les différentes techniques utilisées pour exploiter un signal de

## CHAPITRE 1. LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

---

parole dans les systèmes de reconnaissance automatique de la parole. De plus, nous avons présenté les différentes techniques d'apprentissage des modèles acoustiques probabilistes telles que : les HMMs, les GMMs, les SGMMs ainsi que les DNNs. En outre, nous avons présenté le dictionnaire de phonétisation et son importance dans le fonctionnement des systèmes de RAP. Enfin, nous avons présenté la boîte à outils Kaldi qui nous permet d'apprendre des systèmes de RAP ainsi que la métrique taux d'erreur de mots pour les évaluer.

## CHAPITRE 2

# LA PRÉDICTION DE PERFORMANCES

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>22</b>
<b>2.2</b>	<b>L'estimation des mesures de confiance</b>	<b>23</b>
<b>2.3</b>	<b>La prédiction de performances</b>	<b>25</b>
2.3.1	Principe	27
2.3.2	Granularité	28
2.3.3	Évaluation	29
2.3.4	Travaux connexes	29
<b>2.4</b>	<b>Conclusion</b>	<b>31</b>

---

L'évaluation d'un système de RAP est une opération indispensable pour déterminer la qualité des hypothèses générées. Lorsque la référence est indisponible, la prédiction de performance s'avère utile pour déterminer la fiabilité des sorties produites, notamment lorsque le système utilisé est inconnu.

Nous nous intéressons dans ce chapitre à la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole. Nous introduisons dans la section 2.1 la tâche d'estimation de la qualité des systèmes de RAP. Nous présentons par la suite dans la section 2.2 la tâche d'estimation des mesures de confiance qui représente la méthode classique pour estimer la qualité d'un système de RAP particulier. Nous décrivons ensuite, la tâche de prédiction de performance dans la section 2.3. Nous concluons enfin ce chapitre, dans la section 2.4.

### 2.1 Introduction

Depuis quelques années, la tâche de reconnaissance automatique de la parole constitue un sujet d'intérêt croissant dans des applications "grand public". On a vu ainsi émerger les systèmes de RAP dans plusieurs applications d'intelligence artificielle telles que : SIRI<sup>1</sup>, Alexa<sup>2</sup>, Microsoft Translate<sup>3</sup>, etc. Malgré les avancées spectaculaires dans le domaine, il n'existe toujours pas dans la littérature de système de RAP parfait ou robuste dans toutes conditions. Ainsi, l'évaluation automatique des systèmes est indispensable pour mesurer la fiabilité des transcriptions produites.

Comme décrit dans la section 1.7, l'évaluation automatique d'un système de RAP implique une transcription de référence (produite par des experts), une hypothèse de transcription (sortie d'un SRAP) et une métrique d'évaluation (comme le WER). Étant donné que la production d'une transcription de référence est très coûteuse (en temps et ressources), l'estimation automatique et sans référence de la qualité peut être une tâche utile pour déterminer la fiabilité *a priori* d'une transcription automatique.

Afin d'estimer la qualité des systèmes de reconnaissance automatique de la parole, de nombreux travaux ont proposé d'estimer des mesures de confiance pour

---

1. <https://www.apple.com/fr/siri/>
2. <https://developer.amazon.com/alexa-voice-service>
3. <https://translator.microsoft.com/>

détecter les erreurs dans les sorties d'un système de RAP particulier. La tâche de prédiction de performances se présente comme une nouvelle tâche, visant à prédire un taux d'erreur de mots, notamment lorsque le système de RAP est appliqué sur de nouvelles collections de signaux. L'une de ses caractéristiques par rapport à une tâche d'estimation de mesures de confiance est aussi qu'elle peut faire abstraction du fonctionnement interne du système de reconnaissance automatique de la parole.

Nous introduisons dans les sections suivantes les deux tâches d'estimation de qualité des système de reconnaissance automatique de la parole : l'estimation des mesures de confiance et la prédiction de performances.

## 2.2 L'estimation des mesures de confiance

En reconnaissance automatique de la parole, les mesures de confiance ont été introduites pour la tâche de détection des mots hors vocabulaire (OOV) par [Asadi et al. \[1990\]](#) et exploitées par [Young \[1994\]](#) pour la tâche de reconnaissance de la parole.

L'estimation des mesures de confiance est la méthode classique pour évaluer la fiabilité d'un système de reconnaissance automatique de la parole particulier. Cette opération est souvent basée sur des informations issues du fonctionnement interne d'un système de RAP particulier. Elle consiste à attribuer à chaque mot  $W$  une probabilité  $MC(W) \in [0, 1]$  pour déterminer par la suite si le mot  $W$  est « correct » ( $MC(W)$  proche de 1) ou « incorrect » ( $MC(W)$  proche de 0).

Afin d'avoir une approximation du taux d'erreur de mots (WER) à une granularité pré-définie, la moyenne  $\mu(MC)$  de  $M$  mots  $W$  étiquetés peut être calculée comme suit :

$$\mu(MC) = \frac{1}{M} \sum_{i=1}^M MC(W_i) \tag{2.1}$$

Les mesures de confiance sont très utilisées dans plusieurs applications de TAL telles que : les systèmes de dialogue [[San-Segundo et al., 2001](#), [Hazen et al., 2002](#), [Sarikaya et al., 2005](#)], la traduction automatique [[Blatz et al., 2004](#), [Quirk, 2004](#), [Ueffing and Ney, 2007](#)], la reconnaissance automatique de la parole [[Asadi et al., 1990](#), [Young, 1994](#), [Lecouteux et al., 2009](#)], etc.

**Apprentissage.** les systèmes d'estimation des mesures de confiance sont souvent appris à l'aide d'une technique d'apprentissage supervisé. En reconnaissance automatique de la parole, la probabilité  $MC(W)$  est généralement estimée à une granularité fine telle que : au niveau de phonème et au niveau de mot.

Plusieurs algorithmes d'apprentissage des systèmes d'estimation des MC ont été proposés dans la littérature tels que : le boosting [Freund et al., 1996], les arbres de décision [Brehehy, 1984], les machines à vecteurs de support (SVM) [Vapnik, 2006], les champs aléatoires conditionnels (CRF) [Lafferty et al., 2001], etc.

**Traits.** les mesures de confiance sont souvent estimées pour un système particulier. En effet, le système d'estimation est fondé essentiellement sur des traits issus du fonctionnement interne d'un système de reconnaissance automatique de la parole défini (*glass-box*) tels que : des probabilités a posteriori [Young, 1994, Mauchair, 2006], des paramètres de décodage [Fu and Du, 2005], des scores de confiance binaires reflétant la degré de fiabilité de l'hypothèse [Zhang and Rudnicky, 2001, Moreno et al., 2001], des connaissances a priori [Wiggers and Rothkrantz, 2003], les mesures se basant sur des connaissances externes [Lecouteux et al., 2009].

**Évaluation.** pour mesurer la qualité d'un système d'estimation des mesures de confiance, plusieurs métriques d'évaluation ont été proposées dans la littérature. Étant donné que cette tâche est généralement traitée comme une tâche de détection d'erreurs (en surveillant le score de confiance), plusieurs travaux ont proposé d'utiliser les métriques d'évaluation de classification standards telles que : la précision (P), le rappel (R) et le F-mesure (F1) qui sont définies comme suit :

$$P = \frac{\text{\#mots correctement annotés}}{\text{\#mots annotés}} \quad (2.2)$$

$$R = \frac{\text{\#mots correctement annotés}}{\text{\#mots à annoter}} \quad (2.3)$$

$$F1 = \frac{2 * P * R}{P + R} \quad (2.4)$$

En outre, Stemmer et al. [2002] a introduit la métrique entropie croisée normalisée (NCE - Normalized Cross Entropy) dans le cadre des évaluations NIST.

La NCE est calculée comme suit :

$$NCE = \frac{H_{max} + \sum_{correct_w} \log_2(MC(W)) + \sum_{incorrect_w} \log_2(1 - MC(W))}{H_{max}} \quad (2.5)$$

avec  $H_{max} = -m \log_2(p_c) - (M - m) \log_2(1 - p_c)$

où  $MC(W)$  la mesure de confiance attribuée à  $W$ ,  $M$  le nombre de mots annotés,  $m$  le nombre de mots correctement annotés parmi  $M$  et  $p_c = m/M$  la probabilité moyenne d'un mot correctement annoté.

## 2.3 La prédiction de performances

Prédire la performance d'un système de reconnaissance automatique de la parole sur de nouveaux enregistrements (par exemple de nouveaux types de programmes TV ou radio jamais rencontrés auparavant) est un Graal important de la reconnaissance automatique de la parole, notamment si le système de RAP est inconnu (boîte noire). En effet, cette opération consiste à prédire un score (comme le taux d'erreur de mots) à chaque hypothèse de transcription produite par un système inconnu lorsque les transcriptions références sont indisponibles. Le score prédit reflète la qualité des transcriptions produites par un système de RAP (boîte noire) au niveau d'une granularité pré-définie (mot, phrase, document ...).

La tâche de prédiction des performances va au-delà de l'estimation de confiance puisqu'elle ne se concentre ni sur un système de reconnaissance automatique de la parole particulier (ni sur des treillis ou des N-meilleures hypothèses) ni sur la transcription référence (humaine). Elle a pour but, de donner une estimation générale de la difficulté de la tâche de transcription pour un système de RAP inconnu.

Tandis que la tâche d'estimation de confiance est utilisée pour prédire une probabilité (entre 0 et 1) ou une classe (correcte/incorrecte), la tâche de prédiction de performances consiste principalement à prédire un score tel que le taux d'erreur de mots (WER).

Le tableau 2.1 résume les principales différences entre l'estimation des mesures de confiance et la prédiction de performances en termes de : type de traits, type de sortie, la granularité standard (la plus utilisée dans la littérature), méthode d'apprentissage, algorithme d'apprentissage et métrique d'évaluation.

	Mesures de confiances	Prédiction de performances
Type de traits	<u>SRAP</u> ; acoustiques; textuels	acoustiques; textuels
Sortie	une probabilité (entre 0 et 1); une classe (correct, incorrect)	un taux d'erreur de mots (entre 0 et $+\infty$ )
Granularité standard	phonème, mot	tour de parole ( <i>utterance</i> ) ou document
Méthode	classification, régression	régression
Algorithme	Naïve bayes; champ aléatoire conditionnel (CRF); boosting; réseau de neurones; etc.	régression linéaire; arbre de régression; régression par SVM; réseaux de neurones.
Métrique	Précision/Rappel/F-mesure; entropie croisée normalisée (NCE);	MAE (voir l'équation 2.6); RMSE (voir l'équation 2.7)
Application	estimer l'effort de post-édition pour produire une transcription référence; trier les N meilleures hypothèses d'un SRAP particulier; Ré-appliquer le processus de décodage en utilisant les mesures de confiance dans le graphe de recherche de décodage; exploiter les étiquettes de mots pour les intégrer dans une nouvelle application comme la traduction automatique;	le système de RAP est une boîte noire; prédire un taux d'erreur de mots d'une nouvelle collection de données (jamais rencontrée auparavant); estimer l'effort de post-édition pour produire une transcription référence; estimer le coût de création/d'adaptation d'un système de RAP; trier les hypothèses de transcription de plusieurs SRAP (des boîtes noires)

TABLE 2.1 – Les principales différences entre l'estimation des mesures de confiance et la prédiction de performances

Les principaux objectifs de la tâche de prédiction de performances sont les suivants :

- Prédire les performances d’une nouvelle collection de données transcrite par un système de RAP inconnu (boîte noire )
- Estimer l’effort de post-édition/révision demandé à l’annotateur humain pour produire une transcription de référence
- Trier les hypothèses de transcription d’un signal acoustique lorsqu’il est transcrit par plusieurs systèmes de RAP boîte noire afin de sélectionner les meilleures transcriptions
- Estimer le coût d’adaptation d’un système de RAP appris sur une tâche spécifique à une nouvelle tâche

### 2.3.1 Principe

Les systèmes de prédiction de performances sont souvent appris avec une technique d’apprentissage de régression supervisée qui exige un corpus d’apprentissage sous forme de couple  $\{X, Y\}$ , où  $Y = [y_1, \dots, y_N]$  donne les vraies performances de l’entrée  $X = [x_1, \dots, x_N]$  à une granularité pré-définie. En reconnaissance automatique de la parole, les données  $X$  peuvent être des données de type : acoustique uniquement, textuel uniquement ou un couple {acoustique, texte}.

Étant donné que les données textuelles/acoustiques ne sont pas exploitables directement au moment de l’apprentissage, les systèmes de prédiction sont fondés sur l’extraction de traits pré-définis (traits textuels, traits acoustiques, etc.) des entrées  $X$  ; qui nécessitent des outils et des ressources spécifiques à la tâche de prédiction.

Comme illustré dans la figure 2.1, l’apprentissage des systèmes prédictifs s’appuie essentiellement sur l’extraction des caractéristiques associées à un algorithme d’apprentissage. Plusieurs algorithmes d’apprentissage ont été proposés dans la littérature tels que : la régression linéaire, la régression par SVM, les arbres de régression, etc.

Une fois que le système de prédiction est appris, il peut être exploité afin de prédire les performances d’une nouvelle collection de données tout en passant par le même processus d’extraction des caractéristiques utilisé durant l’apprentissage (comme décrit dans la figure 2.2).

La pertinence des caractéristiques extraites et le choix d’un algorithme d’apprentissage adéquat pour la tâche constituent un vrai défi. Ces choix ont une

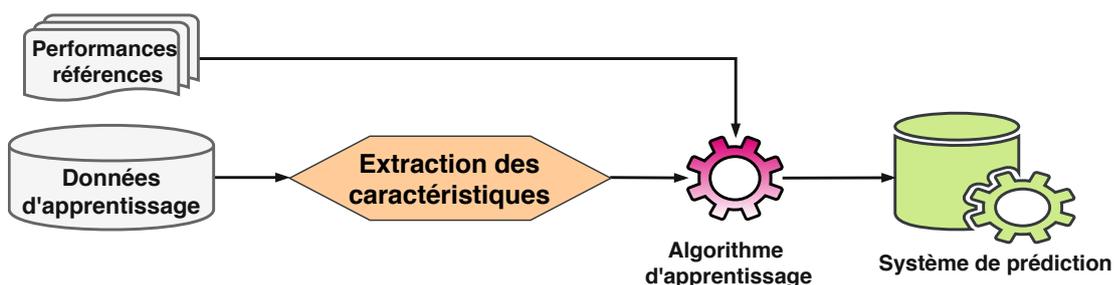


FIGURE 2.1 – Processus d'apprentissage des systèmes de prédiction de performances

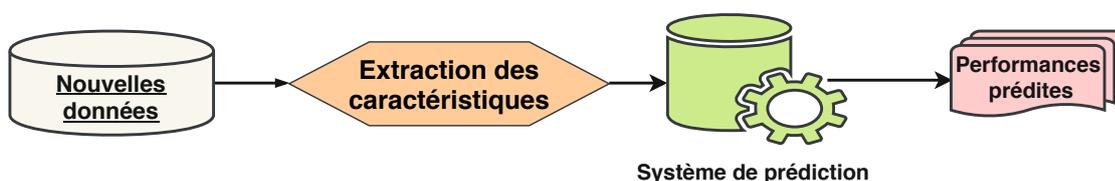


FIGURE 2.2 – Processus de prédiction de performances d'une nouvelle collection de données

incidence directe sur la qualité du système.

### 2.3.2 Granularité

Nous trouvons généralement deux niveaux de prédiction de taux d'erreur de mots des systèmes de reconnaissance automatique de la parole :

- **Tour de parole** (*utterance*) : c'est la granularité de prédiction la plus utilisée dans la littérature. L'objectif est d'associer à chaque tour de parole un score (comme le taux d'erreur de mots) qui reflète la performance de transcription automatique.

Cette granularité est souvent utilisée pour estimer l'effort de révision/post-édition demandé à un annotateur humain pour produire une transcription référence (réduire le coût d'annotation manuelle) ;

- **Document** : c'est une granularité plus large que la prédiction au niveau d'un tour de parole. Elle consiste à associer un score au niveau de chaque document en entrée qui représente un ensemble de tours de parole. En reconnaissance automatique de la parole, un document peut être : un type d'émission, un style de parole (spontanée/préparée), un corpus complet, etc.

Ce type de granularité est souvent utilisé afin de savoir quels sont les types d'émission les plus difficiles à transcrire, et d'estimer le coût d'adaptation d'un système de RAP pour une meilleure transcription.

### 2.3.3 Évaluation

Les systèmes de prédiction de performances (PP) sont souvent évalués en termes d'erreur absolue moyenne (MAE - *Mean Absolute Error*). Le MAE consiste à calculer la moyenne arithmétique des valeurs absolues des écarts entre les performances de référence et les performances prédites par le système automatique. De plus, l'erreur quadratique moyenne (RMSE - *Root Mean Square Error*) peut être utilisée pour mesurer la qualité des systèmes prédictifs.

Étant donné que l'objectif est de mesurer l'erreur produite par le système PP, plus le MAE (ou le RMSE) obtenu est faible et plus la qualité de prédiction est bonne, signifiant ainsi que les performances prédites sont plus proches de la référence.

Soit  $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_N]$  la liste des performances prédites et  $Y = [y_1, \dots, y_N]$  les performances de référence pour  $N$  unités, le MAE et le RMSE se calculent comme suit :

$$MAE = \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{N} \quad (2.6)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (2.7)$$

[Willmott and Matsuura \[2005\]](#) montrent que le MAE est une mesure plus naturelle que le RSME pour évaluer les systèmes prédictifs. De plus, ils prouvent que le RMSE est une mesure inappropriée et mal interprétée de l'erreur moyenne. En effet, il varie en fonction de 3 caractéristiques (la variabilité de la distribution des grandeurs d'erreur, la racine carrée du nombre d'erreurs, ainsi que de la magnitude de l'erreur moyenne).

### 2.3.4 Travaux connexes

Plusieurs travaux se fondent essentiellement sur des traits acoustiques pour prédire les performances, [Hermansky et al. \[2013\]](#) exploitent des caractéristiques

temporelles du signal vocal (*Mean Temporal Distance* – calculées sur le signal et corrélées avec le rapport signal-sur-bruit) pour prédire la performance.

Sébastien et al. [2018] proposent d’analyser le comportement de l’énergie à court terme du bruit et de la parole en tenant compte de divers facteurs tandis que le système RAP est considéré comme une boîte noire. Les auteurs comparent deux approches de régression (MLP et linéaire) en prenant en compte la variabilité des systèmes de RAP en fonction du volume et du type de bruit. Les performances obtenues montrent que la régression MLP est meilleure que la régression linéaire.

Meyer et al. [2017] proposent une méthode de prédiction de performances de RAP apprise avec des données propres et évaluée sur 10 types de bruits inconnus ainsi qu’une large gamme de rapports signal/bruit sur les corpus DRE01 Dreschler et al. [2001], Noisex et BBC sound effects. Les résultats montrent que le bruit dans les données influence la qualité des systèmes de prédiction.

Negri et al. [2014] proposent d’autres types de traits (autres que le signal) comme les informations internes du SRAP, des caractéristiques acoustiques, des caractéristiques hybrides et des caractéristiques textuelles. Trois scénarios de prédiction ont été proposés afin d’étudier l’impact de la présence (*glass-box*) ou l’absence (*black-box*) de caractéristiques particulières extraites des SRAP, ainsi que l’effet de l’homogénéité/non-homogénéité des données d’apprentissage et d’évaluation sur la qualité des systèmes de prédiction. Les performances obtenues montrent que le système appris sur des traits *glass-box* est légèrement meilleur que le système appris sur des traits *black-box*. La qualité des systèmes de prédiction dépend de l’homogénéité entre les données d’entraînement et les données d’évaluation.

Jalalvand et al. [2016] ont proposé un outil open-source nommé *TranscRater* qui se fonde essentiellement sur l’extraction de traits (caractéristiques phonétiques, syntaxiques, acoustiques et d’un modèle de langue) et qui utilise un algorithme basé sur une régression pour prédire un taux d’erreur. Le SRAP est considéré comme une « boîte noire », et l’évaluation a été effectuée sur les données de CHiME-3.<sup>4</sup> Dans ce travail, les expérimentations montrent que les caractéristiques acoustiques (issues directement du *signal*) n’ont pas d’influence forte sur la qualité du système de prédiction.

Les travaux présentés précédemment s’appuient sur des traits (ou *features*) pré-définis qui exigent des outils et des ressources spécifiques (pour les traits textuels)

---

4. [http://spandh.dcs.shef.ac.uk/chime\\_challenge/chime2015/](http://spandh.dcs.shef.ac.uk/chime_challenge/chime2015/)

pour une langue donnée afin de prédire la performance.

## 2.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole. Nous avons commencé par présenter la tâche d'estimation des mesures de confiance qui représente la tâche classique pour estimer la qualité d'un système de RAP particulier. Ensuite, nous avons introduit la tâche de prédiction de performances pour présenter le principe de fonctionnement des système de prédiction de performances, les granularités de prédiction ainsi que les métriques d'évaluation. Enfin, nous avons passé en revue quelques travaux existants sur la tâche de prédiction.

Tandis que la plupart des travaux ont proposé d'utiliser des traits pré-définis, nous visons dans ce manuscrit à proposer une méthode flexible (ne dépendant pas de la langue) qui se fonde sur des traits appris au cours de l'apprentissage du système à l'aide des réseaux de neurones. Dans ce but, nous introduisons dans le chapitre suivant les réseaux de neurones convolutifs en traitement automatique des langues.



## CHAPITRE 3

# LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>34</b>
<b>3.2</b>	<b>Un neurone formel</b>	<b>36</b>
<b>3.3</b>	<b>Extraction et apprentissage des traits</b>	<b>37</b>
3.3.1	L'entrée du réseau	37
3.3.1.1	Traitement de texte	37
3.3.1.2	Traitement de signal acoustique	39
3.3.2	La convolution	40
3.3.3	Le <i>pooling</i>	43
<b>3.4</b>	<b>Modélisation et prédiction</b>	<b>45</b>
<b>3.5</b>	<b>Apprentissage d'un réseau de neurones</b>	<b>46</b>
<b>3.6</b>	<b>Conclusion</b>	<b>49</b>

---

**A**u cours de ces dernières années, les avancées dans le domaine de l'apprentissage automatique ainsi que la disponibilité de dispositifs de calcul puissants ont mené à des méthodes plus efficaces pour l'apprentissage des applications de TAL se basant sur des réseaux de neurones profonds. C'est le cas dans le domaine de la reconnaissance automatique de la parole [Graves et al., 2013, Graves and Jaitly, 2014, Dahl et al., 2012], la traduction automatique [Sutskever et al., 2014, Bahdanau et al., 2014, Wu et al., 2016, Crego et al., 2016], la classification de texte [Collobert and Weston, 2008, Kim, 2014, Lai et al., 2015] ou la classification d'images [Krizhevsky et al., 2012, LeCun et al., 1990].

Dans ce chapitre, nous nous intéressons uniquement aux réseaux de neurones convolutifs. Nous décrivons dans la section 3.2 le fonctionnement ainsi que les différentes caractéristiques d'un perceptron formel simple. Nous introduisons ensuite dans la section 3.3, le fonctionnement de base d'un réseau de neurones convolutif simple, en présentant ses différentes caractéristiques pour le traitement automatique des langues : l'entrée du réseau, l'extraction des caractéristiques, la modélisation et la prédiction. Nous présentons également dans la section 3.5, le principe de base du processus d'apprentissage, les principaux paramètres ainsi que les différentes méthodes de régularisation, avant de conclure dans la section 3.6.

## 3.1 Introduction

Les réseaux de neurones convolutifs (CNN ou ConvNet - Convolutional Neural Network) sont un type particulier de réseaux de neurones multi-couches *feed-forward* souvent utilisés en traitement d'images, introduits initialement par Fukushima [1980] pour une tâche de reconnaissance de forme dont l'architecture du réseau est inspirée du cortex visuel des animaux [Hubel and Wiesel, 1962], et popularisés par LeCun et al. [1990] pour la tâche de la reconnaissance de caractères.

Aujourd'hui, les CNNs sont devenus populaires dans plusieurs applications d'intelligence artificielle en traitement automatique des langues : classification de texte [Collobert et al., 2011, Kim, 2014] et classification de musique et d'environnements sonores [Palaz et al., 2015, Dai et al., 2017].

La différence d'implémentation des réseaux de neurones convolutifs en traitement d'images et en traitement automatique des langues réside principalement au

niveau de l'extraction des traits par les *filtres*. En traitement d'images, l'extraction des caractéristiques s'effectue sur des petites régions de l'image, dont chaque *filtre* se déplace dans deux directions (horizontalement et verticalement) sur l'entrée du réseau. Mais, en traitement automatique des langues, le *filtre* couvre une suite de mots ou une suite de séries temporelles, et il ne se déplace que dans une seule direction (horizontalement) pour l'extraction des caractéristiques.

Contrairement aux approches d'apprentissage standards qui sont fondées sur des caractéristiques pré-définies, les réseaux de neurones convolutifs sont capables de détecter, d'extraire et d'apprendre des traits spécifiques adaptés à la tâche visée au moment de l'apprentissage sans avoir besoin de ressources, d'implémentation des algorithmes ou d'outils pour extraire des « traits pré-définis » (*engineered features*).

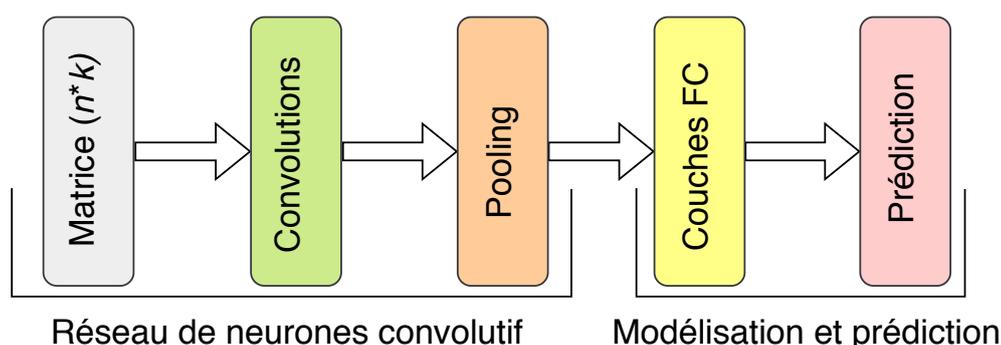


FIGURE 3.1 – Les principaux blocs de construction d'une architecture utilisant un réseau de neurones convolutif simple

Comme illustré dans la figure 3.1, l'architecture d'un réseau de neurones convolutif simple est caractérisée principalement par deux blocs :

- **Extraction des caractéristiques** : ce bloc est composé d'une entrée de dimension  $n \times k$  qui représente une instance (une séquence de mots ou un signal de parole dans notre cas d'usage), des opérations de convolution permettant d'extraire un grand nombre de caractéristiques qui seront par la suite compressées par des opérations de *pooling* (appelées aussi sous-échantillonnages).
- **Modélisation et prédiction** : ce bloc est caractérisé par une suite de couches cachées entièrement connectées pour prédire une unité qui peut être une catégorie ou une valeur, selon la tâche.

## 3.2 Un neurone formel

Un réseau de neurones est un ensemble de neurones formels fortement interconnectés permettant de recevoir des signaux et de les modéliser afin de produire une seule sortie. Le neurone formel a été inventé par [McCulloch and Pitts \[1943\]](#) en proposant une première modélisation simplifiée d'un neurone biologique et prouvant qu'un neurone formel est capable de réaliser des fonctions logiques et arithmétiques. Cela a permis au psychologue [Rosenblatt \[1958\]](#) de proposer le modèle du perceptron appliqué pour la tâche de reconnaissance de forme.

Comme décrit dans la figure 3.2, un neurone formel est caractérisé par des entrées  $X = \{x_1, x_2, \dots, x_n\}$ , leurs poids synaptiques  $W = \{w_1, w_2, \dots, w_n\}$  qui reflètent l'importance d'un neurone à un autre, une fonction d'activation (appelée aussi fonction de transfert) appliquée sur la somme pondérée de  $X$  et  $W$ , un biais  $b$  qui permet d'ajouter la flexibilité au moment de l'apprentissage ainsi qu'une sortie  $y$  (axone) qui peut être considérée comme une sortie finale ou une entrée vers d'autres neurones. La sortie  $y$  d'un neurone formel est obtenue en calculant la fonction  $f(X)$  comme suit :

$$y = \varphi(W \cdot X + b) = \varphi\left(\sum_{i=1}^n w_i \cdot x_i + b\right) \quad (3.1)$$

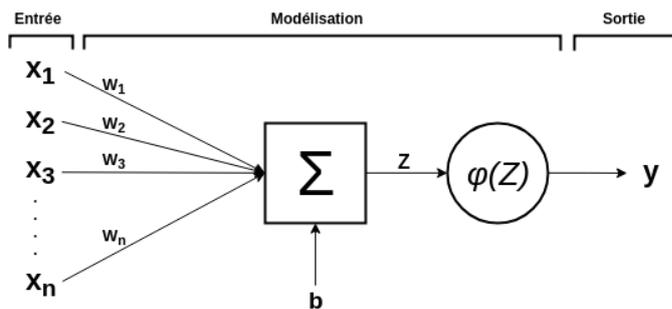


FIGURE 3.2 – Architecture d'un neurone formel

Soit  $Z = W \cdot X + b$ , les principaux types de fonctions d'activation  $\varphi$  proposées dans la littérature sont :

— Sigmoidé :

$$\varphi(Z) = \frac{1}{1 + e^{-Z}} \rightarrow \in [0, 1] \quad (3.2)$$

— Tangente hyperbolique :

$$\varphi(Z) = \tanh(Z) = \frac{e^Z - e^{-Z}}{e^Z + e^{-Z}} \rightarrow \in [-1, 1] \quad (3.3)$$

— Identité (ou linéaire) :

$$\varphi(Z) = Z \rightarrow \in [-\infty, +\infty] \quad (3.4)$$

— ReLu (Rectified linear unit) :

$$\varphi(Z) = Z^+ = \max(0, Z) \rightarrow \in [0, +\infty[ \quad (3.5)$$

### 3.3 Extraction et apprentissage des traits

Comme décrit dans la figure 3.1, le processus d'extraction et d'apprentissage des traits implique principalement : une grille unidimensionnelle ou multidimensionnelle en entrée du réseau, des couches de convolution pour caractériser l'entrée et une couche de *pooling* permettant de sous-échantillonner la sortie de la couche de convolution afin de réduire le grand nombre de traits appris.

#### 3.3.1 L'entrée du réseau

Les réseaux de neurones convolutifs sont souvent utilisés lorsque l'entrée est présentée sous forme d'une grille [Goodfellow et al., 2016] : une grille unidimensionnelle 1D pour le traitement d'un signal acoustique brut, une grille bidimensionnelle 2D pour le traitement d'images et le traitement de texte, ou une grille tridimensionnelle 3D pour le traitement de vidéos.

Nous nous intéressons dans cette section à présenter le processus d'adaptation et de transformation d'une séquence de mots ou d'un signal acoustique en une grille à l'entrée d'un réseau de neurones convolutif (convolution 1D).

##### 3.3.1.1 Traitement de texte

Étant donné que l'entrée du réseau est une séquence de mots ayant une structure unidimensionnelle, Collobert and Weston [2008] proposent d'utiliser une couche d'*embeddings* (appelée aussi *lookup table*) pour transformer cette séquence de mots

### CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

---

en une grille de mots de dimension fixe  $A \in \mathbb{R}^{n \times k}$  en exploitant les représentations vectorielles de mots (*embeddings*), où  $n$  est le nombre de mots et  $k$  un hyperparamètre fixé par l'utilisateur qui représente la dimension des représentations vectorielles des mots. La couche d'*embeddings* *EMBED* est une matrice de poids  $W \in \mathbb{R}^{V \times k}$  apprenable, où chaque ligne représente un mot d'un vocabulaire *Vocab* de taille  $V$ .

La représentation vectorielle d'un mot  $m \in \text{Vocab}$  est obtenue par la couche d'*embeddings* *EMBED* comme suit :

$$\text{EMBED}(m) = W_m \tag{3.6}$$

avec  $W_m$  la représentation vectorielle correspondante du mot  $m$  de dimension  $1 \times k$ .

Soit une séquence  $M$  de  $n$  mots  $M = \{m_1, m_2, \dots, m_n\}$ , la sortie de la couche *EMBED* est une représentation matricielle  $A \in \mathbb{R}^{n \times k}$  définie comme suit :

$$A = \text{EMBED}(M) = \begin{pmatrix} W_{m_1} \\ W_{m_2} \\ \dots \\ W_{m_n} \end{pmatrix} \tag{3.7}$$

Les paramètres de la couche *EMBED* sont généralement initialisés aléatoirement ou à l'aide d'un modèle d'*embeddings* pré-entraîné sur une grande quantité de données en exploitant les outils *word2vec* [Mikolov et al., 2013b] ou *GloVe* [Pennington et al., 2014]. Ces paramètres seront par la suite mis à jour à l'aide de l'algorithme de rétro-propagation au moment de l'apprentissage.

Le nombre de mots des instances en entrée est souvent unifié et fixé à  $n$  par une opération de *Padding*. Cette opération est appliquée sur chaque instance  $i$  ayant un nombre de mots  $l_i < n$  en ajoutant  $n - l_i$  fois un symbole spécial hors-vocabulaire (comme  $\langle PAD \rangle$  dans la figure 3.3) à la fin de la séquence de mots avec une représentation vectorielle de dimension  $k$  initialisée à zéro. Si  $l_i > n$ , les  $n$  premiers mots de  $i$  seront sélectionnés.

Par exemple, comme décrit dans la figure 3.3, pour  $n = 6$ , l'instance "oui c'est vrai" est de longueur  $4 < n$ , nous rajoutons alors 2 fois le symbole  $\langle PAD \rangle$  à la fin de l'instance. Ensuite, une opération de correspondance est effectuée par la couche *EMBED* en cherchant la représentation vectorielle de chaque mot initialisé par un

modèle *Word2vec* pré-entraîné afin de construire la matrice  $A$ .

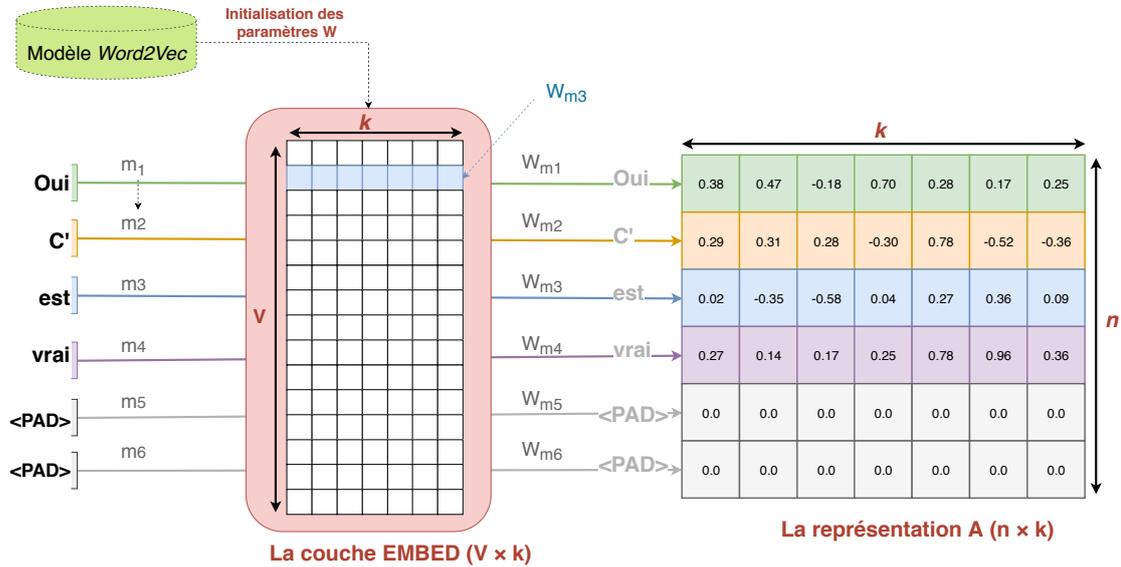


FIGURE 3.3 – Processus de transformation d’une séquence de mots en une représentation matricielle par la couche *EMBED*

### 3.3.1.2 Traitement de signal acoustique

Le traitement des données acoustiques à l’entrée des réseaux de neurones convolutifs est utilisé dans différentes tâches telles que : la classification d’environnements sonores, la classification de styles de musique, l’identification de langue/locuteurs, etc.

Plusieurs travaux ont proposé d’utiliser un signal de parole brut à l’entrée CNN pour résoudre une tâche donnée. Comme décrit dans la figure 3.4, un signal de parole présente une matrice  $S$  unidimensionnelle de taille  $n \times 1$  (où  $n$  = fréquence d’échantillonnage  $\times$  durée ;  $k = 1$ ). Sachant que chaque unité de  $A$  représente la valeur de l’amplitude en fonction du temps.

Comme les CNNs prennent en entrée des instances de taille fixe, la durée des signaux de parole doit être unifiée. Par exemple pour une durée fixée à 5 secondes et une fréquence d’échantillonnage de 8000Hz, nous obtenons une représentation de taille  $40000 \times 1$ . Soit  $l_i$  la représentation acoustique obtenue de l’instance  $i$ , si  $l_i > n$  alors les  $n$  premiers éléments seront sélectionnés, sinon une opération de *zero-Padding* (correspondant à un silence) est appliquée en ajoutant  $n - l_i$  unités initialisées à 0.

## CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

Après avoir unifié la durée des signaux bruts, plusieurs travaux ont proposé d'utiliser des paramètres acoustiques à l'entrée d'un réseau de neurones convolutif comme les MFCC, les PLP, etc. (voir la section 1.3). Par exemple pour une transformation de signal en MFCC, nous obtenons une matrice 2D de dimension  $n \times k$ , dont  $n$  la durée de la piste dans les trames et  $k$  le nombre de MFCC.

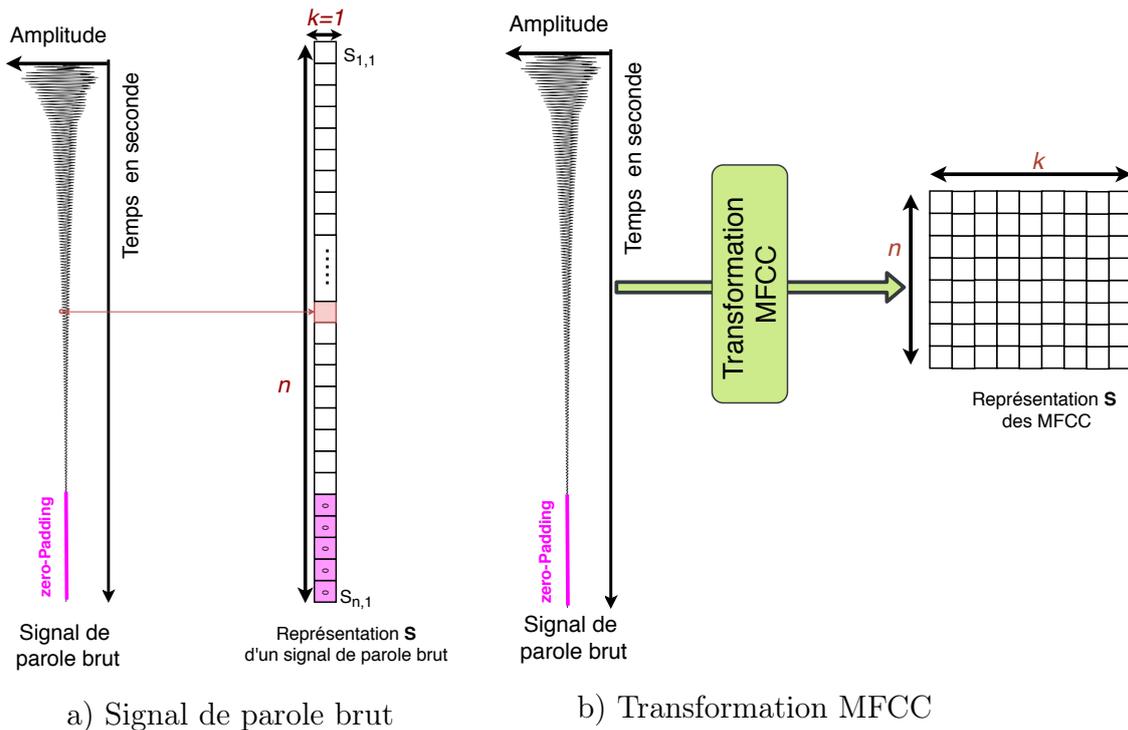


FIGURE 3.4 – Exemple de d'utilisations d'un signal acoustique à l'entrée des réseaux de neurones convolutifs

### 3.3.2 La convolution

La convolution est une opération fondamentale pour les réseaux de neurones convolutifs, c'est l'opération fondamentale des premières couches du réseau qui permettent d'extraire des informations spécifiques caractérisant l'entrée  $A \in \mathbb{R}^{n \times k}$ . Il s'agit d'une opération mathématique qui consiste à appliquer une succession de *filtres*  $W$  (appelés aussi matrices de convolution) sur un ensemble de régions de convolution de l'entrée  $A \in \mathbb{R}^{n \times k}$  par un principe de fenêtre glissante afin de produire en sortie un ensemble de cartes de caractéristiques (appelées aussi cartes d'activations ou *feature map*).

### CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

---

Les filtres de convolution correspondent à des poids initialisés aléatoirement puis mis à jour par l'algorithme de *rétro-propagation du gradient* au moment de l'apprentissage. La dimension des filtres et leurs directions de mouvement varient en fonction de la tâche visée et en fonction de la dimension de l'opération de convolution. Comme décrit dans la figure 3.5, il existe 3 types d'opérations de convolution avec différentes dimensions 1D, 2D et 3D. La différence entre ces types de convolution s'exprime essentiellement au niveau des tailles des filtres, du nombre de direction de mouvement des filtres sur l'entrée  $A$  ainsi que de la dimension de la carte de caractéristiques produite en sortie.

- **Convolution 1D** : cette opération de convolution est souvent utilisée en traitement automatique des langues (en traitement de texte et en traitement de parole). Le filtre est caractérisé par une hauteur  $h$  (correspondant à  $h$  mots ou  $h$  séries temporelles) et une largeur  $l$  qui correspond à la largeur d'entrée  $l = k$ . Ainsi, le filtre ne peut se déplacer que dans une seule direction (de haut en bas comme illustré dans la figure 3.5) avec  $s$  pas de déplacement afin de fournir en sortie une carte de caractéristiques 1D ;
- **Convolution 2D** : cette opération est généralement appliquée en traitement d'images, où la taille d'un filtre dépend essentiellement de la dimension de l'entrée. Par exemple, pour une image 2D, le filtre se caractérise par une hauteur  $h < n$  et une largeur  $l < k$ . Pour une image 3D, le filtre comporte en plus une profondeur  $f$  ayant la même taille que celle de l'entrée  $f = p$ . Ainsi, la convolution est appliquée sur des régions de l'image avec un filtre qui se déplace uniquement dans 2 directions (de gauche à droite et de haut en bas) avec un pas de déplacement pour chaque direction en produisant en sortie une carte de caractéristiques 2D ;
- **Convolution 3D** : cette opération est souvent utilisée en traitement de vidéos en appliquant un filtre de hauteur  $h < n$ , de largeur  $l < k$  et de profondeur  $f < p$  sur des régions de l'entrée 3D. Comme illustré dans la figure 3.5, le filtre se déplace dans 3 directions : de gauche à droite, de haut en bas et en profondeur. Chaque direction comporte un pas de déplacement qui caractérise le déplacement du filtre sur l'entrée.

Nous nous intéressons dans notre travail à la convolution 1D où le filtre  $W \in \mathbb{R}^{h \times k}$  appliqué traite à la fois des séquences de  $h$  mots ainsi que leurs *embeddings*

## CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

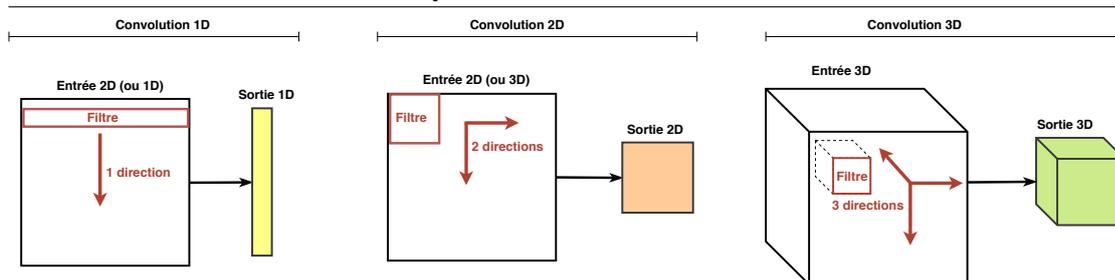


FIGURE 3.5 – Exemple des différentes dimensions des opérations de convolution

de taille  $k$  pour les entrées textuelles, et des séries temporelles de longueur  $h$  et de largeur  $k$  pour les entrées acoustiques.

Nous décrivons dans la figure 3.6 le fonctionnement détaillé d’une couche de convolution 1D appliquée sur l’entrée du réseau  $A \in \mathbb{R}^{n \times k}$  de la couche précédente. Comme illustré, chaque couche de convolution implique :

- Un ensemble de **filtres** qui représente un ensemble de matrices de poids  $W$  ayant la même dimension  $h \times k$ , où  $h$  une hauteur fixée par l’utilisateur (hyper-paramètre) et  $k$  une largeur fixée correspondant à la largeur de l’entrée  $A$  ;
- Une **fenêtre de convolution** (ayant la même dimension que le filtre :  $h \times k$ ) identifiée par  $i$  et représentée par  $A_{i:i+h-1}$  permettant de sélectionner  $h$  unités (mots ou séries temporelles) de la matrice en entrée  $A$  de la ligne  $i$  à la ligne  $i + h - 1$  ;
- Un **pas de déplacement**  $s \geq 1$  permettant la fenêtre de convolution de se déplacer de haut en bas sur la matrice de l’entrée  $A$ .

Ainsi, chaque *filtre*  $W$  est appliqué sur chaque fenêtre de convolution possible en calculant le produit de convolution. Chaque opération de convolution produit en sortie un vecteur unidimensionnel de  $T_o$  valeurs  $\mathbf{o} = [o_1, o_2, \dots, o_{T_o}] \in \mathbb{R}^{T_o \times 1}$ , dont la valeur de  $T_o$  est déterminée comme suit :

$$T_o = \frac{n - h}{s} + 1 \quad (3.8)$$

avec  $n$  la hauteur de l’entrée  $X$ ,  $h$  la hauteur du filtre  $W$  et  $s$  le nombre de pas de déplacement de la fenêtre de convolution.

## CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

La sortie de la  $i^{\text{ème}}$  opération de convolution est calculée comme suit :

$$o_i = W \cdot A_{i:i+h-1} + b \quad (3.9)$$

Chaque couche de convolution doit être par la suite suivie d'une fonction de non-linéarité  $\varphi$  (voir la section 3.2) afin d'adapter et de transférer les traits appris vers les couches suivantes du réseau. Cette opération produit la carte de caractéristiques (*feature map*)  $\mathbf{c} = [c_1, c_2, \dots, c_{T_o}] \in \mathbb{R}^{T_o \times 1}$ , où la  $i^{\text{ème}}$  valeur de  $\mathbf{c}$  est définie comme suit :

$$c_i = \varphi(o_i) \quad (3.10)$$

Une couche de convolution peut avoir  $F$  (un hyper-paramètre défini par l'utilisateur) *filters* ayant la même dimension. La couche de convolution produit en sortie  $F$  cartes de caractéristiques qui peuvent être par la suite transférées vers une nouvelle couche de convolution ou vers une couche de *pooling*.

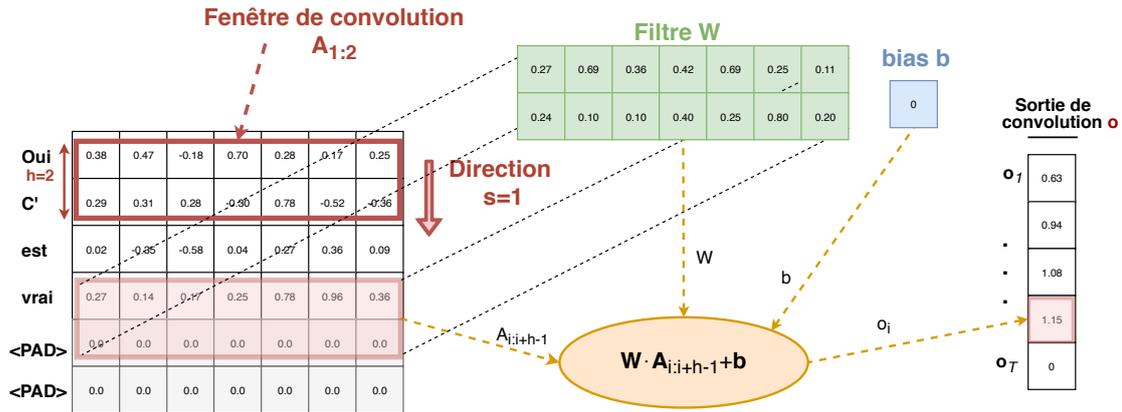


FIGURE 3.6 – Exemple d'application d'une convolution 1D sur une grille de mots  $A$

### 3.3.3 Le *pooling*

L'opération de *pooling* (appelée aussi sous-échantillonnage) est un élément clef pour les réseaux de neurones convolutifs, car, elle permet d'améliorer le processus d'apprentissage du réseau et d'éviter le sur-apprentissage en réduisant progressivement les tailles des cartes de caractéristiques obtenues durant les couches de convolution tout en gardant les informations les plus pertinentes.

### CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

---

Comme pour les opérations de convolution, 3 dimensions de *pooling* (1D, 2D et 3D) ont été proposées dans la littérature en fonction de la tâche visée. La dimension de *pooling* correspond souvent à la dimension de l'opération de convolution effectuée. Étant donné que nous appliquons des convolutions 1D sur nos entrées textuelles et acoustiques, nous nous intéressons ainsi à des opérations de *pooling* 1D.

L'opération de *pooling* 1D est caractérisée par une fenêtre de *pooling* de hauteur  $h_p$  (hyper-paramètre défini par l'utilisateur) qui se déplace dans une seule direction (de haut en bas) avec  $s_p$  pas de déplacement sur chaque carte de caractéristiques  $\mathbf{c} \in \mathbb{R}^{T_o \times 1}$  produite par la couche de convolution. Elle est souvent abordée par deux approches principales :

- *Max-pooling* : consiste à retourner la valeur maximale locale au niveau de chaque fenêtre de *pooling* ;
- *Avg-pooling* : permet de calculer la moyenne des valeurs locales de chaque fenêtre de *pooling*.

Soit  $P$  une fonction de *pooling* et  $m$  le nombre de valeurs retournées au niveau de chaque fenêtre de *pooling*  $\mathbf{c}_{i:i+h_p-1}$ . La sortie  $\hat{\mathbf{c}}$  de la couche de *pooling* est exprimée comme suit :

$$\hat{\mathbf{c}} = P_m \{ \mathbf{c}_{i:i+h_p-1} \} \quad (3.11)$$

La sortie produite est un nouveau vecteur de taille  $T_p$  contenant les informations les plus importantes de l'entrée (carte de caractéristiques). La valeur de  $T_p$  est calculée comme suit :

$$T_p = \frac{T_o - h_p}{s_p} + 1 \quad (3.12)$$

où  $T_o$  la taille de la carte de caractéristiques à traiter,  $h_p$  la hauteur de la fenêtre de *pooling* et  $s_p$  le nombre de pas de déplacement.

Nous trouvons aussi les variantes de ces deux approches de *pooling* appelées *Max-pooling Global* et *Avg-pooling Global*. Celles-ci s'appliquent sur la totalité de la carte de caractéristiques  $\mathbf{c}$  en retournant respectivement la valeur maximale et la moyenne des valeurs locales.

Nous présentons dans la figure 3.7 un exemple d'application de l'opération *Max-pooling*. L'opération de *Max-pooling* s'effectue par un principe de fenêtre glissante de hauteur  $h = 2$  et de  $s = 2$  pas de déplacement, et appliquée sur une carte de caractéristiques de taille  $T_o = 8$  paramètres dans une seule direction (de haut en

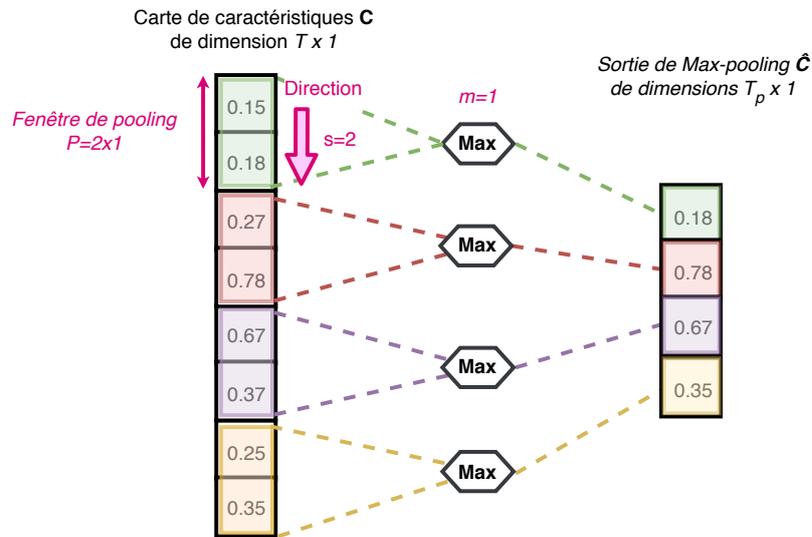


FIGURE 3.7 – Fonctionnement d’une opération de *Max-pooling* appliquée sur une carte de caractéristiques

bas). Nous récupérons la valeur maximale ( $m = 1$ ) locale de chaque fenêtre de *pooling* en produisant en sortie  $T_p = \frac{8-2}{2} + 1 = 4$  unités (en appliquant l’équation 3.12) qui représentent les caractéristiques les plus importantes de la carte  $\mathbf{c}$ .

### 3.4 Modélisation et prédiction

La modélisation est une opération de raisonnement de haut-niveau d’un réseau de neurones convolutif, car elle permet de déterminer le lien entre les représentations apprises et la sortie souhaitée du réseau.

Comme illustré dans la figure 3.8, cette opération est caractérisée par un ensemble de couches cachées entièrement connectées (FC - *Fully Connected Layer*) : une couche d’entrée qui correspond aux sorties de la couche de *pooling*  $\hat{\mathbf{c}}$  concaténées, une suite de couches cachées pour la modélisation, ainsi qu’une couche de sortie qui nous permet de prédire une unité spécifique en fonction de la tâche.

Une couche FC est un ensemble de neurones (voir la section 3.2) qui n’ont pas de connexion entre eux, mais chacun des neurones est relié à tous ceux de la couche précédente et suivante. La sortie de chaque neurone peut être considérée par la suite comme une entrée pour la couche suivante en appliquant l’équation 3.2.

## CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

---

La prédiction est la dernière opération d'un réseau de neurones convolutif qui permet de prédire une classe ou une valeur continue en fonction des caractéristiques apprises selon la tâche visée (classification ou régression).

La dernière couche du réseau est une couche totalement connectée de taille fixe. La sortie est obtenue en appliquant une fonction spécifique selon la tâche souhaitée :

- **Tâche de régression** : la dernière couche du réseau est une couche entièrement connectée ayant un seul neurone. La fonction d'activation Identité est généralement appliquée afin de renvoyer en sortie une valeur continue  $\hat{y} \in [-\infty, +\infty]$  ;
- **Tâche de classification** : la taille de la dernière couche est égale au nombre de classes. Généralement, deux types de fonctions peuvent être appliqués. Pour une tâche de classification binaire, la fonction *sigmoïde* est utilisée afin d'attribuer une valeur entre 0 et 1 pour chaque classe. Si la valeur obtenue est proche de 0 (moins de 0,5), alors, la classe prédite  $\hat{y} = 1$ , sinon,  $\hat{y} = 2$ . En classification multi-classes, la fonction *Softmax* est souvent utilisée pour convertir la sortie du réseau en une distribution de probabilités (ayant une somme égale à 1).

La fonction *Softmax* est définie comme suit :

$$\text{Softmax}(\mathbf{Z})_j = \frac{e^{Z_j}}{\sum_{k=1}^K e^{Z_k}} ; j \in \{1, \dots, K\} \quad (3.13)$$

avec  $Z = W \cdot X + b$ ,  $j$  l'index d'une classe et  $K$  le nombre de classes.

Pour obtenir la classe prédite  $\hat{y}$ , il suffit d'appliquer la fonction *argmax* sur la sortie de la fonction *Softmax* comme décrit dans l'équation 3.14.

$$\hat{y} = \text{argmax}(\text{Softmax}(\mathbf{Z})) \quad (3.14)$$

### 3.5 Apprentissage d'un réseau de neurones

Le processus d'apprentissage d'un réseau de neurones consiste à ajuster, optimiser et adapter les paramètres  $\theta$  (les poids  $W$  et les termes biais  $b$ ) de chaque couche en minimisant l'erreur de prédiction jusqu'à atteindre l'état désiré du réseau.

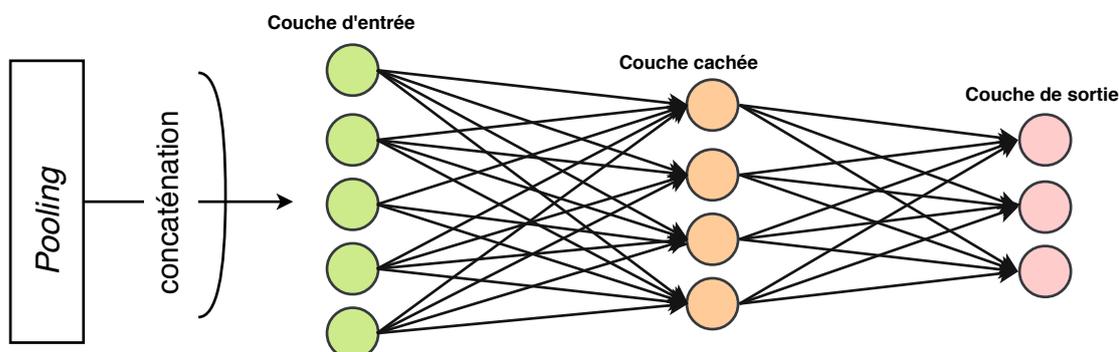


FIGURE 3.8 – Architecture du bloc de modélisation et prédiction

L'erreur de prédiction est calculée entre les vraies valeurs  $y$  (références) et les valeurs prédites  $\hat{y}$  (hypothèses) à l'aide d'une fonction de coût spécifique à la tâche visée (par exemple : entropie croisée pour une tâche de classification et MAE ou MSE pour une tâche de régression). La fonction de coût  $C$  (appelée aussi fonction de perte ou fonction objective) est un élément fondamental pour l'apprentissage supervisé. Elle est capable de déterminer le coût d'adaptation des paramètres  $\theta$  du réseau pour atteindre la prédiction souhaitée.

Les réseaux de neurones sont souvent appris à l'aide d'un algorithme de *descente du gradient*. Trois types de variantes de gradient ont été proposés dans la littérature :

- **Descente de gradient classique** : Le gradient est au niveau de l'ensemble du corpus d'apprentissage de  $N$  instances.

Le gradient est calculé au niveau du corpus d'apprentissage (de  $N$  instances) complet qui rend le processus d'apprentissage coûteux en termes de ressources et de temps. Les paramètres  $\theta$  sont mis à jour comme suit :

$$\theta_{t+1} = \theta_t - \eta \frac{1}{N} \sum_{i=1}^N \frac{\partial C(\hat{y}^i, y^i)}{\partial \theta} \quad (3.15)$$

où  $\eta$  le taux d'apprentissage (*learning rate*) ;

- **Descente de gradient stochastique (SGD)** : Le gradient est calculé sur chaque instance du corpus d'entraînement. Cela rend le processus d'apprentissage coûteux en termes de temps et de ressources. De plus, si le corpus

### CHAPITRE 3. LES RÉSEAUX DE NEURONES CONVOLUTIFS EN TRAITEMENT AUTOMATIQUE DES LANGUES

---

d'entraînement est bruité, le SDG peut être instable et renvoie ainsi de mauvaises prédictions. Les nouveaux paramètres  $\theta$  sont obtenus comme suit :

$$\theta_{t+1} = \theta_t - \eta \frac{\partial C(\hat{y}^i, y^i)}{\partial \theta} \quad (3.16)$$

- **Descente de gradient mini-batch** : c'est une variante de l'algorithme SGD. Le gradient est calculé sur un ensemble de  $m$  instances (mini-batch) du corpus d'entraînement au lieu d'une seule instance. C'est l'algorithme le plus utilisé aujourd'hui, il est beaucoup plus efficace et rapide par rapport aux autres algorithmes. La mise à jour des paramètres s'effectue comme suit :

$$\theta^{i+1} = \theta^i - \eta \frac{1}{m} \sum_{i=1}^m \frac{\partial C(\hat{y}^i, y^i)}{\partial \theta} \quad (3.17)$$

Une fois que la fonction de coût et la méthode de descente de gradient sont définies, le processus d'apprentissage est ensuite effectué à l'aide de l'algorithme *rétro-propagation du gradient*. C'est un algorithme récursif de calcul du gradient, où l'apprentissage s'effectue en deux étapes :

1. Une *propagation avant* qui consiste à parcourir le réseau de la première couche vers la dernière couche  $L$  en calculant les sorties d'activation  $f_\theta^l$  de chaque couche  $l \in [1, \dots, L]$ . Sachant que  $f_\theta^l = W^l \cdot f_\theta^{l-1} + b^l$  lorsque  $l > 1$  ;
2. Une *propagation arrière* : la propagation arrière est effectuée de la dernière couche vers la première couche après avoir calculé la fonction de coût  $C(\hat{y}, y)$ . Ainsi, la première étape de propagation consiste à calculer le gradient  $\frac{\partial C}{\partial f_\theta^L}$  de la fonction de coût par rapport à la sortie de la dernière couche  $L$  du réseau. Ensuite, le gradient  $\frac{\partial C}{\partial \theta^l}$  de la fonction  $C$  par rapport aux paramètres  $\theta^l$  est calculé pour chaque couche  $l$  comme suit :

$$\frac{\partial C}{\partial \theta^l} = \frac{\partial f_\theta^l}{\partial \theta^l} \frac{\partial C}{\partial f_\theta^l} \quad (3.18)$$

Les équations de dérivées de gradient de chaque couche d'un réseau de neurones convolutif sont détaillées dans [Collobert et al., 2011].

Nous trouvons également d'autres types d'algorithmes d'apprentissage tels que : Adam [Kingma and Ba, 2014], AdaGrad [Duchi et al., 2011], AdaDelta [Zeiler,

2012], Momentum [Sutskever et al., 2013]. Garantir une bonne qualité de prédiction et une convergence rapide du gradient vers l'erreur minimale représentent de vrais défis pour apprendre efficacement un réseau de neurones. Pour cela, plusieurs techniques de régularisation ont été implémentées comme : *Weight Decay* [Collobert and Bengio, 2004], *Dropout* [Srivastava et al., 2014], *Batch normalization* [Ioffe and Szegedy, 2015].

### 3.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux réseaux de neurones convolutifs que nous allons utiliser pour proposer une nouvelle approche de prédiction de performances des systèmes de reconnaissance automatique de la parole. Nous avons présenté tout d'abord le perceptron formel simple. Nous avons décrit ensuite le fonctionnement de base d'un réseau de neurones convolutif simple en détaillant les différents blocs de construction : l'entrée du réseau, l'opération de convolution, l'opération de *pooling* ainsi que la phase de modélisation et de prédiction permettant d'adapter la sortie du réseau en fonction de la tâche visée. Nous avons présenté enfin le processus d'apprentissage d'un réseau de neurones.



Deuxième partie

Contributions



## CHAPITRE 4

# CADRE EXPÉRIMENTAL

### Sommaire

---

<b>4.1</b>	<b>Scénario envisagé</b>	<b>54</b>
<b>4.2</b>	<b>Corpus</b>	<b>56</b>
<b>4.3</b>	<b>Métriques d'évaluation</b>	<b>57</b>
<b>4.4</b>	<b>Systèmes de reconnaissance de la parole construits</b>	<b>58</b>
4.4.1	Processus de pré-traitement	58
4.4.2	Modèles acoustiques	59
4.4.3	Modèles de langage	60
4.4.4	Dictionnaire de prononciation	62
4.4.5	Évaluation des systèmes	64
<b>4.5</b>	<b>Conclusion</b>	<b>65</b>

---

Dans ce chapitre, nous proposons un protocole expérimental dédié à la tâche de prédiction de performances. D’abord, nous présentons le scénario envisagé pour créer et évaluer des systèmes de prédiction. Puis nous présentons un corpus en français large et hétérogène (multiples programmes TV ou radio, mélange de parole non spontanée et spontanée, différents accents) dédié à la tâche. Enfin, nous proposons un système de reconnaissance automatique de la parole construit spécifiquement pour implémenter notre protocole et obtenir des transcriptions automatiques ainsi que leurs performances.

Ce chapitre est organisé comme suit. Nous décrivons dans un premier temps dans la section 4.1 le scénario envisagé pour la tâche de prédiction de performances. Nous présentons par la suite dans la section 4.2, un ensemble de corpus hétérogènes spécifiques pour créer des systèmes de reconnaissance automatique de la parole et des systèmes de prédiction de performances. Enfin, dans la section 4.4, nous présentons nos systèmes de RAP en reportant les performances sur des corpus dédiés pour la tâche de prédiction des performances.

## 4.1 Scénario envisagé

Dans notre protocole, nous nous intéressons à la prédiction de performances des systèmes de reconnaissance automatique de la parole sur des émissions qui n’ont jamais été vues durant l’apprentissage. Nous envisageons un scénario de prédiction difficile où les informations du système de RAP sont indisponibles (pas d’accès aux modèles, pas de treillis, ni N-meilleures hypothèses, etc.), où les données d’apprentissage/d’évaluation sont hétérogènes. Nos travaux se basent sur un système de prédiction de performances n’utilisant que les transcriptions automatiques (fournies par un SRAP) et/ou le signal audio afin de prédire la performance de la transcription correspondante. Les transcriptions de référence (humaines) ne sont disponibles que pour évaluer le système de transcription de la parole et pour produire un rapport de performances. Comme décrit dans la figure 4.1, un corpus nommé  $\text{Train}_{pred}$  est utilisé pour construire nos systèmes de prédiction. Il est constitué de triplets {signaux, transcription automatique, performance} (pour 75k tours de parole). Le corpus  $\text{Test}_{pred}$  est utilisé pour évaluer le système de prédiction, il contient aussi les triplets {signaux, transcriptions automatiques, performances}

(6.8k tours de parole) mais la performance du système est inconnue au moment de la prédiction et dévoilée uniquement au moment de l'évaluation de la qualité de prédiction.

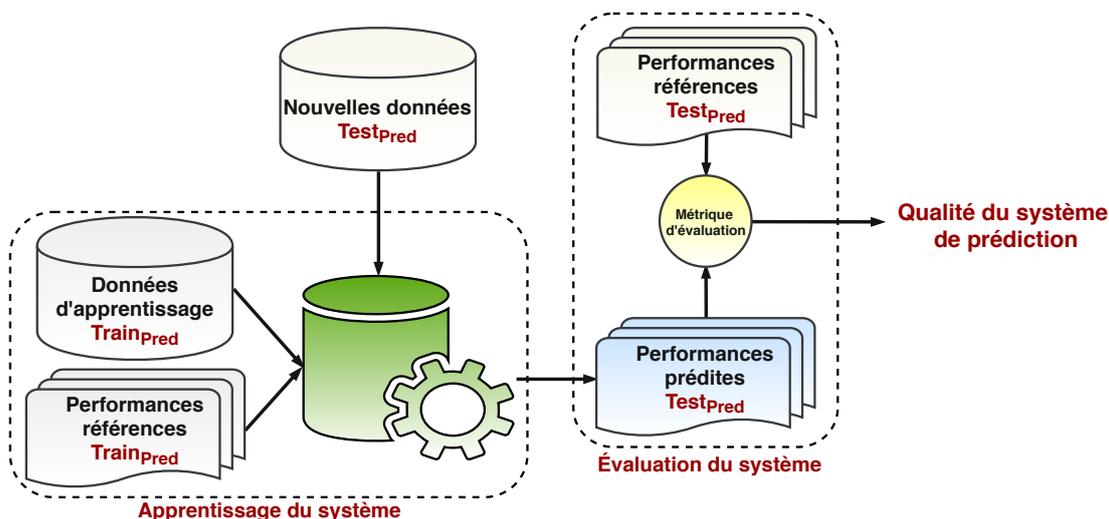


FIGURE 4.1 – Protocole d'évaluation pour la tâche de prédiction de performance

Pour créer des systèmes de prédiction de performances, nous proposons deux méthodes :

- Une approche à l'état de l'art (en utilisant un outil existant) basée sur des « traits techniques » et un algorithme de machine Learning (voir figure 4.2). La qualité du système prédictif dépend alors de la pertinence des caractéristiques sélectionnées, de la manière dont elles sont présentées au système ainsi que du choix de l'algorithme d'apprentissage.
- Une nouvelle approche *end-to-end* à l'aide des réseaux de neurones convolutifs (voir figure 4.3).

Afin de valider notre approche *end-to-end*, nous la comparons à l'approche état de l'art.

Afin d'implémenter ce protocole, nous avons donc besoin d'un système de reconnaissance de la parole pour produire les transcriptions automatiques ainsi que la performance associée pour l'intégralité des corpus  $\text{Train}_{pred}$  et  $\text{Test}_{pred}$ , ce qui nous permettra d'entraîner et d'évaluer des systèmes de prédiction de performances.

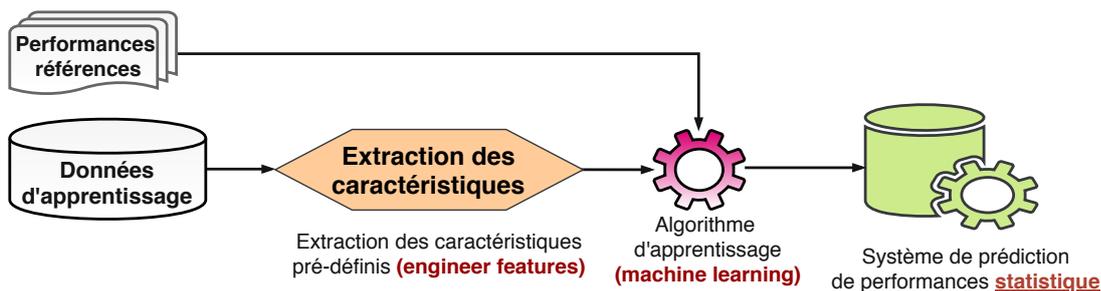


FIGURE 4.2 – Apprentissage à l'état de l'art d'un système de prédiction de performances

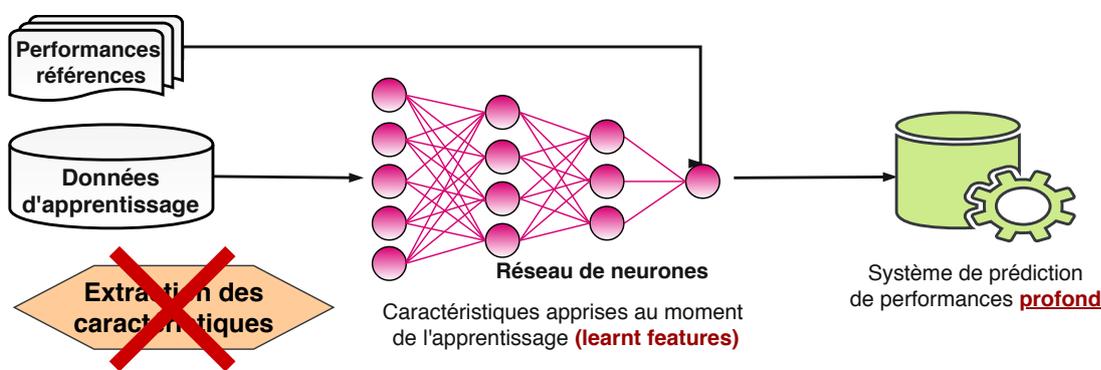


FIGURE 4.3 – Apprentissage de bout-en-bout d'un système de prédiction profond en utilisant les réseaux de neurones

## 4.2 Corpus

Les données utilisées dans notre protocole proviennent de différentes collections d'émissions en français :

- Un sous-ensemble du corpus Quaero<sup>1</sup> qui contient 41 heures de discours radiodiffusés de différents programmes de radio et de télévision français sur divers sujets.
- Les données du projet ETAPE [Gravier et al., 2012] qui comportent 37 heures d'émissions de radio et de télévision (principalement des discours spontanés avec des locuteurs qui se chevauchent).
- Des données des campagnes d'évaluation ESTER 1 & ESTER 2 [Galliano et al., 2005] qui contiennent 111 heures d'enregistrement audio transcrit. Ce

1. <http://www.quaero.org>

sont principalement des programmes de radio français et africains (mélange de discours préparés et plus spontanés : parole du présentateur, interviews, reportages).

- Les données de la campagne d’évaluation REPERE [Kahn et al., 2012] : 54 heures d’émissions transcrites de parole spontanée (des débats TV) et de la parole préparée (journaux télévisés).

Comme décrit dans le tableau 4.1, nos données contiennent de la parole non spontanée (NS) et de la parole spontanée (S). Les données d’entraînement ( $\text{Train}_{SRAP}$ ) de notre système de transcription de la parole automatique sont sélectionnées à partir des données non spontanées qui correspondent essentiellement à des journaux télévisés. Les données utilisées pour la tâche de prédiction ( $\text{Train}_{Pred}$  et  $\text{Test}_{Pred}$ ) sont un mélange des deux styles de parole (S et NS). Il est important de mentionner que les émissions du corpus  $\text{Test}_{Pred}$  n’existent pas dans le  $\text{Train}_{Pred}$  et vice versa. En outre, des émissions plus difficiles (ayant des taux d’erreur de mots plus élevés) ont été sélectionnées pour  $\text{Test}_{Pred}$ .

	$\text{Train}_{SRAP}$	$\text{Train}_{Pred}$	$\text{Test}_{Pred}$
<b>NS</b>	100h51	30h27	04h17
<b>S</b>	-	59h25	04h42
Durée	100h51	89h52	08h59

TABLE 4.1 – Distribution de nos corpus entre les styles de parole non spontanés (NS) et spontanés (S)

### 4.3 Métriques d’évaluation

Nous avons utilisé la boîte à outils LNE-Tools [Galibert, 2013] afin d’évaluer la qualité de notre système de transcription et pour produire les rapports de performance en termes de taux d’erreur de mots. La parole superposée et les tours de parole vides sont supprimés.

Afin d’évaluer la tâche de prédiction de performance, nous utilisons la métrique *Mean Absolute Error* (MAE) définie comme suit :

$$MAE = \frac{\sum_{i=1}^N |WER_{Ref}^i - WER_{Pred}^i|}{N} \quad (4.1)$$

Avec  $N$  le nombre d'unités (tours de parole, type d'émission, une instance d'un type d'émission ou document complet selon la granularité de prédiction choisie).

Nous utilisons également le coefficient de corrélation de rang Kendall  $\tau$  entre le score de référence et la sortie du système de prédiction au niveau des tours de parole.

## 4.4 Systèmes de reconnaissance de la parole construits

Afin d'obtenir les transcriptions automatiques des corpus  $\text{Test}_{Pred}$  et  $\text{Train}_{Pred}$  pour nos systèmes de prédiction de performances, nous avons construit un système de reconnaissance automatique de la parole hybride HMM-DNN basé sur la boîte à outils KALDI [Povey et al., 2011b], en suivant la "recette" standard. Ce système a été appris sur le corpus  $\text{Train}_{SRAP}$  (100 heures de journaux issus de ESTER, REPERE, ETAPE et Quaero). Pour améliorer la tâche d'apprentissage, les données textuelles ont été normalisées et pré-traitées de la même façon afin d'unifier les formes et réduire le vocabulaire. Ces données ont été exploitées pour entraîner des modèles de langage et des modèles acoustiques pertinents.

### 4.4.1 Processus de pré-traitement

L'opération de pré-traitement des données textuelles est une étape coûteuse en termes de temps et très importante pour obtenir des données exploitables durant l'apprentissage des modèles. Cette opération facilite et améliore la tâche d'entraînement des différents composants du système de reconnaissance automatique de la parole. Nous avons donc effectué une étude sur nos données afin de minimiser la taille du vocabulaire et avoir des formes homogènes et uniformes. Ce pré-traitement consiste à : enlever la casse, convertir les symboles existant dans les liens hypertextes et les adresses mails en lettres, convertir les chiffres romains en lettres, normaliser et convertir les chiffres en lettres, convertir les unités de mesures, convertir les symboles en lettres, transformer les abréviations en mots, segmenter en unités lexicales (tokenisation) et supprimer les ponctuations.

Nous avons aussi sélectionné les 762 2-grammes et 177 3-grammes les plus fréquents (nombre d'occurrences supérieur à 100) de notre corpus  $\text{Train}_{SRAP}$  pour

les considérer comme une seule unité lexicale. De plus, nous avons récupéré la liste des mots composés dans la ressource BDLEX pour les ajouter dans le vocabulaire de nos systèmes de RAP.

Le tableau 4.2 présente un exemple d’un tour de parole avant et après le processus de pré-traitement.

État	Transcription
Avant	proches de <b>23</b> à <b>26</b> degrés sur <b>la plupart</b> des villes de france
Après	proches de <b>vingt-trois</b> à <b>vingt-six</b> degrés sur <b>la-plupart</b> des villes de france

TABLE 4.2 – Exemple de tour de parole avant et après le processus de pré-traitement

#### 4.4.2 Modèles acoustiques

Un modèle acoustique HMM-DNN a été construit à l’aide des scripts fournis par la boîte à outils KALDI [Povey et al., 2011b]. Ce modèle a été appris sur le corpus  $\text{Train}_{SRAP}$  qui contient 100 heures de journaux issus de ESTER, REPERE, ETAPE et Quaero. Comme paramètres acoustiques, nous avons utilisé des MFCCs de dimension 13, leurs dérivées premières  $\Delta$ , leurs dérivées secondes  $\Delta\Delta$  et l’énergie. Une vecteur acoustique de dimension 40 a été obtenue pour chaque trame de parole.

L’apprentissage d’un modèle HMM-DNN avec la boîte à outils Kaldi nécessite tout d’abord un modèle acoustique de type HMM-GMM permettant de générer les alignements au niveau des données acoustiques. Comme décrit dans le tableau 4.3, nous avons commencé le processus d’apprentissage par l’entraînement d’un modèle acoustique mono-phone (nommé mono) qui a été utilisé pour effectuer un alignement forcé entre les signaux et les états HMMs en exploitant les transcriptions références de notre corpus  $\text{Train}_{SRAP}$ . Ensuite, un modèle tri-phone (nommé tri-phone\_2a) a été appris respectivement sur les caractéristiques MFCC,  $\Delta$  et  $\Delta\Delta$ . Une analyse discriminante linéaire (*LDA*) et une transformation linéaire à vraisemblance maximale (*MLLT*) ont été appliquées sur une fenêtre de trames de largeur 7 (3 contextes gauches et 3 droits) et projetées dans un espace de 40 dimensions pour apprendre un modèle acoustique triphone conventionnel nommé

Tri-phone\_2b (40k Gaussiennes). Ensuite, un modèle dépendant du locuteur (150k Gaussiennes) a été appris en appliquant une régression linéaire à maximum de vraisemblance fMLLR aux paramètres acoustiques. Également, un modèle SGMM a été appris (50k gaussiennes). Enfin, le dernier modèle obtenu est le modèle HMM-DNN (nommé *DNN*) qui exploite les réseaux de neurones de type DNN. Le réseau DNN utilisé est composé de 4 couches cachées de taille 1024 et une couche de sortie de 4782 unités en appliquant la fonction Softmax. Le modèle a été appris sur 15 époques avec un taux d'apprentissage qui varie entre 0,01 et 0,001.

Nom du modèle	Détails d'apprentissage	Script Kaldi
<b>Mono</b>	Mono-phone	mono
<b>Tri-phone_2a</b>	Tri-phone + $\Delta$ + $\Delta\Delta$	tri2a
<b>Tri-phone_2b</b>	LDA + MLLT	tri2b
<b>GMM</b>	LDA + MLLT + <u>fMLLR</u> + SAT	tri3b
<b>SGMM</b>	LDA + MLLT + <u>SGMM</u>	sgmm2_5b2
<b>DNN</b>	LDA + MLLT + <u>DNN</u>	nnet5c

TABLE 4.3 – Description des modèles acoustiques produits en termes de méthode d'apprentissage et script utilisé

### 4.4.3 Modèles de langage

Deux modèles de langage 3-grammes et 5-grammes ont été produits en utilisant l'outil SRIIM [Stolcke, 2002]. Ces modèles ont été appris sur une grande quantité de données monolingues en français disponibles sur Opus<sup>2</sup>. Le tableau 4.4 présente la liste des corpus utilisés pour l'apprentissage des modèles.

Afin d'exploiter ces données, nous avons commencé par une étape de normalisation des formes en appliquant la chaîne de pré-traitement proposée dans la section 4.4.1. Ensuite, nous avons divisé le corpus  $\text{Train}_{SRAP}$  en deux :  $\text{Train}_{LM}$  et  $\text{Dev}_{LM}$ . Le corpus  $\text{Train}_{LM}$  a été ajouté à la liste des corpus conçus pour l'apprentissage des modèles et le corpus  $\text{Dev}_{LM}$  a été utilisé comme un corpus de développement pour optimiser le modèle construit.

Afin de construire les deux modèles de langage n-grammes ( $n=3,5$ ), nous avons produit un modèle n-gramme pour chacun des corpus à utiliser. Ensuite, nous avons effectué une interpolation des modèles en utilisant le corpus  $\text{Dev}_{LM}$  afin

---

2. <http://opus.nlpl.eu/>

Corpus	#Phrases	#Tokens	Vocab
EUbookshop	18M	432M	1,71M
TED2013+wit3	0,16M	2M	0,06M
GlobalVoices	0,37M	7M	0,18M
giga	18M	57M	1,23M
europarl-v7	2,24M	60M	0,13M
MultiUN	13M	404M	0,41M
OpenSubtitles2016	90M	534M	0,87M
DGT	3.1M	61,7M	0,28M
News-Commentary11+news-wmt	30M	661M	1,43M
lemonde	13M	368M	1,12M
Trames	0,21M	0,79M	0,03M
wikipedia	20M	502M	2M
Total	208,08M	3089,49M	5,14M

TABLE 4.4 – Description des données monolingues utilisés pour construire les modèles de langage pour le système RAP-LIG

de minimiser la perplexité du modèle interpolé. Enfin, étant donné que le modèle obtenu est volumineux, nous avons effectué une opération de filtrage sur le modèle interpolé en ne gardant que les n-grammes ayant une probabilité supérieure à  $1e-9$ .

Le tableau 4.5 décrit les performances des deux modèles de langage produits en termes de perplexité ainsi que le nombre des mots hors vocabulaire (OOVs). Les performances obtenues montrent que le modèle 5-grammes obtient une meilleure perplexité que le modèle 3-grammes sur les 4 corpus. Étant donnée que  $\text{Train}_{LM}$  est déjà utilisé durant l'apprentissage des modèles, il obtient une meilleure perplexité sur les deux modèles : 68,7 pour le modèle 3-grammes et 50,0 pour le modèle 5-grammes.

Pour les corpus dédiés à la tâche de prédiction de performances, nous remarquons que les perplexités obtenues au niveau des corpus  $\text{Train}_{Pred}$  et  $\text{Test}_{Pred}$  sont élevées. Par exemple, pour les modèles 3-grammes et 5-grammes, le corpus  $\text{Train}_{Pred}$  obtient respectivement 153.00 et 127.21 de perplexité avec 2300 mots hors vocabulaire.

	3-grammes	5-grammes	%OOVs	#Types
<b>Dev<sub>ML</sub></b>	133,6	106,4	0	20,6k
<b>Train<sub>ML</sub></b>	68,7	50,0	0	37,8k
<b>Train<sub>Pred</sub></b>	153,0	127,2	6,47 %	36,4k
<b>Test<sub>Pred</sub></b>	156,8	130,9	5,18 %	11,8k

TABLE 4.5 – Performances des deux modèles de langage filtrés LM-3G et LM-5G évalués sur différents corpus en termes de perplexité

#### 4.4.4 Dictionnaire de prononciation

Notre dictionnaire de prononciation est construit à l’aide de la ressource lexicale *BDLEX* [De Calmès and Pérennou, 1998] et de l’outil de conversion automatique de graphèmes-à-phonèmes *LIA\_Phon*<sup>3</sup> afin de trouver les variantes de prononciation des mots. Il a été produit en 4 étapes :

1. **Conversion des phonèmes** : nous avons unifié en API<sup>4</sup> (Alphabet Phonétique International) les codes *BDLEX* (basés sur *SAMPA*<sup>5</sup>) et *LIA\_PHON*. 35 codes ont été utilisés dans notre dictionnaire : /a/, /e/, /i/, /o/, /u/, /ø/, /œ/, /y/, /ɛ/, /ɔ/, /ə/, /ẽ/, /œ̃/, /õ/, /ã/, /p/, /b/, /t/, /d/, /k/, /g/, /f/, /v/, /s/, /z/, /ʃ/, /ʒ/, /ɥ/, /j/, /w/, /m/, /n/, /ŋ/, /l/ et /ʁ/ ;
2. **Construire le vocabulaire** : nous avons sélectionné le vocabulaire de notre corpus *Train<sub>SRAP</sub>* (38k mots). Ensuite, nous avons créé un modèle de langage 1-gramme (appris comme dans la section 4.4.3) afin d’extraire les 80k mots les plus probables. Enfin, nous avons ajouté la liste des mots composés proposée dans la ressource *BDLEX* (de-la, afin-de, etc.). Le vocabulaire a été obtenu après avoir appliqué la chaîne de pré-traitement des données textuelles (voir la sous section 4.4.1). La taille de notre vocabulaire final est de 82000 mots.
3. **Phonétisation *BDLEX*** : attribuer à chaque mot de notre vocabulaire la liste des phonétisations possibles proposée par *BDLEX*.  
Étant donné que notre corpus est hétérogène, nous avons ajouté des variantes de prononciation détaillées dans le tableau 4.6 ;

3. [http://lia.univ-avignon.fr/chercheurs/bechet/download/lia\\_phon.v1.2.jul06.tar.gz](http://lia.univ-avignon.fr/chercheurs/bechet/download/lia_phon.v1.2.jul06.tar.gz)

4. [https://fr.wikipedia.org/wiki/Alphabet\\_phon%C3%A9tique\\_international](https://fr.wikipedia.org/wiki/Alphabet_phon%C3%A9tique_international)

5. [https://fr.wikipedia.org/wiki/Symboles\\_SAMPA\\_fran%C3%A7ais](https://fr.wikipedia.org/wiki/Symboles_SAMPA_fran%C3%A7ais)

Phonèmes	Variantes ajoutées
/ɛ/	/ɛ/ ou /e/
/ɔ/	/ɔ/ ou /o/
/œ̃/	/œ̃/ ou /ɛ̃/
/ə/	/ə/, /ø/, /œ/ ou //

TABLE 4.6 – Liste des variantes de phonèmes supplémentaires pour BDLEX

4. **Phonétisation automatique** : si un mot de notre corpus  $\text{Train}_{SRAP}$  n'existe pas dans  $BDLEX$ , nous utilisons l'outil  $LIA\_PHON$  pour obtenir automatiquement les prononciations correspondantes du mot.

Tandis que  $LIA\_PHON$  est un outil automatique et qui peut produire des erreurs, nous avons vérifié manuellement les prononciations des 1000 mots les plus fréquents. Par la suite nous avons effectué les corrections suivantes :

- Supprimer les phonèmes /ã t/ et /ã/ pour les mots qui se terminent par « *aient* » ;
- remplacer la suite de phonèmes /w a d/ par /o i d/ pour les mots qui se terminent par « *oide* » ;
- Générer des variantes au niveau de {/e/,/ɛ/} et {/ɔ/,/o/} s'ils se sont situées au milieu ;
- Pour les mots qui se terminent par « *ent* », nous avons effectué un traitement spécifique comme décrit dans le tableau 4.7.

Catégorie	Terminaison	Phonèmes supprimés
<b>Verbe</b>	ent	(/ə/) (/t/)
<b>Nom</b>	ent	/ã/
<b>Adverbe</b>	ment	/m ã (t)/

TABLE 4.7 – Les règles suivies pour corriger les phonétisations

Parmi les 82k mots de notre vocabulaire, 45k de mots ont été trouvés dans  $BDLEX$ , dont nous avons récupéré les différentes variantes de prononciation. Les autres mots (37k) ont été phonétisés automatiquement à l'aide de  $LIA\_PHON$ .

### 4.4.5 Évaluation des systèmes

En exploitant le modèle acoustique DNN et le modèle de langage 3-grammes présentés respectivement dans les sections 4.4.2 et 4.4.3, un système de reconnaissance automatique de la parole hybride HMM-DNN a été appris sur le corpus Train SRAP (100 heures de journaux issus des corpus ESTER, REPERE, ETAPE et Quaero). Le modèle 5-grammes (voir section 4.4.3) a été utilisé pour recalculer le taux d'erreur de mots (*rescoring*) et obtenir des transcriptions automatiques plus pertinentes. L'évaluation de nos systèmes de RAP a été effectuée en termes de WER en utilisant l'outil *asr-eval*. Nous rappelons que les tours de parole vides et la parole superposée ont été supprimés. De plus, une étape de post-traitement des mots composés est requise pour évaluer correctement les sorties des systèmes. Cette étape de post-traitement consiste à détokeniser les mots composés dans les transcriptions de référence ainsi que dans les transcriptions hypothèses (comme : afin-de → afin de).

Systemes	MA	ML	Train <sub>Pred</sub>	Test <sub>Pred</sub>
SRAP <sub>1</sub>	HMM-DNN	5-grammes	<b>22.29</b>	<b>31.20</b>
SRAP <sub>2</sub>	HMM-DNN	3-grammes	23.64	32.80

TABLE 4.8 – Description des 2 systèmes de reconnaissance automatique de la parole produits et leurs performances WER évalués sur nos corpus Train<sub>Pred</sub> et Test<sub>Pred</sub>

Dans le tableau 4.8, nous évaluons le système hybride HMM-DNN avant et après le *rescoring*. Les performances obtenues montrent que le système SRAP<sub>1</sub> produit la meilleure qualité de transcription, il obtient 22.29 % et 31.20 % de WER respectivement sur les corpus Train<sub>Pred</sub> et Test<sub>Pred</sub>. Les WER obtenus sont élevés à cause de l'hétérogénéité des deux corpus (qui contiennent de la parole spontanée/préparée, différents accents, des appels téléphoniques, etc.) ainsi que le type de données des corpus Train<sub>Pred</sub> et Test<sub>Pred</sub> qui est très différent par rapport aux données d'apprentissage du système de reconnaissance automatique de la parole (Train<sub>SRAP</sub>).

Pour valider les performances de notre système SRAP<sub>1</sub>, nous comparons dans la figure 4.4 les performances de notre système (colorées en rose) à celles obtenues au cours des différentes campagnes d'évaluation (colorées en noir) sur les mêmes

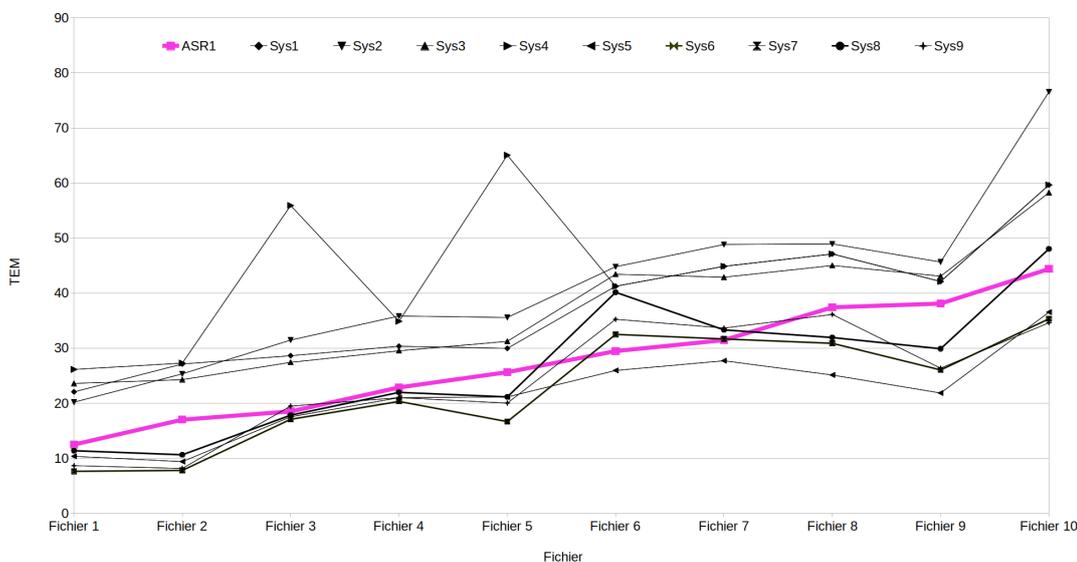


FIGURE 4.4 – Performance (WER) du système SRAP<sub>1</sub> comparé aux performances d’autres systèmes de la campagne d’évaluation REPERE sur des données identiques. Chaque fichier en abscisse représente une instance d’un type d’émission.

fichiers (10 au total). Le système que nous avons développé est situé au milieu des systèmes proposés. Cela signifie que notre système produit des transcriptions correctes et les performances sont corrélées avec celles des autres systèmes.

Dans les tableaux 4.9 et 4.10, nous présentons les performances en termes de WER de nos données d’apprentissage ( $Train_{Pred}$ ) et d’évaluation ( $Test_{Pred}$ ) des systèmes de prédiction par type d’émission transcrites par le système SRAP<sub>1</sub>. Nos émissions ayant un style de parole spontanée ont systématiquement un taux d’erreur de mots plus élevé (de 28,74 % à 45,15 % selon l’émission) par rapport aux émissions ayant un style de parole non spontanée (de 12,06 % à 25,41 % selon l’émission). Cette division S/NS nous permettra de comparer nos systèmes de prédiction de performances sur différents types de documents contenant du discours non spontanés et spontanés.

## 4.5 Conclusion

Dans ce chapitre, nous avons proposé un protocole expérimental spécifique pour la tâche de prédiction de performances. Notre cas d’utilisation repose sur un

Source	Type d'émission	TP	Mots	WER
<b>Non spontanées (NS)</b>				
<b>Quaero</b>	Arte News (AN)	108	3726	12.06
<b>ESTER 2</b>	Tvme (T)	915	10706	18.44
<b>Quaero</b>	France Culture TEMPS (FCT)	324	10091	20.92
<b>Quaero</b>	Fab Histoire (FH)	578	10022	22.76
<b>ESTER 2</b>	Africa1 (A1)	1184	15257	25.41
<b>Spontanées (S)</b>				
<b>Quaero</b>	Ce Soir Ou Jamais (CSOJ)	208	10992	28.74
<b>REPERE</b>	Planete Showbiz (PS)	884	15946	36.74
<b>REPERE</b>	Culture Et Vous (CV)	740	16026	39.79
<b>ETAPE</b>	La Place Du Village (PV)	1896	20396	45.15

 TABLE 4.9 – Performance sur le corpus  $\text{Test}_{Pred}$  en termes de WER (Taux d'Erreur Mots)

Source	Type d'émission	TP	Mots	WER
<b>Non spontanées (NS)</b>				
<b>ETAPE</b>	PileEtFace	5012	103320	17.94
<b>ESTERE</b>	RTM	26943	243741	18.84
<b>ESTERE</b>	RFI	28396	283830	21.65
<b>Spontanées (S)</b>				
<b>Quaero</b>	TELSONNE (TELS)	5210	269442	23.23
<b>Quaero</b>	France3 Débat (FD)	1005	43780	26.17
<b>Quaero</b>	France Inter Débat (FID)	8465	150916	29.26

 TABLE 4.10 – Performance sur le corpus  $\text{Train}_{Pred}$  en termes de WER (Taux d'Erreur Mots)

système de prédiction qui prend en entrée des transcriptions automatiques et/ou des signaux acoustiques pour prédire la performance. Le processus de prédiction de performances est décrit dans la figure 4.1. Nous avons proposé également un corpus large et hétérogène en français dédié à la tâche de prédiction de performances. Trois corpus ont été proposés :  $\text{Train}_{SRAP}$  pour créer des systèmes de reconnaissance automatique de la parole ainsi que  $\text{Train}_{Pred}$  et  $\text{Test}_{Pred}$  afin d'apprendre et évaluer les systèmes de prédiction de performances.

Dans notre protocole, nous envisageons un scénario de prédiction difficile :

i) les informations internes des systèmes sont indisponibles (pas de N-meilleures hypothèses ni de mesures de confiance). ii) les données dédiées à la prédiction sont non-homogènes et contiennent de la parole spontanée (difficile à transcrire → des WER élevés). iii)  $\text{Train}_{Pred}$  et  $\text{Test}_{Pred}$  contiennent des types d'émissions totalement différents.

Enfin, nous avons proposé un système de reconnaissance automatique de la parole hybride HMM-DNN français à l'état de l'art (nommé  $\text{SRAP}_1$ ) pour obtenir des transcriptions automatiques. Notre système obtient respectivement 22.29 % et 31.20 % de WER sur les données  $\text{Train}_{Pred}$  et  $\text{Test}_{Pred}$  et obtient des performances correctes comparé à d'autres systèmes de RAP du français.



## CHAPITRE 5

# IMPLÉMENTATION DES SYSTÈMES DE PRÉDICTION DE PERFORMANCES

### Sommaire

---

<b>5.1</b>	<b>Prédiction basée sur des traits explicites (<i>baseline</i>) . . .</b>	<b>70</b>
<b>5.2</b>	<b>Prédiction par les réseaux neuronaux convolutifs (CNNs)</b>	<b>72</b>
5.2.1	Architecture . . . . .	72
5.2.2	Expériences . . . . .	74
5.2.3	Résultats . . . . .	76
5.2.4	Analyse des taux d'erreur de mots prédits . . . . .	78
<b>5.3</b>	<b>Conclusion . . . . .</b>	<b>79</b>

---

Dans le chapitre précédent, nous avons présenté notre protocole expérimental pour la prédiction de performances. Nous envisageons de créer des systèmes de prédiction en utilisant des données textuelles (transcriptions automatiques) et/ou leurs signaux acoustiques. Pour cela, nous avons proposé deux corpus :  $\text{Train}_{Pred}$  pour apprendre et  $\text{Test}_{Pred}$  pour évaluer les systèmes de prédiction. Afin d’obtenir les transcriptions de ces deux corpus, nous avons créé un système de reconnaissance automatique de la parole hybride HMM-DNN (nommé SRAP<sub>1</sub>).

Dans ce chapitre, nous proposons un premier système neuronal pour la tâche de prédiction tout en comparant l’utilisation de plusieurs types d’entrées. Afin de valider la qualité de prédiction de notre approche, nous proposons une étude comparative entre deux approches de prédiction : l’une fondée sur des traits prédéfinis (système à l’état de l’art) et une nouvelle approche de prédiction fondée sur un système se basant sur des réseaux de neurones convolutifs.

## 5.1 Prédiction basée sur des traits explicites (*baseline*)

Afin d’avoir un système de prédiction état de l’art (extraction des traits prédéfinis), nous avons adapté l’outil TranscRater [Jalalvand et al., 2016] de l’anglais vers le français. Ce dernier s’appuie sur l’extraction de traits explicites (*engineered features*) pour prédire la performance de chaque entrée en termes de WER. En exploitant des résultats empiriques antérieurs dans [Negri et al., 2014, de Souza et al., 2013, Jalalvand et al., 2015b, de Souza et al., 2015], TranscRater exploite l’algorithme Extremely Randomized Trees [Geurts et al., 2006] pour l’apprentissage du système. La sélection des traits est effectuée avec l’algorithme Randomized Lasso [Meinshausen and Bühlmann, 2010]. Les principaux hyper-paramètres du modèle sont optimisés à l’aide d’une grille de recherche [Bergstra and Bengio, 2012] avec une validation croisée sur l’ensemble des données d’apprentissage, afin de minimiser l’erreur absolue moyenne (MAE) entre les vrais WER et les WER prédits.

TranscRater est capable d’extraire 63 traits de quatre types :

- **9 traits morphosyntaxiques (POS)** : permettent de capturer la plausibilité de la transcription d’un point de vue syntaxique en utilisant l’outil

Treetagger [Schmid, 1995]. Pour chaque mot compris dans un tour de parole transcrit, un score de prédiction d'étiquette POS est attribué au niveau mot lui-même ainsi qu'au précédent et au suivant. Cette fenêtre glissante de 3 mots est utilisée pour calculer la valeur moyenne de l'ensemble du tour de parole transcrit. De plus, le vecteur de traits comporte également le nombre et le pourcentage de classes de tokens (nombres, noms, verbes, adjectifs et adverbes). Ces traits ont été testés dans diverses conditions (données propres/bruitées, microphones simples/multiples) [Jalalvand et al., 2015a,c];

- **3 traits issus du modèle de langue (LM)** : permettent de capturer la plausibilité de la transcription selon un modèle n-gramme. Ils comprennent la moyenne des probabilités des mots, la somme des log-probabilités et le score de perplexité pour chaque transcription. Un modèle 5-grammes est entraîné en utilisant l'outil SRILM [Stolcke et al., 2002] sur l'ensemble des corpus textes de 3 milliards de mots mentionné dans le tableau 4.4;
- **7 traits lexicaux (LEX)** : les traits sont extraits à partir du lexique de notre système de transcription : un vecteur de traits contenant la fréquence des catégories de phonèmes liées à la prononciation de chaque mot. Comme décrit dans le tableau 5.1, l'outil transcRater exploite 6 catégories de phonèmes;

Catégorie	Phonèmes
<b>Voyelle</b>	/a/ /e/ /i/ /o/ /u/ /ø/ /œ/ /y/ /ɛ/ /ɔ/ /ə/
<b>Voyelle nasale</b>	/ẽ/ /œ̃/ /õ/ /ã/
<b>Plosive</b>	/p/ /b/ /t/ /d/ /k/ /g/
<b>Fricative</b>	/f/ /v/ /s/ /z/ /ʃ/ /ʒ/
<b>Approximante</b>	/ɥ/ /j/ /w/
<b>Nasale et liquide</b>	/m/ /n/ /ŋ/ /l/ /ʁ/

TABLE 5.1 – Liste des phonèmes de notre dictionnaire de phonétisation avec leurs types

- **44 traits acoustiques (SIG)** : ils capturent des informations sur le signal d'entrée (conditions générales d'enregistrement, accents spécifiques au locuteur). Pour l'extraction des traits, TranscRater calcule 13 paramètres de type MFCC (en utilisant openSMILE Eyben et al. [2010]), leurs dérivées, accélération et log-énergie, fréquence fondamentale (F0), probabilité de voisement, contours d'intensité et le pitch pour chaque trame de parole. Pour

l'ensemble du signal d'entrée, le vecteur de traits SIG est obtenu en calculant la moyenne des valeurs de chaque trame.

## 5.2 Prédiction par les réseaux neuronaux convolutifs (CNNs)

Afin de prédire le TEM, nous proposons une nouvelle approche de régression supervisée fondée sur des réseaux de neurones convolutifs. Notre réseau prend en entrée des données textuelles et/ou des données acoustiques (signal brut, des MFCC ou des spectrogrammes). Suivant notre protocole expérimental, le système de reconnaissance automatique de la parole est considéré comme une boîte noire, et seuls les signaux et/ou les transcriptions automatiques sont fournis pour créer et évaluer des systèmes de prédiction. Nous avons construit notre modèle en utilisant à la fois Keras [Chollet et al., 2015] et Tensorflow<sup>1</sup> [Abadi et al., 2015].

### 5.2.1 Architecture

**Pour l'entrée textuelle**, nous proposons une architecture inspirée de Kim [2014] (en vert dans Figure 5.1). La première couche du réseau est une couche d'*embeddings* (voir la section 3.3.1.1) apprenable permettant de convertir une séquence de mots en une représentation matricielle. L'entrée est un tour de parole complété à  $n$  mots ( $n$  est défini comme la longueur de la plus longue phrase dans notre corpus complet) présentée sous forme d'une matrice (nommée *EMBED*) de taille  $n \times k$ , où  $k$  la taille des embeddings de mots.

Des opérations de convolution (voir la section 3.3.2) parallèles sont appliquées sur la matrice *EMBED* avec des fenêtres de convolution de différentes hauteurs  $h \in [1, 3, 5, 7, 9]$ . Chaque fenêtres de convolution implique 256 filtres de dimension  $h \times k$  qui sont appliqués sur des segments de  $h$  mots afin de produire en sortie un ensemble de cartes de caractéristiques (*feature map*). Ensuite, une opération de *Max-pooling* de taille  $4 \times 1$  suivie par une opération d'agrégation (*Global Average Pooling*) (voir la section 3.3.3) sont appliquées sur chaque carte de caractéristiques produites. Les opérations de convolution et de *pooling* fournissent une entrée de taille fixe (nombre de filtres  $\times$  nombre de fenêtres de convolution) aux deux couches

---

1. <https://www.tensorflow.org>

cachées entièrement connectées (256 et 128 unités) suivies respectivement d'une régularisation de type *dropout* (0,6 et 0,2) avant la prédiction de la performance (TEM).

**Pour l'entrée du signal**, nous proposons une architecture inspirée de la meilleure architecture (*m18*) proposée dans [Dai et al., 2017] (colorée en rouge dans Figure 5.1). Il s'agit d'un CNN profond avec 17 couches convolution+max-pooling suivies d'une opération d'agrégation (*Global Average Pooling*) et de trois couches cachées complètement connectées (512, 256 et 128 unités). Nous avons ajouté une régularisation (Dropout) de 0,2 entre les deux dernières couches (256 et 128). Nous proposons plusieurs méthodes pour encoder le signal avec le CNN en utilisant Librosa [McFee et al., 2015] : les échantillons du signal brut (RAW-SIG), le spectrogramme (MEL-SPEC) ou des coefficients MFCCs.

Afin de prédire un taux d'erreur (TEM) à l'aide des réseaux CNN, nous proposons deux approches différentes :

- **CNN<sub>Softmax</sub>** : nous utilisons les probabilités Softmax et un vecteur fixe externe nommé  $TEM_{Vector}$  pour calculer le WER prédit ( $TEM_{Pred}$ ).  $TEM_{Vector}$  et les probabilités Softmax doivent avoir la même dimension.  $TEM_{Pred}$  est alors défini comme suivant :

$$TEM_{Pred} = \sum_{C=1}^{NC} P_{Softmax}(C) * TEM_{Vector}(C) \quad (5.1)$$

Avec **NC est la dimension**  $NC$  du vecteur  $TEM_{Vector}$ . Dans nos expériences,  $NC$  est égale à 6 et  $TEM_{Vector}=[0 \%, 25 \%, 50 \%, 75 \%, 100 \%, 150 \%]$

- **CNN<sub>ReLU</sub>** : nous appliquons la fonction ReLU (la taille de sortie est égale à 1 à la dernière couche cachée du réseau). Cette fonction permettra d'estimer directement le WER en retournant une valeur de type réel entre 0 et  $+\infty$ .

**Pour l'utilisation jointe des données textuelles et acoustiques**, nous fusionnons les deux dernières couches cachées de CNN EMBED et CNN RAW-SIG (ou MEL-SPEC ou MFCC) en les concaténant et en les faisant passer à une nouvelle couche cachée (de taille 128) avant la prédiction de WER avec le CNN<sub>Softmax</sub> ou le CNN<sub>ReLU</sub> (représentées par des lignes pointillées dans la Figure 5.1). Nous entraînons par la suite le réseau de la même manière.

Contrairement aux traits de l’approche de base (extraction qui nécessite d’avoir défini les traits au préalable, on parle dans ce cas d’*engineered features*), les traits CNN textuels sont extraits et entraînés à partir des représentations vectorielles des mots (on parle alors de *learnt features*). Ces traits sont appris par le réseau neuronal jusqu’à ce que le comportement désiré soit obtenu.

### 5.2.2 Expériences

Dans cette section, nous comparons les deux approches de prédiction de performances des SRAP : prédiction fondée sur des caractéristiques explicites et prédiction fondée sur des caractéristiques entraînées en utilisant les CNNs. La prédiction par l’outil TranscRater s’appuie sur les traits issus de la sortie du SRAP et du signal (POS, LEX, LM et SIG), tandis que le CNN est basé sur la sortie du SRAP et le signal brut. Pour le CNN, nous sélectionnons aléatoirement 10 % des données  $\text{Train}_{Pred}$  comme corpus de développement DEV. Le reste est considéré comme un corpus d’apprentissage du réseau (TRAIN). Dix modèles de prédiction sont entraînés selon 10 sélections aléatoires de la partition TRAIN/DEV. Nous utilisons 50 époques d’apprentissage.

L’entraînement est effectué à l’aide de l’algorithme *Adadelta* [Zeiler, 2012] sur des mini-batches de taille 32. La métrique *MAE* est utilisée à la fois comme fonction de perte (coût) et comme mesure d’évaluation. Après la phase d’apprentissage, nous prenons le meilleur modèle (parmi les 10 sélections aléatoires de la partition TRAIN/DEV) obtenu en termes de MAE sur le corpus DEV et nous l’évaluons sur le corpus  $\text{Test}_{Pred}$ .

Nous étudions plusieurs entrées pour le CNN :

- Entrées textuelles (transcription automatique) uniquement (**EMBED**) : l’entrée du réseau est une matrice d’embeddings de mots de dimension 296x100, où 296 la longueur de l’hypothèse SRAP la plus longue dans notre corpus et 100 la dimension des embeddings. Les paramètres de la couche d’embeddings sont initialisés à l’aide d’un modèle pré-entraîné sur l’ensemble de corpus textes de 3 milliards de mots (mentionné dans la section 4.4.3) en exploitant l’outil Word2Vec [Mikolov et al., 2013a]. Ces paramètres seront par la suite mis à jour au moment de l’apprentissage du réseau ;
- Signal brut uniquement (**RAW-SIG**) : les modèles sont entraînés sur des

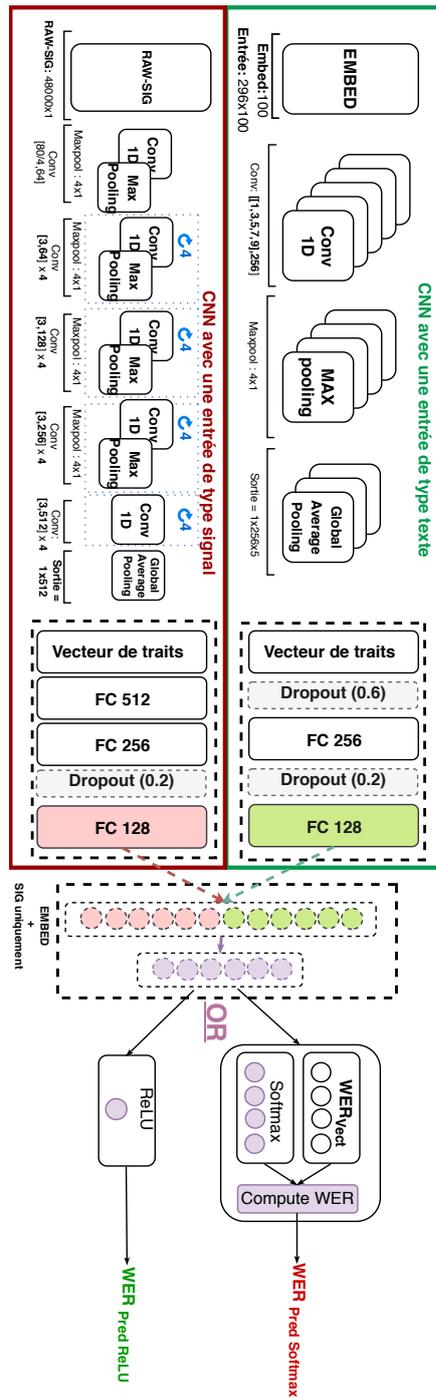


FIGURE 5.1 – Architecture de nos CNN à partir d’entrées texte (vert) et signal (rouge). Les couches avec des pointillés correspondent à l’utilisation conjointe texte+signal

tours de parole de 6 secondes et échantillonnés à 8kHz seulement (pour éviter les problèmes de surcharge de mémoire au cours de l'apprentissage du CNN). Les tours de parole courts ( $< 6s$ ) sont complétés par des zéros (silence). Notre entrée est un vecteur de dimension  $48000 \times 1$ . Les paramètres des filtres sont détaillés dans la Figure 5.1 ;

- Spectrogramme seulement (**MEL-SPEC**) : nous utilisons la même configuration que pour le signal brut ; nous avons des vecteurs en entrée de dimension 96 (chaque dimension correspond à une plage de fréquence particulière) extraits toutes les 10 ms (la fenêtre d'analyse est de 25ms). Notre entrée a donc une dimension  $601 \times 96$ ,<sup>2</sup> ;
- Paramètres **MFCC** seulement : nous calculons 13 MFCC toutes les 10 ms pour fournir au réseau CNN une entrée de dimension  $601 \times 13$  ;
- Entrées conjointes (texte et signal) (**EMBED+RAW-SIG** ou **EMBED+MEL-SPEC** ou **EMBED+MFCC**) : dans ce cas, nous concaténons les dernières couches cachées des réseaux CNN texte et signal (lignes pointillées dans la figure 5.1).

### 5.2.3 Résultats

Les lignes TranscRater du tableau 5.2 présentent les résultats obtenus avec le système de base fondé sur l'extraction des traits prédéfinis. Nous pouvons observer que la meilleure performance est obtenue avec les caractéristiques textuelles POS+LEX+LM (MAE de 22,01 %) alors que l'ajout du SIG n'améliore pas le modèle (MAE de 21,99 %). Cette difficulté à intégrer correctement les caractéristiques issues du signal, dans les modèles de TranscRater, a également été observée par Jalalvand et al. [2016].

En utilisant essentiellement des caractéristiques textuelles, nous constatons que  $\text{CNN}_{Softmax}$  et  $\text{CNN}_{ReLU}$  ont des performances équivalentes (meilleurs en termes de MAE mais moins performantes en termes de Kendall) par rapport au modèle de TranscRater.  $\text{CNN}_{Softmax}$  montre une meilleure performance que  $\text{CNN}_{ReLU}$  en termes de MAE et de coefficient de corrélation.

Toutefois, il faut noter aussi que la tâche de prédiction de performance est difficile en se basant essentiellement sur les caractéristiques acoustiques (MAE supé-

---

2. Les paramètres détaillés des filtres sont représentés dans la Figure 5.1

CHAPITRE 5. IMPLÉMENTATION DES SYSTÈMES DE PRÉDICTION DE PERFORMANCES

Modèle	Entrée	MAE	Kendall
<b>Caractéristiques textuelles (TXT)</b>			
<b>TranscRater</b>	POS+LEX+LM	22,01	<b>44,16</b>
$CNN_{Softmax}$	EMBED	<b>21,48</b>	38,91
$CNN_{ReLU}$	EMBED	22,30	38,13
<b>Caractéristiques acoustiques (SIG)</b>			
<b>TranscRater</b>	SIG	25,86	23,36
$CNN_{Softmax}$	RAW-SIG	25,97	23,61
$CNN_{ReLU}$	RAW-SIG	26,90	21,26
$CNN_{Softmax}$	MEL-SPEC	29,11	19,76
$CNN_{ReLU}$	MEL-SPEC	26,07	24,29
$CNN_{Softmax}$	MFCC	<b>25,52</b>	<b>26,63</b>
$CNN_{ReLU}$	MFCC	26,17	25,41
<b>Caractéristiques textuelles et acoustiques (TXT+SIG)</b>			
<b>TranscRater</b>	POS+LEX+LM+SIG	21,99	45,82
$CNN_{Softmax}$	EMBED+RAW-SIG	<b>19,24</b>	<b>46,83</b>
$CNN_{ReLU}$	EMBED+RAW-SIG	20,56	45,01
$CNN_{Softmax}$	EMBED+MEL-SPEC	20,93	40,96
$CNN_{ReLU}$	EMBED+MEL-SPEC	20,93	44,38
$CNN_{Softmax}$	EMBED+MFCC	19,97	44,71
$CNN_{ReLU}$	EMBED+MFCC	20,32	45,52

TABLE 5.2 – TranscRater *vs*  $CNN_{Softmax}$  *vs*  $CNN_{ReLU}$  évaluées au niveau des tours de parole avec la métrique MAE ou Kendall sur le corpus  $Test_{pred}$

rieur à 25 %). Cependant, parmi les différentes entrées du signal testées, de simples MFCCs conduisent à une meilleure performance en termes de MAE et Kendall. Bien que l'utilisation conjointe de caractéristiques textuelles et acoustiques n'ait pas donné des bons résultats pour la prédiction par TranscRater, elle mène à de meilleures performances en utilisant les CNNs. La meilleure performance est obtenue avec le système  $CNN_{Softmax}$  (EMBED + RAW-SIG)<sup>3</sup> qui dépasse l'approche de régression (le MAE est réduit de 21,99 % à 19,24 %, et la corrélation entre les vrais WER et les WER prédits est améliorée de 45,82 % à 46,83 % en termes de Kendall). Un *test de Wilcoxon*<sup>4</sup> permet de confirmer que la différence entre les

3. Les MAE obtenus sur les 10 modèles sont entre 15.24 % et 15.96 % sur les corpus de développement, tandis que les MAE obtenus sur le corpus  $Test_{Pred}$  sont entre 19.24 % et 20.70 %

4. <http://www.r-tutor.com/elementary-statistics/non-parametric-methods/wilcoxon-signed-rank-test>

WER prédits par ces deux systèmes est significative ( $p < 0,001$ ).

### 5.2.4 Analyse des taux d’erreur de mots prédits

Le tableau 5.3 présente les WER prédits sur le corpus TEST en utilisant les deux approches de prédiction (régression et CNN) pour les différents styles de parole (Spontanée et Non-Spontanée). Les performances montrent que notre approche (à  $-3,83\%$  du WER référence) est meilleure que l’approche de régression ( $-5,38\%$ ) sur l’ensemble du corpus. Les performances montrent que le système CNN a bien prédit le WER sur la parole non-spontanée (NS) et spontanée (S). Le  $TEM_{Pred}$  est à  $-2,54\%$  sur la parole non spontanée et à  $-4,84\%$  sur la parole spontanée. En revanche, la méthode de régression n’arrive pas à bien prédire la performance sur la parole spontanée ( $-10,11\%$ ).

	NS	S	NS + S
$TEM_{REF}$	21,47	38,83	31,20
$TEM_{Pred}$ TranscRater	<b>22,08</b>	28,72	25,82
$TEM_{Pred}$ CNN <sub>Softmax</sub>	18,93	<b>33,99</b>	<b>27,37</b>
#Tours Parole	3,1k	3,7k	6,8k
#Mots <sub>REF</sub>	49,8k	63,3k	113,1k

TABLE 5.3 – TranscRater (POS+LEX+LM+SIG) *vs* CNN<sub>Softmax</sub> (EMBED+RAW-SIG) des WER prédits (moyennés sur toutes les phrases) par type de parole (NS/S) sur le corpus Test<sub>pred</sub>

La figure 5.2 présente l’analyse de prédiction de WER au niveau des tours de parole.<sup>5</sup> Elle montre la distribution des tours de parole en fonction de leur WER réel ou prédit. Il est clair que la prédiction CNN permet d’approximer la vraie distribution de WER sur le corpus Test<sub>pred</sub>. La distribution produite par *TranscRater* ressemble à une distribution gaussienne autour de la moyenne WER observée sur les données d’apprentissage. Il est également intéressant de relever que les deux pics de TEM=0 % et TEM=100 % sont prédits correctement par notre système CNN.

Pour confirmer cette hypothèse, nous avons créé une référence artificielle en attribuant le WER observé sur les données d’apprentissage (22,29 %) à tous les

---

5. Les sorties des modèles sont disponibles sur <http://www.lne.fr/LNE-LIG-WER-Prediction-Corpus>

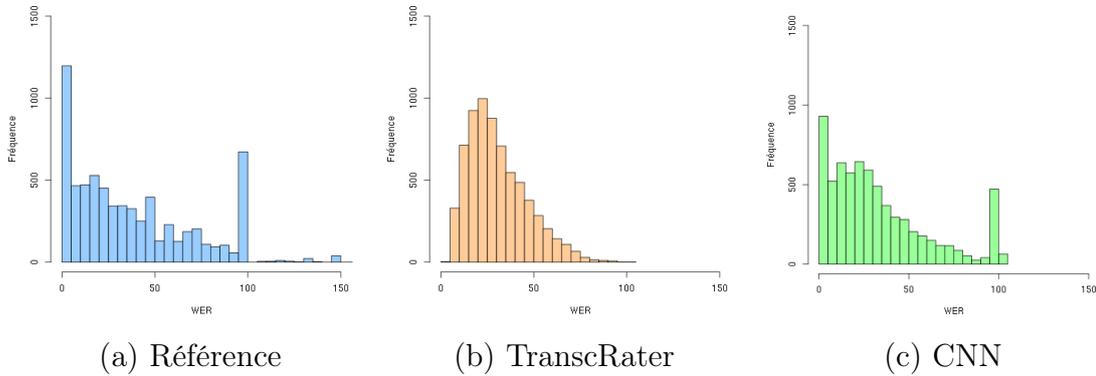


FIGURE 5.2 – Distribution des tours de parole en fonction de leurs WER : (a) Référence (b) Prédit par le meilleur système TranscRater (c) Prédit par le meilleur système CNN

tours de parole de notre corpus  $\text{Test}_{Pred}$ . L'évaluation des résultats de nos systèmes à l'aide de cette référence conduit aux scores MAE suivants : 13,15 % et 21,58 % sur les systèmes TR et CNN respectivement, ce qui confirme notre intuition.

### 5.3 Conclusion

Dans ce chapitre, nous avons abordé la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole (SRAP). Dans un premier temps, nous avons proposé de comparer deux différentes approches de prédiction de performances : une approche par régression (système de référence) nommée TR basée sur des traits pré-définis (*engineered features*) et notre nouvelle approche basée sur des traits estimés au cours de l'apprentissage d'un système neuronal de type CNN (*learnt features*).

Nos expérimentations montrent que l'approche de prédiction par les CNNs est meilleure que l'approche de prédiction par régression en termes de scores MAE et Kendall. Plus précisément, l'utilisation des entrées jointes texte et signal ne donne pas de résultats positifs pour l'approche par régression, tandis qu'elle permet d'obtenir de meilleures performances en utilisant des CNNs. Nous montrons également que les CNNs prédisent correctement la distribution des taux d'erreur de mots sur une collection d'enregistrements, contrairement à l'approche état de l'art qui génère une distribution éloignée de la réalité.

## CHAPITRE 5. IMPLÉMENTATION DES SYSTÈMES DE PRÉDICTION DE PERFORMANCES

---

## CHAPITRE 6

# ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES DE PRÉDICTION DE PERFORMANCES

### Sommaire

---

<b>6.1</b>	<b>Effet de la durée et des styles de parole sur la qualité des SPPs . . . . .</b>	<b>82</b>
6.1.1	Analyse par durée des tours de parole . . . . .	82
6.1.2	Évaluation de l'impact du style de parole sur la qualité des SPPs . . . . .	83
6.1.2.1	Analyse au niveau d'un document complet . . .	83
6.1.2.2	Analyse au niveau des instances d'émission . . .	84
6.1.2.3	Analyse au niveau des types d'émission . . . . .	85
<b>6.2</b>	<b>Évaluation de la robustesse des systèmes de prédiction de performances . . . . .</b>	<b>85</b>
6.2.1	Impact de la taille du corpus d'apprentissage sur la qualité des SPPs . . . . .	87
6.2.2	Effet de la qualité du SRAP ayant généré les données d'apprentissage sur l'apprentissage des SPPs . . . . .	89
<b>6.3</b>	<b>Conclusion . . . . .</b>	<b>91</b>

---

Dans ce chapitre, nous essayons de mieux comprendre le comportement des systèmes de prédiction CNN et TR obtenus dans le chapitre 5 en fonction de différents facteurs.

Nous commençons tout d’abord par évaluer l’impact de la durée des tours de parole et du style de parole (Non Spontanée et Spontanée) sur la qualité des systèmes de prédiction de performances. Nous étudions par la suite l’influence de la quantité des données d’apprentissage ( $\text{Train}_{Pred}$ ) sur la qualité des SPPs. Enfin, nous évaluons la robustesse des systèmes de prédiction lorsque le système est entraîné avec les sorties d’un système de RAP particulier et évalué sur des données transcrites avec un système de reconnaissance automatique de la parole différent.

## 6.1 Effet de la durée et des styles de parole sur la qualité des SPPs

### 6.1.1 Analyse par durée des tours de parole

Dans le tableau 6.1, nous évaluons les systèmes de prédiction TR (POS+LEX+LM+SIG) et CNN ( $\text{CNN}_{Softmax}$  EMBED+RAW-SIG) sur le corpus  $\text{Test}_{Pred}$  (transcrit par  $\text{SRAP}_1$ ) en terme de MAE selon la durée des tours de parole. Les tours de parole (TP) de notre corpus d’évaluation ont été classés suivant leur durée  $d$  en 10 intervalles.

Les performances obtenues montrent que le système CNN génère des prédictions meilleures que le système TR sur tous les intervalles de durée. De plus, nous remarquons que les systèmes de prédiction sont moins performants (des MAE élevés) sur les tours de parole courts (moins de 3 secondes). Cela signifie que les systèmes de prédiction n’ont pas réussi à bien caractériser les tours de parole courts pour bien prédire les performances. Par exemple, tandis que les systèmes TR et CNN atteignent respectivement 43,30% et 33,11% de MAE sur des tours de parole courts de 0 à 1 seconde, ils obtiennent 9,75% et 9,58% respectivement sur des tours de parole de plus de 9 secondes. Enfin, nous observons que la durée des tours de parole est un facteur très important qui influence sur la qualité des systèmes de prédiction.

CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES  
DE PRÉDICTION DE PERFORMANCES

Id	Durée (s)	#TP	TR	CNN
<b>1</b>	$0 \leq d < 1$	712	43,30	33,11
<b>2</b>	$1 \leq d < 2$	1246	31,99	28,52
<b>3</b>	$2 \leq d < 3$	1054	24,02	20,94
<b>4</b>	$3 \leq d < 4$	877	18,57	16,42
<b>5</b>	$4 \leq d < 5$	761	15,16	14,33
<b>6</b>	$5 \leq d < 6$	522	14,50	12,82
<b>7</b>	$6 \leq d < 7$	409	12,99	12,70
<b>8</b>	$7 \leq d < 8$	316	13,35	12,70
<b>9</b>	$8 \leq d < 9$	206	10,88	10,18
<b>10</b>	$d \geq 9$	734	<b>9,75</b>	<b>9,58</b>

TABLE 6.1 – Performances des systèmes TR et CNN évalué sur le corpus  $\text{Test}_{Pred}$  en terme de MAE selon les durées des tours de parole

### 6.1.2 Évaluation de l’impact du style de parole sur la qualité des SPPs

Nous proposons dans cette section d’évaluer l’impact du style de parole sur la qualité des systèmes de prédiction de performances au niveau des différentes granularités : ensemble de données, instance de diffusion et type d’émission.

#### 6.1.2.1 Analyse au niveau d’un document complet

Nous avons commencé par classer les tours de parole de notre corpus  $\text{Test}_{Pred}$  suivant leur style de parole afin de créer les deux ensembles :  $\text{Test}_{NS}$  (4 h 17 de parole non spontanée) et  $\text{Test}_S$  (4 h 42 de parole spontanée).

Style	Durée	#TP	TR	CNN
<b><math>\text{Test}_{NS}</math></b>	4 h 17	3109	<b>17,34</b>	<b>15,62</b>
<b><math>\text{Test}_S</math></b>	4 h 42	3728	25,86	22,25
<b><math>\text{Test}_{Pred}</math> (NS+S)</b>	8 h 59	6837	21,99	19,24

TABLE 6.2 – Performances des systèmes de prédiction TR et CNN évalués sur les deux sous ensembles NS et S et la totalité du corpus  $\text{Test}_{Pred}$  (NS+S) en termes de MAE.

Le tableau 6.2 présente la description des données utilisées pour évaluer les SPPs en terme de durée et nombre de tours de parole (#TP). De plus, il contient

## CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES DE PRÉDICTION DE PERFORMANCES

les performances des deux systèmes de prédiction TR et CNN évalués sur les corpus  $\text{Test}_{NS}$ ,  $\text{Test}_S$  et sur la totalité du corpus  $\text{Test}_{Pred}$ . Les performances obtenues montrent que le système CNN produit des prédictions meilleures que le système TR sur les deux styles de parole. La différence entre les performances des systèmes CNN et TR est respectivement  $-1,72\%$  et  $-3,61\%$  sur de la parole NS et S en terme de MAE. Nous remarquons aussi que les MAE obtenus sur la parole Spontanée ( $\text{Test}_S$ ) sont élevés. Ils sont à  $-8,52\%$  et  $-6,63\%$  de MAE respectivement sur les systèmes TR et CNN par rapport à la parole Non Spontanée ( $\text{Test}_{NS}$ ). Cela signifie que les performances sur les tours de parole ayant un style de parole Spontanée sont difficiles à prédire pour les deux systèmes.

### 6.1.2.2 Analyse au niveau des instances d'émission

Dans cette analyse, nous évaluons l'effet du style de parole sur la qualité des SPPs au niveau des instances d'émissions.

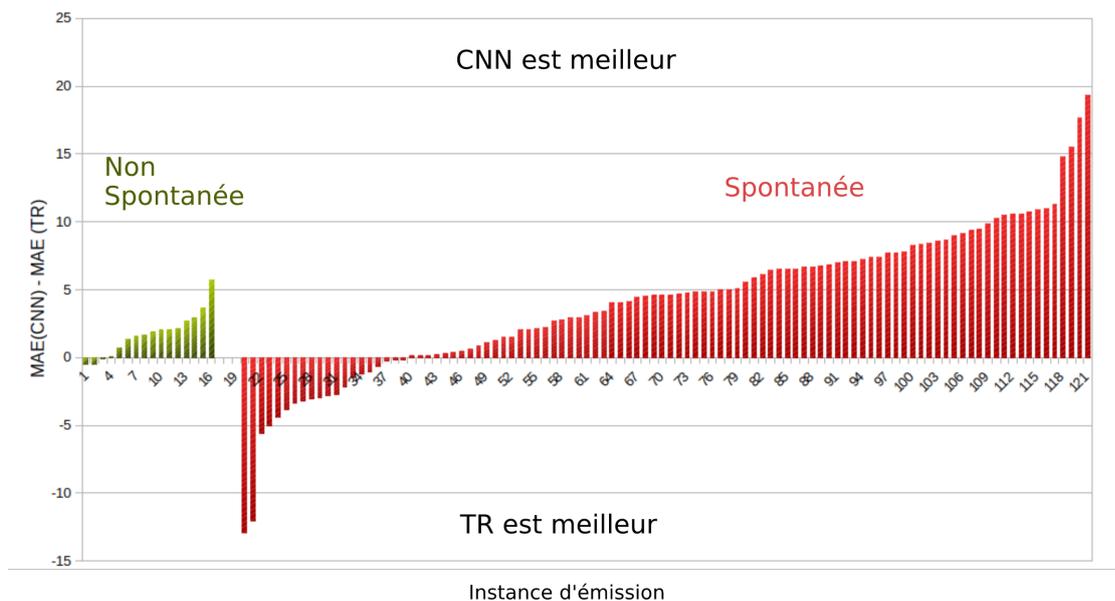


FIGURE 6.1 – Évaluation des systèmes TR et CNN en terme de  $\Delta_{MAE}$  (CNN est meilleur lorsque  $\Delta_{MAE} > 0$ ) sur le corpus  $\text{Test}_1$  (transcrit par  $\text{SRAP}_1$ ) au niveau des instances d'émission pour les styles de parole NS (vert) et S (rouge)

Dans la figure 6.1, nous comparons les systèmes TR et CNN en termes de  $\Delta_{MAE}$  en calculant la différence entre leurs performances ( $\text{MAE}(\text{CNN}) - \text{MAE}(\text{TR})$ ). Si  $\Delta_{MAE}$  est positif, alors le système CNN est meilleur, sinon le système TR est

meilleur. Les résultats obtenus montrent que notre système CNN est meilleur que le système TR sur 80,51% des émissions (95 fichiers *wav* sur 118 fichiers *wav*). De plus, nous remarquons que les prédictions du système CNN sont bonnes pour les styles de parole NS (vert) et S (rouge). Notamment, pour le style S, CNN est meilleur que TR sur 82/102 instances d'émission par une large marge (50 instances d'émission présentent un  $\Delta_{MAE}$  plus grand que 5%). Cela signifie que notre système CNN profond est capable de détecter les caractéristiques des deux styles de parole non spontanée et spontanée et qu'il est capable de prédire des taux d'erreur de mots élevés (pour la parole spontanée) contrairement au système TR (voir la figure 5.2).

### 6.1.2.3 Analyse au niveau des types d'émission

Dans la figure 6.2, nous comparons les systèmes de prédiction CNN et TR en terme de MAE sur le corpus  $\text{Test}_{pred}$  au niveau du type d'émission afin de comprendre l'effet du style de parole sur la tâche de prédiction de performance. Comme décrit, nous avons classé les émissions de nos corpus en deux groupes : Non Spontanée (NS) et Spontanée (S). Les performances obtenues confirment que la performance sur la parole spontanée est plus difficile à prédire que sur la parole *NS*. Dans la partie spontanée, nous remarquons que l'écart entre la courbe CNN et la courbe TR est plus large que pour la parole Non Spontanée. Cela signifie que le système CNN est capable de prédire un WER élevé, alors que TR prédit une performance autour du WER moyen observé sur les données d'entraînement  $\text{Train}_1$ .

## 6.2 Évaluation de la robustesse des systèmes de prédiction de performances

Précédemment, nous avons utilisé notre meilleur système de reconnaissance automatique de la parole  $\text{SRAP}_1$  (un modèle acoustique hybride HMM-DNN avec un modèle de langage 5-grammes) pour obtenir les transcriptions automatiques afin d'apprendre et d'évaluer les systèmes de prédiction de performances.

Dans cette section, nous visons à évaluer la robustesse des systèmes de prédiction appris et/ou évalués sur de nouvelles sorties issues d'autres systèmes  $\text{SRAP}$ .

## CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES DE PRÉDICTION DE PERFORMANCES

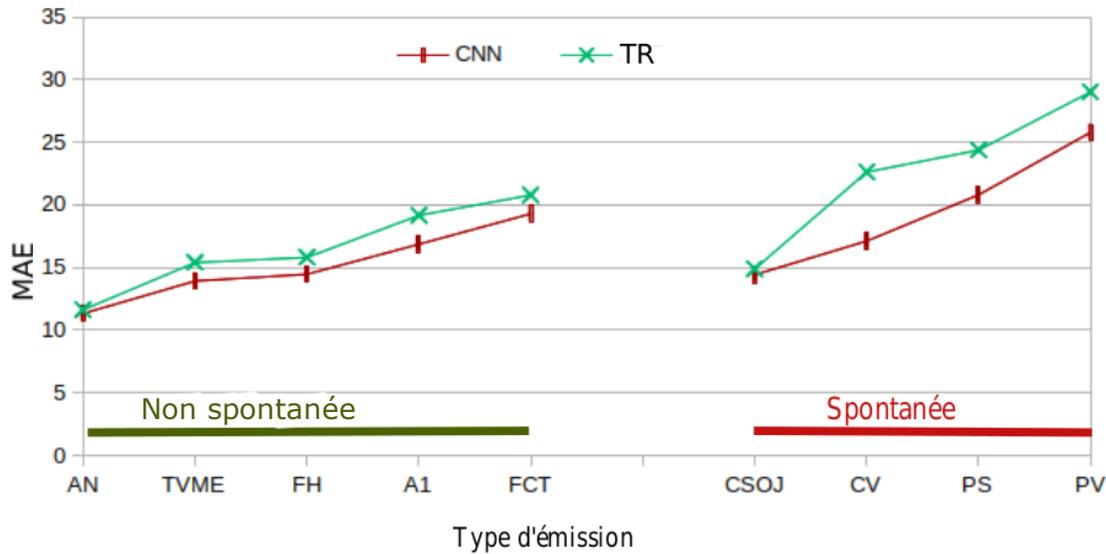


FIGURE 6.2 – Évaluation des systèmes de prédiction sur le corpus  $\text{Test}_1$  (transcrit avec  $\text{SRAP}_1$ ) en termes de MAE au niveau du type d'émission

Tandis que les systèmes  $\text{SRAP}_1$  et  $\text{SRAP}_2$  ont des performances proches, nous avons créé deux nouveaux systèmes de RAP supplémentaires moins performants nommées  $\text{SRAP}_3$  et  $\text{SRAP}_4$  obtenus avec des modèles acoustiques tri-phone GMM et SGMM (voir section 4.4.2).

Systèmes	MA	ML	$\text{Train}_i$	$\text{Test}_i$
$\text{SRAP}_1$	DNN	5-grammes	<b>22,29</b>	<b>31,20</b>
$\text{SRAP}_2$	DNN	3-grammes	23,64	32,80
$\text{SRAP}_3$	SGMM	3-grammes	24,58	34,01
$\text{SRAP}_4$	GMM	3-grammes	27,02	36,79

TABLE 6.3 – Description des 4 systèmes de reconnaissance automatique de la parole produits et leurs performances évalués sur nos corpus pour tâche prédiction de performance  $\text{Train}_i$  et  $\text{Test}_i$  en termes de WER-  $i$  l'id du SRAP utilisé pour la transcription automatique

Le tableau 6.3 présente les composants principaux de chaque SRAP (modèle acoustique et modèle de langage) ainsi que les performances des SRAP produits pour étudier la robustesse des systèmes de prédiction. Les systèmes de RAP ont été évalués sur les corpus  $\text{Train}_i$  et  $\text{Test}_i$  ( $i$  l'identifiant du système de RAP utilisé pour la transcription automatique -  $i = 1, 2, 3, 4$ ). Les résultats obtenus montrent une

différence de  $-4,73\%$  et  $-5,59\%$  de WER entre les systèmes SRAP<sub>1</sub> (le meilleur système) et SRAP<sub>4</sub> (le système le moins performant) évalués respectivement sur les corpus Train<sub>*i*</sub> et Test<sub>*i*</sub>.

### 6.2.1 Impact de la taille du corpus d'apprentissage sur la qualité des SPPs

La taille des données d'apprentissage et son influence sur la qualité des systèmes reste toujours une question importante pour de nombreuses tâches en traitement automatique du langage (reconnaissance de la parole, traduction automatique, classification des images, etc.).

Dans cette sous-section, notre objectif est d'étudier l'impact de la quantité des données d'apprentissage sur nos systèmes de prédiction de performances (TR et CNN).

D'abord, nous avons créé deux nouveaux corpus nommés Train<sub>20%</sub> et Train<sub>50%</sub> en sélectionnant au hasard 20% et 50% à partir du corpus Train<sub>Pred</sub> tout en respectant la même distribution des données (en termes de durée et type d'émission).

Comme décrit dans le tableau 6.4, nous avons évalué la qualité de transcription automatique du système SRAP<sub>1</sub> sur les corpus Train<sub>100%</sub> (Train<sub>Pred</sub>), Train<sub>50%</sub> et Train<sub>20%</sub>. Les performances obtenues montrent que les WER sur les sous-échantillons sont stables (néanmoins, le corpus Train<sub>20%</sub> obtient le meilleur WER (21.50%) par rapport aux autres corpus).

Corpus	Durée	#TP	SRAP1
<b>Train</b> <sub>100%</sub> (Train <sub>Pred</sub> )	90 h	75 k	22,29
<b>Train</b> <sub>50%</sub>	45 h	35 k	22,40
<b>Train</b> <sub>20%</sub>	18 h	15 k	<b>21,50</b>

TABLE 6.4 – Évaluation du système SRAP<sub>1</sub> sur des sous-échantillons corpus d'apprentissage des systèmes de prédiction en terme de WER

En exploitant les corpus Train<sub>20%</sub>, Train<sub>50%</sub> et Train<sub>100%</sub>, nous avons construit des nouveaux systèmes de prédiction de performances nommés TR<sub>*y*</sub> et CNN<sub>*y*</sub> (*y* est la quantité de données utilisée pour l'apprentissage - *y* = 20%, 50%, 100%). Les systèmes TR<sub>100%</sub> et CNN<sub>100%</sub> correspondent respectivement aux meilleurs systèmes de prédiction TR (POS+LEX+LM+SIG) et CNN<sub>Softmax</sub> (EMBED+RAW-SIG)

## CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES DE PRÉDICTION DE PERFORMANCES

décrits dans le tableau 5.2. Ces systèmes de prédiction ont été évalués sur les données  $Test_i$  transcrites automatiquement en utilisant les systèmes  $SRAP_i$  ( $i=1, 2, 3, 4$ ).

Corpus	TR-100%	TR-50%	TR-%20
<b>Test<sub>1</sub></b>	<b>21,99</b>	<b>22,50</b>	<b>21,81</b>
<b>Test<sub>2</sub></b>	22,15	22,67	22,01
<b>Test<sub>3</sub></b>	23,23	23,68	22,94
<b>Test<sub>4</sub></b>	23,00	23,43	22,64

TABLE 6.5 – Évaluation des nouveaux systèmes **TR** sur 4 corpus d'évaluation  $Test_i$  ( $ASR_i$ ) en termes de MAE

Corpus	CNN-100%	CNN-50%	CNN-%20
<b>Test<sub>1</sub></b>	<b>19,24</b>	<b>20,55</b>	<b>21,53</b>
<b>Test<sub>2</sub></b>	19,67	20,79	21,87
<b>Test<sub>3</sub></b>	20,64	21,70	22,90
<b>Test<sub>4</sub></b>	21,34	22,44	23,62

TABLE 6.6 – Évaluation des nouveaux systèmes **CNN** sur 4 corpus d'évaluation  $Test_i$  ( $ASR_i$ ) en termes de MAE

Les tableaux 6.5 et 6.6 résument les résultats expérimentaux obtenus avec 6 systèmes de prédiction de performances (3 systèmes TR et 3 systèmes CNN) appris sur les ensembles de données  $Train_y$  ( $y=100\%$ ,  $50\%$  et  $20\%$ ) transcrites avec le système  $SRAP_1$ . Ces systèmes ont été évalués sur 4 corpus  $Test_i$  afin d'évaluer la robustesse des systèmes PP en termes de MAE. Nous insistons sur le fait que tous les ensembles d'évaluation ( $Test_i$ ) correspondent à la même collection acoustique, la seule différence est que les données textuelles correspondent à des sorties SRAP différentes (voir tableau 6.3).

Tout d'abord, nous remarquons que les systèmes CNN sont meilleurs que tous les systèmes TR en termes de MAE pour 11 conditions d'apprentissage ( $Train_y$ )/évaluation ( $Test_i$ ) sur 12 (sauf  $Train-20\%/Test_4$ ).

Si nous nous concentrons sur la différence entre les ensembles d'évaluation  $Test_i$  (lignes), les résultats montrent que  $Test_1$  a obtenu la meilleure prédiction en terme de MAE sur les systèmes CNN et TR, sachant que  $Test_1$  a la meilleure qualité de sortie SRAP (WER de 31.20%) dans le tableau 6.3. Nous remarquons également

que la qualité des sorties SRAP (voir Tableau 6.3) et la qualité des systèmes de prédiction semblent corrélées (lorsque la qualité des SRAP est inférieure - par exemple  $i = 4$ ; le MAE des systèmes de prédiction augmente). Cela confirme la tendance, déjà remarquée pour la parole spontanée, qu'il est plus difficile de prédire des WER plus élevés. Quoi qu'il en soit, il est intéressant de noter qu'un système PP appris pour un système ASR particulier ( $ASR_1$  par exemple) n'est pas trop dégradé lorsqu'il est appliqué sur les sorties ASR obtenues avec un système de transcription différent ( $ASR_i$  pour  $i=2,3,4$  par exemple).

En examinant la quantité de données d'apprentissage (colonnes), nous observons que la réduction de la taille des ensembles des données d'apprentissage augmente le MAE pour le système CNN. Par exemple, sur le corpus  $Test_1$ , nous avons obtenu respectivement 19,24% et 21,53% sur les systèmes CNN-100% et CNN-20% en terme de MAE. Cela signifie que la taille de l'ensemble d'apprentissage a une forte influence sur la performance des systèmes de prédiction fondés sur des réseaux de neurones convolutifs.

Contrairement aux systèmes CNN, le tableau 6.5 montre que l'approche TR n'est pas trop dégradée lorsque la taille des données d'apprentissage diminue et que le système  $TR_{20\%}$  a la meilleure qualité de prédiction par rapport à  $TR_{100\%}$  et  $TR_{50\%}$ .

### 6.2.2 Effet de la qualité du SRAP ayant généré les données d'apprentissage sur l'apprentissage des SPPs

Dans cette analyse, nous visons à étudier l'effet de la qualité des systèmes de reconnaissance automatique de la parole (les transcriptions automatiques) sur la qualité des systèmes de prédiction de performances. Cette évaluation consiste à estimer la robustesse des systèmes prédictifs lorsque le système de prédiction est appris avec les sorties d'un SRAP particulier et utilisé pour prédire la performance sur de nouvelles données (transcrites avec le même ou un nouveau système de RAP).

Nous avons créé 4 systèmes de prédiction pour chaque approche de prédiction nommée  $TR_i$  et  $CNN_i$  en utilisant les transcriptions automatiques  $Train_i$ , afin de les évaluer sur les ensembles de données  $Test_i$  ( $i$  est l'id du SRAP utilisé -  $i = 1, 2, 3, 4$ ). Nous obtenons ainsi une matrice de performance 4x4 pour chaque

## CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES DE PRÉDICTION DE PERFORMANCES

---

système PP. Les résultats sont décrits dans le tableau 6.7 et le tableau 6.8.

SPP	Test <sub>1</sub>	Test <sub>2</sub>	Test <sub>3</sub>	Test <sub>4</sub>
TR <sub>1</sub>	21,99	22,15	23,33	23,00
TR <sub>2</sub>	21,68	21,72	22,67	22,33
TR <sub>3</sub>	21,62	21,67	<b>22,37</b>	22,13
TR <sub>4</sub>	<b>21,58</b>	<b>21,60</b>	22,66	<b>21,95</b>

TABLE 6.7 – Effet de la qualité des sorties des systèmes de RAP sur la performance des systèmes TR en termes de MAE

SPP	Test <sub>1</sub>	Test <sub>2</sub>	Test <sub>3</sub>	Test <sub>4</sub>
CNN <sub>1</sub>	<b>19,24</b>	19,67	20,64	21,34
CNN <sub>2</sub>	19,75	19,78	20,54	21,18
CNN <sub>3</sub>	19,87	19,81	20,62	21,39
CNN <sub>4</sub>	19,26	<b>19,28</b>	<b>19,94</b>	<b>20,22</b>

TABLE 6.8 – Effet de la qualité des sorties des systèmes de RAP sur la performance des systèmes CNN en termes de MAE

Les résultats expérimentaux montrent que les deux systèmes de prédiction (CNN et TR) sont plutôt stables quelle que soit la qualité du système de RAP au moment de l'apprentissage ( $Train_i$ ). Il est remarquable de noter que les systèmes TR et CNN appris sur  $Train_4$  sont légèrement meilleurs pour prédire la performance sur une nouvelle collection de données.  $Train_4$  est le corpus où les WER sont les plus élevés par rapport aux corpus  $Train_1$   $Train_2$   $Train_3$  avec un WER de 36.79% (voir le tableau 6.3). Cela permet sans doute aux systèmes de prédiction d'avoir plus d'exemples d'erreurs (au moment de l'apprentissage) qui peuvent être produites par un système de reconnaissance automatique de la parole. Cette analyse est importante pour la portabilité et l'utilisation des systèmes de prédiction des performances dans des scénarios pratiques.

### 6.3 Conclusion

Dans ce chapitre, nous avons proposé une analyse plus détaillée de la robustesse de deux approches de prédiction de performances des SRAP (CNN et TR).

Nous avons commencé à étudier l'effet de la durée des tours de parole et du style de parole (spontanée et non spontanée) sur la qualité des systèmes de prédiction. Les résultats expérimentaux montrent que la prédiction est plus difficile sur les tours de parole courts ainsi que sur la parole spontanée.

Nous avons également évalué l'impact de la quantité de données d'apprentissage des systèmes de prédiction. Nous avons constaté que le système CNN est plus sensible que le système TR à la réduction de la quantité des données d'apprentissage.

Enfin, nous avons étudié la robustesse des systèmes de prédiction lorsque le système est appris avec les sorties d'un système SRAP particulier et utilisé pour prédire la performance sur des nouveaux enregistrements (jamais rencontrés auparavant) transcrits avec des nouveaux systèmes de RAP (plus ou moins performants). Nous avons montré que les systèmes de prédiction de performance sont plutôt robustes, quelle que soit la qualité de sortie des SRAP au moment de l'apprentissage, il semble même qu'un système de RAP de qualité moyenne permet de rencontrer plus d'exemples d'erreurs au moment de l'apprentissage et donne, par conséquent des systèmes de prédiction légèrement plus performants.

CHAPITRE 6. ANALYSE DES FACTEURS IMPACTANT NOS SYSTÈMES  
DE PRÉDICTION DE PERFORMANCES

---

## CHAPITRE 7

# ÉVALUATION DES REPRÉSENTATIONS APPRISSES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

### Sommaire

---

<b>7.1</b>	<b>Travaux existants</b>	<b>94</b>
<b>7.2</b>	<b>Méthodologie</b>	<b>95</b>
<b>7.3</b>	<b>Analyse par classification</b>	<b>96</b>
7.3.1	Classifieur peu profond pour l'analyse	96
7.3.2	Données	97
7.3.3	Résultats	98
<b>7.4</b>	<b>Analyse par visualisation</b>	<b>101</b>
<b>7.5</b>	<b>Apprentissage multi-tâche</b>	<b>101</b>
<b>7.6</b>	<b>Conclusion</b>	<b>104</b>

---

Nous avons présenté dans le chapitre 5 une étude comparative sur la tâche de prédiction de performances entre une approche fondée sur des caractéristiques pré-définies (en utilisant l’outil TranscRater) et notre nouvelle méthode de prédiction fondée sur des caractéristiques apprises à l’aide des réseaux de neurones convolutifs. Les résultats montrent que le  $\text{CNN}_{\text{Softmax}}$  ( en entrée :  $\text{EMBED}_{\text{Mot}} + \text{RAW-SIG}$ ) est le meilleur système obtenu avec un MAE de 19.24%.

Dans ce chapitre, nous essayons de comprendre quelles informations sont capturées par notre meilleur modèle CNN profond et leurs liens avec différents facteurs. D’abord, nous présentons un état de l’art sur la tâche d’analyse des représentations intermédiaires des modèles profonds. Nous détaillons ensuite la méthodologie suivie, le classifieur ainsi que les données utilisées pour analyser et évaluer des représentations intermédiaires apprises dans les tâches de classifications annexes. Enfin, nous étudions le potentiel d’un apprentissage structuré consistant à donner implicitement les informations connues au moment de l’entraînement du système de prédiction via un apprentissage multi-tâche.

### 7.1 Travaux existants

En apprentissage profond , il est important d’interpréter les représentations intermédiaires apprises par le réseau afin de comprendre quelles informations ont été capturées. Des travaux récents sur la tâche de reconnaissance automatique de la parole ont proposé d’analyser les représentations capturées par les SRAP profonds. [Mohamed et al. \[2012\]](#) et [Belinkov and Glass \[2017\]](#) ont analysé les représentations intermédiaires apprises (d’un SRAP profond) en utilisant la visualisation t-SNE [[Maaten and Hinton, 2008](#)]. Ils essaient aussi de comprendre quelles couches capturent mieux les informations phonétiques en entraînant un classifieur de phonèmes peu profond. Par ailleurs, [Wu and King \[2016\]](#) ont évalué les représentations de plusieurs variantes de LSTM pour une tâche de synthèse vocale. [Wang et al. \[2017\]](#) ont quant à eux proposé une étude sur trois types de représentations apprises pour une tâche de reconnaissance de locuteur : i-vecteur, d-vecteur et s-vecteur (basée sur un réseau RNN/LSTM). Des tâches de classification annexes ont été conçues pour mieux comprendre comment sont encodées les informations sur les locuteurs. Également, un apprentissage multi-tâche est

proposé pour intégrer ces différents types de représentations, ce qui mène à une meilleure performance d'identification du locuteur. Nous trouvons aussi des travaux similaires dans d'autres applications du traitement automatique du langage naturel (TALN), comme en traduction automatique neuronale par exemple. Parmi ces travaux récents, nous pouvons citer les travaux de [Shi et al. \[2016\]](#) et [Belinkov et al. \[2017\]](#) qui ont essayé de comprendre les représentations apprises par un système de traduction neuronal. Ces représentations sont fournies à un classifieur peu profond afin de prédire des étiquettes syntaxiques [[Shi et al., 2016](#)], grammaticales ou sémantiques [[Belinkov et al., 2017](#)]. L'analyse montre que les couches inférieures sont meilleures pour l'étiquetage grammatical, tandis que les couches supérieures sont meilleures pour l'étiquetage sémantique.

## 7.2 Méthodologie

Dans cette section, nous essayons de comprendre ce que notre meilleur système de prédiction de performance (EMBED+RAW-SIG) a appris. Nous analysons les représentations textuelles et acoustiques obtenues par notre architecture. Nous nous inspirons de travaux de [Belinkov and Glass \[2017\]](#) : le modèle pré-entraîné (EMBED+RAW-SIG) est utilisé pour générer des représentations au niveau des tours de parole. Nous nous intéressons à l'analyse des représentations qui correspondent à différentes couches supérieures de notre réseau (colorées en jaune dans la figure 7.1). Ces représentations sont utilisées par la suite pour entraîner un classifieur peu profond et résoudre des tâches de classification annexes telles que :

- **STYLE** : classer les tours de parole entre les styles de parole (spontanée et non spontanée) (voir le tableau 7.1) ;
- **ACCENT** : classer les tours de parole entre locuteur natif et non natif (comme il est indiqué dans le tableau 7.1), nous avons utilisé les annotations des locuteurs fournies avec nos données afin d'étiqueter nos tours de parole entre natif/non natif ;
- **EMISSION** : classer les tours de parole suivant les émissions. Comme cela est décrit dans le tableau 7.2, chaque tour de parole de notre corpus est étiqueté avec le nom de l'émission.

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISSES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

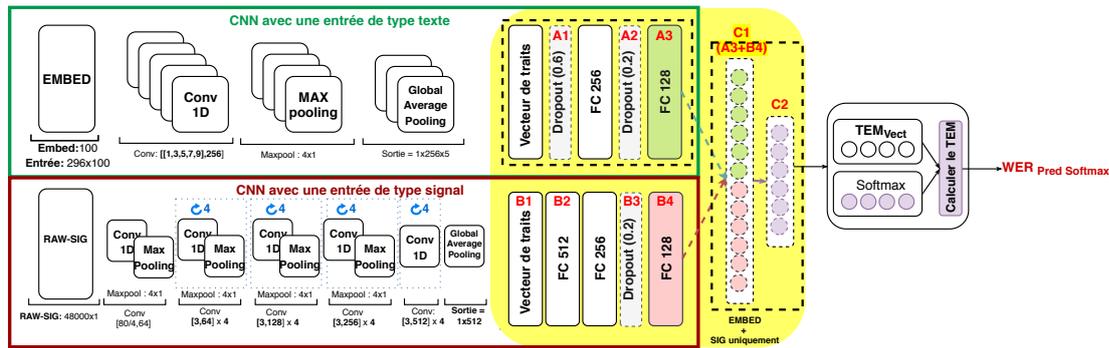


FIGURE 7.1 – Architecture de notre CNN - en jaune les couches de représentations qui vont être analysées

Les performances de ces classifieurs peu profonds nous permettront de savoir quelles informations (style, accent, émission) sont le mieux capturées par les différentes couches du réseau ; c'est-à-dire ce que modélise un réseau CNN qui prédit les performances d'un SRAP.

Comme analyse visuelle, nous projetons également un exemple des représentations dans un espace à deux dimensions à l'aide de l'algorithme t-SNE (t-Distributed Stochastic Neighbor Embedding) [Maaten and Hinton, 2008].

### 7.3 Analyse par classification

#### 7.3.1 Classifieur peu profond pour l'analyse

Nous avons construit trois classifieurs peu profonds (EMISSION, STYLE, ACCENT) avec une architecture similaire. Le classifieur est un réseau neuronal supervisé avec une seule couche cachée (la taille de la couche cachée est fixée à 128) suivie d'un *Dropout* (taux de 0,5) et d'une non-linéarité *ReLU*. Enfin, une couche *Softmax* est utilisée afin de convertir la sortie du réseau en une catégorie prédite. Nous avons choisi un classifieur simple et peu profond car nous nous intéressons à l'évaluation de la qualité des représentations apprises par notre modèle de prédiction SRAP, plutôt qu'à l'optimisation des tâches de classification secondaires. La taille de l'entrée du réseau dépend de la couche à analyser (voir figure 7.1).

L'apprentissage est effectué en utilisant l'algorithme *Adam* [Kingma and Ba, 2014] (en utilisant les paramètres par défaut) sur des mini-batches de taille 16. La fonction de coût est l'entropie croisée. Les modèles sont entraînés avec 30 époques.

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

Après l'apprentissage, nous conservons le modèle ayant les meilleures performances sur l'ensemble DEV et nous l'évaluons sur le corpus TEST (voir section suivante pour détail sur DEV et TEST). Les sorties du classifieur sont évaluées en terme de taux de bonne classification (*accuracy*).

### 7.3.2 Données

Nous avons utilisé les mêmes données que celles proposées dans la section 4.2. Nous récupérons tout d'abord les corpus d'apprentissage (TRAIN) et de développement (DEV) du meilleur modèle obtenu (EMBED+RAW-SIG), tout en gardant le même corpus de TEST ( $\text{Test}_{Pred}$ ), sachant que les émissions du corpus  $\text{Test}_{Pred}$  n'existent ni dans le corpus TRAIN ni dans le corpus DEV.

Catégorie	TRAIN	DEV	TEST
Non spontané	54 250	6 101	<b>3 109</b>
Spontané	<b>13 277</b>	<b>1 403</b>	3 728
Native	44 487	4 945	5 298
Non native	<b>23 040</b>	<b>2 559</b>	<b>1 539</b>

TABLE 7.1 – Distribution des tours de parole entre les styles non spontanés et spontanés et accents natifs/non natifs

Émission	TRAIN	DEV	TEST
FINTER-DEBATE	7 632	833	-
FRANCE3-DEBATE	928	77	-
LCP-PileEtFace	<b>4 487</b>	525	-
TELSONNE	4 717	<b>493</b>	-
RFI	25 565	2 831	-
RTM	24 198	2 745	-
<b>Total</b>	67 527	7 504	-

TABLE 7.2 – Nombre des tours de parole pour chaque émission

Les tableaux 7.1 et 7.2 décrivent l'ensemble des données disponibles en termes de tours de parole pour chaque tâche de classification. Nous constatons clairement que les données sont déséquilibrées pour les trois catégories (STYLE, ACCENT, EMISSION). Étant donné que nous nous intéressons à évaluer le pouvoir discriminant de nos représentations apprises pour ces 3 tâches, nous avons extrait une version équilibrée de nos données TRAIN/DEV/TEST en filtrant les étiquettes

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

---

sur-représentées (le nombre final de tours de parole conservés correspond aux nombres en gras dans les tableaux 7.1 et 7.2). Le corpus TEST ne contient aucun type d'émission présent dans le tableau 7.2, car selon notre protocole expérimental, les émissions du corpus TEST (voir tableau 4.9) n'existent pas dans les corpus TRAIN/DEV et vice versa.

Le tableau 7.3 montre la distribution de nos corpus TRAIN/DEV/TEST équilibrés définitifs ainsi que le nombre de catégories pour chaque tâche. Pour la tâche de classification ÉMISSION, les tours de parole de l'émission *FRANCE3-DEBATE* ont été supprimés puisqu'ils représentent une trop petite quantité de données.

	#Catégories	Tours de parole par catégorie		
		TRAIN	DEV	TEST
<b>EMISSION</b>	5	4 487 <sub>×5</sub>	493 <sub>×5</sub>	-
<b>STYLE</b>	2	13 277 <sub>×2</sub>	1 403 <sub>×2</sub>	3 109 <sub>×2</sub>
<b>ACCENT</b>	2	23 040 <sub>×2</sub>	2 559 <sub>×2</sub>	1 539 <sub>×2</sub>

TABLE 7.3 – Description de notre ensemble de données équilibré pour chaque catégorie

### 7.3.3 Résultats

Pour chaque tâche de classification, nous avons construit un classifieur peu profond en utilisant les représentations cachées des caractéristiques *EMBED* (texte), *RAW-SIG* (signal) et *EMBED+RAW-SIG* en entrée. Le tableau 7.4 présente les résultats expérimentaux obtenus sur les corpus DEV et TEST séparés par deux barres verticales (||). Les performances des systèmes de classification sont toutes supérieures à un taux de bonne classification correspondant à une décision aléatoire ( $> 50\%$  pour les tâches STYLE et ACCENT et  $> 20\%$  pour la tâche EMISSION). Cela montre que l'apprentissage d'un système de prédiction de WER profond produit des représentations (au niveau des couches) qui contiennent une quantité significative d'informations sur le style de parole, l'accent du locuteur ainsi que sur l'émission. La prédiction du style des tours de parole (spontané ou non spontané) est légèrement plus facile que la prédiction de l'accent (natif/non-natif), en particulier à partir de l'entrée de type texte (EMBED). Cela pourrait être lié à la durée courte ( $< 6$  s) des tours de parole, étant donné que l'identification de l'accent a probablement besoin de séquences plus longues. Nous observons égale-

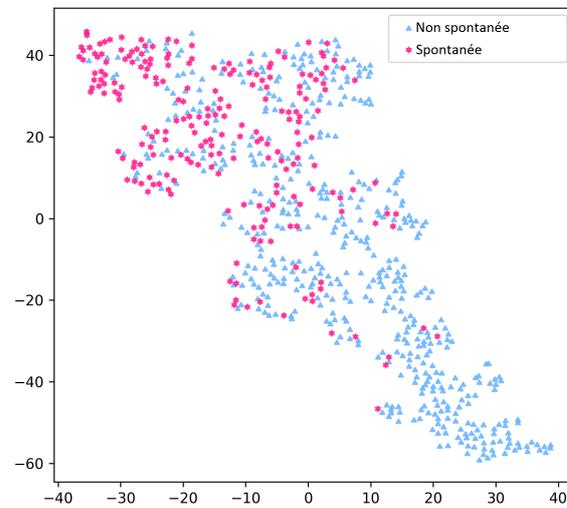
## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

ment que l'utilisation du texte et de la parole améliore les représentations apprises pour la tâche STYLE alors que cela est moins clair pour la tâche ACCENT (étant donné que l'amélioration observée sur DEV n'est pas confirmée sur TEST).

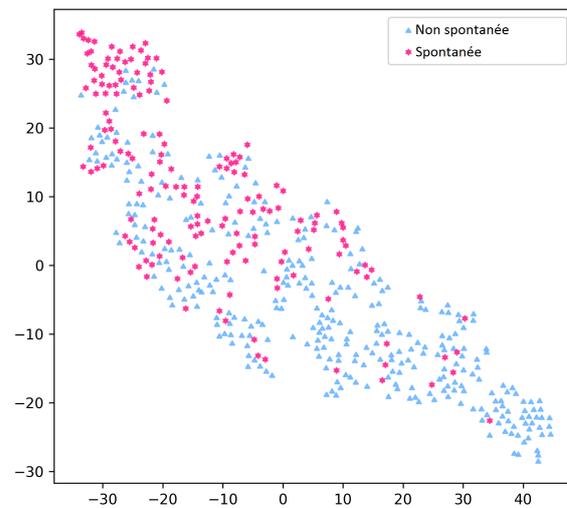
Enfin, l'entrée textuelle est significativement meilleure que l'entrée acoustique pour toutes les tâches de classification, alors que nous anticipions de meilleures performances sur l'entrée acoustique pour la tâche EMISSION (le signal audio transmet des informations sur les caractéristiques acoustiques d'un programme diffusé). Parmi les représentations analysées, les sorties des CNN (A1, B1) conduisent aux meilleurs résultats de classification, ceci est cohérent avec les résultats de la littérature qui présentent les convolutions comme de bons extracteurs de traits. En utilisant les couches supérieures (entièrement connectées), nous remarquons que la performance se dégrade. Cela signifie que l'information sur le style de parole, l'accent du locuteur ou l'émission est plutôt capturée dans les couches moins hautes de notre architecture neuronale de prédiction de performances de SRAP.

Couche	Dim.	EMISSION	STYLE	ACCENT
EMBED				
A1	1280	<b>57,12</b>   -	<b>80,72</b>   68,99	<b>70,75</b>   66,54
A2	256	54,89  -	80,01   <b>69,56</b>	69,30  69,43
A3	128	51,04  -	79,23  68,27	68,25   <b>70,89</b>
RAW-SIG				
B1	512	<b>42,35</b>   -	<b>72,92</b>    <b>58,64</b>	64,60   <b>55,85</b>
B2	512	41,22  -	72,20  58,41	64,44  54,84
B3	256	41,22  -	72,38  58,44	64,50  54,65
B4	128	40,77  -	72,38  58,52	<b>64,74</b>   54,87
EMBED + RAW-SIG				
C1 (A3+B4)	256	<b>57,04</b>   -	<b>81,29</b>   70,36	<b>71,41</b>    <b>65,98</b>
C2	128	53,06  -	79,62   <b>70,55</b>	70,01  65,20
<b>Aléatoire</b>	-	<b>20,00</b>	<b>50,00</b>	<b>50,00</b>

TABLE 7.4 – Performances des systèmes de classification Émission/Style/Accent en termes de taux de bonne classification en utilisant les représentations apprises durant l'apprentissage de notre système de prédiction



(a) de 4 à 5 secondes



(b) de 5 à 6 secondes

FIGURE 7.2 – Visualisation des représentations des tours de parole de la couche C2 pour les différents styles de parole (Spontanée/Non spontanée). (a) des tours de parole ayant une durée de 4 à 5 s et (b) de 5 à 6 s

## 7.4 Analyse par visualisation

La méthode t-SNE (t-Distributed Stochastic Neighbor Embedding) [Maaten and Hinton, 2008] est un algorithme de visualisation des données fondé sur des interprétations probabilistes. C’est une méthode non-linéaire qui a pour objectif de projeter des représentations à haute-dimension dans un espace de deux ou trois dimensions. Ces représentations peuvent être par la suite visualisés sous forme d’un nuage de points.

En utilisant l’algorithme t-SNE<sup>1</sup>, nous projetons les activations de notre corpus TEST dans un espace de 2 dimensions afin de les visualiser.

Dans la figure 7.2, nous visualisons un exemple de représentations des tours de parole de la couche C2 (EMBED+RAW-SIG) en utilisant t-SNE. Pour une durée fixe de 4 à 5 s (716 tours de parole) et de 5 à 6 s (489 tours de parole), les tours de parole non spontanée sont colorées en bleu tandis que les tours de parole spontanée sont en rose. La couche C2 produit des *clusters* qui montrent que les tours de parole spontanés se trouvent dans la partie supérieure gauche de l’espace 2D. Cela suggère que la représentation cachée C2 véhicule une information (signal faible) sur le style de parole.

Enfin, la figure 7.3 présente la matrice de confusion produite à l’aide de la couche C2 (EMBED+RAW-SIG). Les classifieurs ont très bien prédit la catégorie *TELSONNE* (taux de bonne classification de 82%), qui contient de nombreux appels téléphoniques des auditeurs de la radio. Cette émission est assez différente des 4 autres émissions de DEV (débat et actualités).

## 7.5 Apprentissage multi-tâche

Dans la section précédente, nous avons montré que les couches cachées de notre système de prédiction capturent une information sur le style de la parole, l’accent et le type d’émission. Cela suggère que ces trois types d’informations pourraient être utiles pour structurer l’apprentissage des modèles neuronaux de prédiction de performance. Dans cette section, nous examinons l’impact de la connaissance de ces étiquettes (style, accent, émission) au moment de l’apprentissage sur les performances des systèmes de prédiction. Pour cela, nous effectuons un appren-

---

1. [https://lvdmaaten.github.io/tsne/code/tsne\\_python.zip](https://lvdmaaten.github.io/tsne/code/tsne_python.zip)

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

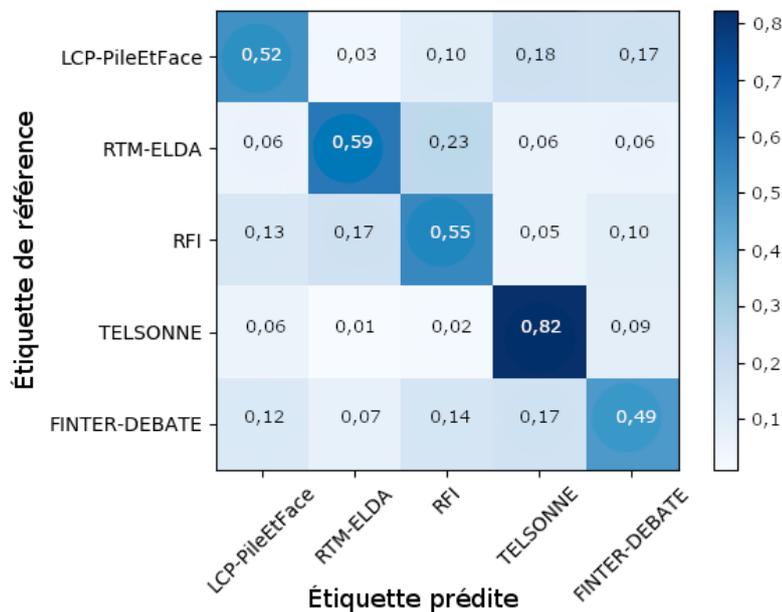


FIGURE 7.3 – Matrice de confusion de la classification EMISSION en utilisant les représentations de la couche C2 (EMBED+RAW-SIG) comme entrée - évaluée sur le corpus DEV

tissage multi-tâche en fournissant des informations supplémentaires sur le type d’émission, le style de parole ainsi que l’accent du locuteur pendant l’apprentissage. L’architecture du modèle multi-tâche est similaire au modèle de prédiction de WER (mono-tâche) présenté dans la figure 5.1 en ajoutant des sorties supplémentaires : une fonction Softmax est ajoutée pour chaque nouvelle tâche de classification après la dernière couche entièrement connectée (C2). La dimension de sortie dépend essentiellement de la tâche visée : 6 pour les tâches EMISSION et 2 pour les tâches STYLE et ACCENT.

Nous utilisons la totalité des données (non équilibrées) décrites dans les tableaux 7.1 et 7.2. L’entraînement du modèle multi-tâche utilise *Adadelta*. Les modèles sont appris pendant 50 époques avec une taille de minibatch de 32. La métrique MAE est utilisé comme fonction de coût pour la tâche de prédiction, tandis que l’entropie croisée est utilisée pour les tâches de classification secondaires. Nous définissons aussi une fonction de coût composite dans le cas de l’apprentissage multi-tâche : nous attribuons une pondération de 1 pour le coût MAE (tâche principale) et une pondération plus petite de 0,3 pour le (les) coût(s) d’entropie

CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR  
LE SYSTÈME DE PRÉDICTION NEURONAL

croisée (tâche de classification secondaire).

Après la phase d'apprentissage, nous prenons le modèle qui donne le meilleur *MAE* sur le corpus *DEV* et nous l'évaluons sur le corpus *TEST*. Nous expérimentons plusieurs modèles qui traitent simultanément les 1, 2, 3 et 4 tâches. Les modèles sont évalués avec une métrique spécifique pour chaque tâche : *MAE* et Kendall<sup>2</sup> pour la tâche de prédiction *WER* et le taux de bonne classification (*ac-*

2. Corrélacion entre les vraies valeurs *WER* (référence) et les valeurs *WER* prédites

Modèles	Tâche de prédiction de performance	
	MAE	Kendall
<b>Baseline : Mono-tâche</b>		
<b>WER</b>	15,24  19,24	45,00  46,83
<b>2-tâches</b>		
<b>WER EMISSION</b>	<b>14,83</b>   19,15	<b>47,25</b>   47,05
<b>WER STYLE</b>	15,07  19,66	45,92  45,49
<b>WER ACCENT</b>	15,05  19,60	46,17  45,60
<b>3-tâches</b>		
<b>WER STYLE ACCENT</b>	15,12  20,23	45,75  44,09
<b>WER EMISSION ACCENT</b>	14,94  19,76	46,19  43,61
<b>WER EMISSION STYLE</b>	14,90   <b>19,14</b>	45,87   <b>47,28</b>
<b>4-tâches</b>		
<b>WER EMISSION STYLE ACCENT</b>	15,15  19,64	45,59  45,42
<b>COMBINAISON TOUTES SORTIES</b>	<b>14,50</b>    <b>18,87</b>	<b>48,16</b>    <b>48,63</b>

TABLE 7.5 – Évaluation de la prédiction de performance du SRAP avec des modèles multi-tâche (*DEV*||*TEST*) en terme de *MAE* et Kendall - en termes de taux de bonne classification pour les tâches de classification secondaires

Modèles	Tâche de classification		
	EMISSION	STYLE	ACCENT
<b>2-tâches</b>			
<b>WER EMISSION</b>	<b>99,29</b>   -	-	-
<b>WER STYLE</b>	-	99,01  65,24	-
<b>WER ACCENT -</b>	-	91,72   75,30	-
<b>3-tâches</b>			
<b>WER STYLE ACCENT</b>	-	98,63  69,07	88,99   <b>77,46</b>
<b>WER EMISSION ACCENT</b>	98,38  -	-	89,87  71,44
<b>WER EMISSION STYLE</b>	99,12  -	<b>99,47</b>    <b>81,98</b>	-
<b>4-tâches</b>			
<b>WER EMISSION STYLE ACCENT</b>	99,04  -	99,29  81,55	<b>91,92</b>   73,60

TABLE 7.6 – Évaluation des tâches de classification secondaires des modèles multi-tâche (*DEV*||*TEST*) en termes de taux de bonne classification

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

---

*curacy*) pour les tâches de classification.

Les tableaux 7.5 et 7.6 résument les résultats expérimentaux sur les corpus DEV et TEST séparés par deux barres verticales (||). Nous avons considéré le modèle mono-tâche décrit dans la section 5.2 comme un système de référence.

Nous rappelons que nous avons évalué la tâche de classification EMISSION uniquement sur l'ensemble DEV (les émissions du corpus TEST n'existent pas dans notre TRAIN).

Tout d'abord, nous constatons que la performance des tâches de classification dans les scénarios multi-tâches est très bonne : nous sommes capables de former des systèmes efficaces de prédiction de performance SRAP qui annotent simultanément les tours de parole analysés en fonction de leur style de parole, leur accent et de l'origine du programme de diffusion. De tels systèmes multi-tâche pourraient être utilisés comme outils de diagnostic pour analyser et prédire les WER sur de grandes collections acoustiques.

De plus, nos meilleurs systèmes multi-tâche montrent une meilleure performance (MAE, Kendall) par rapport au système de base. Cela signifie que le fait de donner implicitement les informations sur le style, l'accent et le type d'émission peut être utile pour structurer l'apprentissage du système de prédiction.

Par exemple, pour les systèmes à deux tâches, le meilleur modèle est obtenu sur les tâches WER+EMISSION avec une différence respective de +0,41% et +2,25% en termes de MAE et Kendall (sur le corpus DEV) par rapport au système de base sur la tâche de prédiction WER.

Il faut cependant noter que l'impact de l'apprentissage multi-tâche sur la tâche principale (prédiction de la performance) est limité : des légères améliorations sur le corpus TEST sont observées en termes de MAE et Kendall. Néanmoins, les systèmes appris semblent complémentaires étant donné que leur combinaison (moyennage, sur l'ensemble des systèmes multi-tâche, du WER prédit au niveau des tours de parole) conduit à une amélioration significative des performances (voir dernière ligne du tableau pour MAE et Kendall).

### 7.6 Conclusion

Dans ce chapitre, nous avons essayé de comprendre ce qu'apprend le système CNN en analysant les représentations intermédiaires produites par notre meilleur

## CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR LE SYSTÈME DE PRÉDICTION NEURONAL

---

système de prédiction ( $\text{CNN}_{\text{Softmax}} \text{ EMBED+RAW-SIG}$ ). Afin de comprendre quelles sont les informations capturées par le modèle au cours de l’entraînement, nous avons suivi une méthode d’analyse inspirée de [Belinkov and Glass \[2017\]](#). L’idée est d’utiliser les représentations apprises pour des tâches de classification annexes (ou de les visualiser). Nos expérimentations montrent que notre modèle capture des informations sur le style de parole, l’accent du locuteur et le type d’émission durant l’apprentissage du système. Enfin, nous avons étudié le potentiel d’un apprentissage structuré consistant à donner ces trois informations au moment de l’entraînement du système de prédiction via un apprentissage multi-tâche. Les performances obtenues montrent que la création d’un système multi-tâche améliore légèrement la prédiction de WER tout en générant une prédiction correcte des informations additionnelles de type style de parole, accent du locuteur et type d’émission qui peuvent être des informations complémentaires utiles.

CHAPITRE 7. ÉVALUATION DES REPRÉSENTATIONS APPRISES PAR  
LE SYSTÈME DE PRÉDICTION NEURONAL

---

## Conclusion

Dans ce manuscrit de thèse, nous avons abordé la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole. C'est une nouvelle tâche qui consiste à prédire un taux d'erreur de mots au niveau d'un tour de parole ou au niveau d'un document lorsque les transcriptions références sont indisponibles et le système de RAP, à évaluer, inconnu.

Nous avons proposé dans un premier temps un protocole expérimental spécifique pour la tâche de prédiction de performances, où nous avons détaillé le scénario de prédiction envisagé. Afin d'apprendre et d'évaluer nos systèmes de prédiction, nous avons proposé un corpus hétérogène en français spécifique pour cette tâche. Pour obtenir des transcriptions automatiques à partir des signaux acoustiques, nous avons créé un système de transcription état de l'art en détaillant les différentes étapes de construction de ses modèles.

Nous avons proposé ensuite de comparer deux différentes approches de prédiction de performances : une approche basée sur des traits pré-définis (*engineered features*) en utilisant l'outil *TranscRater* [Jalalvand et al., 2016] et notre nouvelle approche basée sur des traits estimés au cours de l'apprentissage à l'aide des réseaux de neurones convolutifs (*learnt features*). Nos expérimentations ont montré que l'approche de prédiction par les CNNs est meilleure que l'approche de prédiction de base (par *TranscRater*) en termes de scores *MAE* et *Kendall*. Plus

précisément, l'utilisation conjointe en entrée des textes et signaux ne donne pas de résultats positifs pour les systèmes TR (TranscRater), tandis qu'elle permet de produire de meilleures performances en utilisant des CNNs. Nous avons montré également que les CNNs prédisent correctement la distribution des taux d'erreur de mots (WER) sur une collection d'enregistrements, contrairement à TranscRater qui prédit une distribution très éloignée de la réalité.

Nous avons proposé dans un second temps une analyse profonde des facteurs impactant nos systèmes PP tels que : la durée des tours de parole, les types d'émission, le style de parole, etc. Les résultats expérimentaux ont montré que la tâche de prédiction est plus difficile sur les tours de parole courts ainsi que sur les types d'émission ayant un style de parole spontané. Nous avons étudié également la robustesse des systèmes de prédiction lorsque le système est appris avec les sorties d'un système SRAP particulier et utilisé pour prédire la performance sur des nouveaux enregistrements (jamais rencontrés auparavant) transcrits avec des nouveaux systèmes de RAP (plus ou moins performants). Nous avons montré que les systèmes de prédiction de performances sont plutôt robustes, quelle que soit la qualité de sortie des SRAP au moment de l'apprentissage, il semble même qu'un système de RAP de qualité moyenne permet de rencontrer plus d'exemples d'erreurs au moment de l'apprentissage et donne, par conséquent des systèmes de prédiction légèrement plus performants.

Nous avons essayé enfin de comprendre ce qu'apprend le système CNN en analysant les représentations intermédiaires produites par notre meilleur système de prédiction ( $\text{CNN}_{\text{Softmax}} \text{ EMBED+RAW-SIG}$ ). Afin de comprendre quelles sont les informations capturées par le modèle au cours de l'entraînement, nous avons suivi une méthode d'analyse proposée par [Belinkov and Glass \[2017\]](#). L'idée a été d'utiliser les représentations apprises pour des tâches de classification annexes (ou de les visualiser). Nos expérimentations ont montré que notre modèle capture des informations sur le style de parole, l'accent du locuteur et le type d'émission durant l'apprentissage du système. Nous avons par la suite, étudié le potentiel d'un apprentissage structuré consistant à donner implicitement ces trois informations au moment de l'entraînement du système de prédiction via un apprentissage multi-tâche. Les performances obtenues ont montré que la création d'un système multi-tâche améliore légèrement la prédiction de WER tout en générant une prédiction correcte d'informations additionnelles telles que le style de parole, l'accent du

locuteur et le type d'émission qui peuvent être des informations complémentaires utiles.

## Perspectives

À partir des contributions présentées dans ce manuscrit, diverses perspectives peuvent être envisagées :

- **À court terme** : nous prévoyons dans un premier temps, d'intégrer des informations additionnelles à l'entrée des réseaux de neurones convolutifs telles que des *embeddings* : de lemmes, d'étiquettes morphosyntaxiques, de phonèmes, etc. De plus, nous proposons d'exploiter les représentations apprises (au niveau des tours de parole) par notre meilleur système de prédiction CNN afin de prédire des performances à une granularité plus large qu'un tour de parole comme : le type d'émission, l'instance de type d'émission, le style de parole, le locuteur, etc. En outre, nous proposons d'exploiter les informations (style de parole, accent du locuteur, type d'émission) disponibles au moment de l'apprentissage afin d'apprendre des représentations à l'aide des réseaux de neurones siamois [Koch et al., 2015]. Ces représentations seront intégrées par la suite, dans les couches de haut-niveau de notre réseau CNN afin d'apprendre un système de prédiction plus performant qui prend en compte les trois informations capturées. Étant donné que le genre du locuteur influence sur la qualité des systèmes de reconnaissance automatique de la parole [Mendoza et al., 1996], nous proposons d'étudier l'influence du genre du locuteur sur la qualité des systèmes de prédiction de performances.
- **À long terme** : nous pourrions envisager de contribuer à l'organisation d'une première campagne d'évaluation sur la tâche de prédiction de performances des systèmes de reconnaissance automatique de la parole. Les participants pourraient utiliser les mêmes corpus que nous avons exploités dans nos expériences. Cela peut booster les travaux sur cette tâche et nous permettre aussi de comparer nos systèmes de prédiction à d'autres systèmes de la campagne en utilisant les mêmes données d'apprentissage et d'évaluation ainsi que le même scénario de prédiction.

De même, nous envisageons d'étudier la distance entre les dialectes (tels que : le tunisien, l'égyptien, le levantin, etc.) de la langue arabe afin de

générer des représentations spécifiques et les intégrer dans notre système de prédiction CNN appris sur des données arabes. Ainsi, l'objectif sera d'évaluer l'impact de ces informations au moment de la prédiction de performances, de minimiser le coût de développement des systèmes de RAP appris sur un dialecte (comme le tunisien) et/ou de l'adapter à un nouveau dialecte (comme le marocain).

## BIBLIOGRAPHIE PERSONNELLE

- **Z. Elloumi**, L. Besacier, O. Galibert, and B. Lecouteux. Analyzing learned re- presentations of a deep ASR performance prediction model. In Black Box NLP Workshop (at EMNLP), 2018.
- **Z. Elloumi**, L. Besacier, O. Galibert, and B. Lecouteux. Prédiction de performances des systèmes de reconnaissance automatique de la parole à l'aide de réseaux de neurones convolutifs. In TAL-59-2, 2018.
- **Z. Elloumi**, L. Besacier, O. Galibert, J. Kahn, and B. Lecouteux. ASR performance prediction on unseen broadcast programs using convolutional neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP, Calgary-Canada, 2018.
- K. Bouzidi, **Z. Elloumi**, L. Besacier, B. Lecouteux, and M. F. Benzeghiba. Traitement des mots hors vocabulaire pour la traduction automatique de document Ocrisés en arabe. In TALN, 2017.
- C. Servan, A. Bérard, **Z. Elloumi**, H. Blanchon, and L. Besacier. Word2vec vs dbnary : Augmenting meteor using vector representations or lexical resources ? In Coling, 2016.
- **Z. Elloumi**, H. Blanchon, G. Serasset, and L. Besacier. Meteor for multiple target languages using dbnary. In MT Summit, 2015.
- **Z. Elloumi**, L. Besacier, and O. Kraif. Integrating multi word expressions in statistical machine translation. Multi-word units in machine translation and translation technologies, In MUMTTT Workshop, page 83, 2015.



## BIBLIOGRAPHIE

- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow : Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. Openfst : A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer, 2007.
- T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 1137–1140. IEEE, 1996.
- A. Asadi, R. Schwartz, and J. Makhoul. Automatic detection of new words in a large vocabulary continuous speech recognition system. *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 1990.
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv :1409.0473*, 2014.

## BIBLIOGRAPHIE

---

- L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, 3 :1–8, 1972.
- F. Béchet. Lia phon : un systeme complet de phonétisation de textes. *Traitement automatique des langues*, 42(1) :47–67, 2001.
- Y. Belinkov and J. Glass. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, pages 2438–2448, 2017.
- Y. Belinkov, L. Màrquez, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, volume 1, pages 1–10, 2017.
- J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb) :281–305, 2012.
- J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics, 2004.
- P. Breheny. Classification and regression trees. 1984.
- S. Brunessaux, P. Giroux, B. Grilheres, M. Manta, M. Bodin, K. Choukri, O. Galibert, and J. Kahn. The maurdor project : improving automatic processing of digital documents. In *2014 11th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 349–354. IEEE, 2014.
- F. Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015.
- R. Collobert and S. Bengio. Links between perceptrons, mlps and svms. In *Proceedings of the twenty-first international conference on Machine learning*, page 23. ACM, 2004.

- R. Collobert and J. Weston. A unified architecture for natural language processing : Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.
- R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(8) :2493–2537, 2011.
- J. Crego, J. Kim, G. Klein, A. Rebollo, K. Yang, J. Senellart, E. Akhanov, P. Brunelle, A. Coquard, Y. Deng, et al. Systran’s pure neural machine translation systems. *arXiv preprint arXiv :1610.05540*, 2016.
- G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1) :30–42, 2012.
- W. Dai, C. Dai, S. Qu, J. Li, and S. Das. Very deep convolutional neural networks for raw waveforms. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 421–425. IEEE, 2017.
- S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pages 65–74. Elsevier, 1990.
- M. De Calmès and G. Pérennou. Bdlx : a lexicon for spoken and written french. In *Proceedings of 1st International Conference on Langage Resources & Evaluation*, pages 1129–1136, 1998.
- J. G. de Souza, H. Zamani, M. Negri, M. Turchi, and F. Daniele. Multitask learning for adaptive quality estimation of automatically transcribed utterances. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 714–724, 2015.
- J. G. C. de Souza, C. Buck, M. Turchi, and M. Negri. Fbk-uedin participation to the wmt13 quality estimation shared task. In *Proceedings of the eighth workshop on statistical machine translation*, pages 352–358, 2013.

## BIBLIOGRAPHIE

---

- V. V. Digalakis and L. G. Neumeyer. Speaker adaptation using combined transformation and bayesian methods. *IEEE transactions on speech and audio processing*, 4(4) :294–300, 1996.
- W. A. Dreschler, H. Verschuure, C. Ludvigsen, and S. Westermann. Icra noises : artificial noise signals with speech-like spectral and temporal properties for hearing instrument assessment : Ruidos icra : Señates de ruido artificial con espectro similar al habla y propiedades temporales para pruebas de instrumentos auditivos. *Audiology*, 40(3) :148–157, 2001.
- J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul) :2121–2159, 2011.
- F. Eyben, M. Wöllmer, and B. Schuller. Opensmile : The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1459–1462, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi : 10.1145/1873951.1874246. URL <http://doi.acm.org/10.1145/1873951.1874246>.
- Y. Freund, R. E. Schapire, et al. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Citeseer, 1996.
- Y. Fu and L. Du. Combination of multiple predictors to improve confidence measure based on local posterior probabilities. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 1, pages I–93. IEEE, 2005.
- K. Fukushima. Neocognitron : a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4) :193–202, 1980.
- O. Galibert. Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. In F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, editors, *INTERSPEECH*, pages 1131–1134. ISCA, 2013. URL <http://dblp.uni-trier.de/db/conf/interspeech/interspeech2013.html#Galibert13a>.

- 
- O. Galibert, S. Rosset, C. Grouin, P. Zweigenbaum, and L. Quintard. Structured and extended named entity evaluation in automatic speech transcriptions. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 518–526, 2011.
- S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Interspeech*, pages 1149–1152, 2005.
- J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2) :291–298, 1994.
- P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1) :3–42, 2006.
- I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4) :237–264, 1953.
- I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- 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, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, pages 6645–6649. IEEE, 2013.
- G. Gravier, G. Adda, N. Paulson, M. Carré, A. Giraudel, and O. Galibert. The etape corpus for the evaluation of speech-based tv content processing in the french language. In *LREC-Eighth international conference on Language Resources and Evaluation*, page na, 2012.
- T. J. Hazen, S. Seneff, and J. Polifroni. Recognition confidence scoring and its use in speech understanding systems. *Computer Speech & Language*, 16(1) :49–67, 2002.

## BIBLIOGRAPHIE

---

- H. Hermansky and L. A. Cox Jr. Perceptual linear predictive (plp) analysis-synthesis technique. In *Second European Conference on Speech Communication and Technology*, 1991.
- H. Hermansky, E. Variani, and V. Peddinti. Mean temporal distance : Predicting asr error from temporal properties of speech signal. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7423–7426. IEEE, 2013.
- G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal processing magazine*, 29(6) :82–97, 2012.
- D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1) : 106–154, 1962.
- S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*, 2015.
- S. Jalalvand, D. Falavigna, M. Matassoni, P. Svaizer, and M. Omologo. Boosted acoustic model learning and hypotheses rescoring on the chime-3 task. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 409–415. IEEE, 2015a.
- S. Jalalvand, M. Negri, F. Daniele, and M. Turchi. Driving rover with segment-based asr quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, volume 1, pages 1095–1105, 2015b.
- S. Jalalvand, M. Negri, D. Falavigna, and M. Turchi. Driving rover with segment-based asr quality estimation, 01 2015c.
- S. Jalalvand, M. Negri, M. Turchi, J. G. de Souza, D. Falavigna, and M. R. Qwaidar. Transcrater : a tool for automatic speech recognition quality estimation.

- 
- Proceedings of ACL-2016 System Demonstrations. Berlin, Germany : Association for Computational Linguistics*, pages 43–48, 2016.
- F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4) :532–556, 1976.
- F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. Perplexity - measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1) :S63–S63, 1977.
- J. Kahn, O. Galibert, L. Quintard, M. Carré, A. Giraudel, and P. Joly. A presentation of the repere challenge. In *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, pages 1–6. IEEE, 2012.
- Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv :1408.5882*, 2014.
- D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *CoRR*, 6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *icassp*, volume 1, page 181e4, 1995.
- G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. 2001.
- S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273, 2015.
- B. Lecouteux, G. Linares, and B. Favre. Combined low level and high level features for out-of-vocabulary word detection. In *Interspeech 2009*, 2009.

## BIBLIOGRAPHIE

---

- Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.
- C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer speech & language*, 9(2) :171–185, 1995.
- L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11) :2579–2605, 2008.
- J. D. Markel and A. J. Gray. *Linear prediction of speech*, volume 12. Springer Science & Business Media, 2013.
- A. F. Martin and C. S. Greenberg. Nist 2008 speaker recognition evaluation : Performance across telephone and room microphone channels. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.
- J. Mauclair. *Mesures de confiance en traitement automatique de la parole et applications*. PhD thesis, Ph. D. thesis, LIUM, Le Mans, France, 2006.
- W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, 1943.
- B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. *librosa : Audio and music signal analysis in python*. 2015.
- N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(4) :417–473, 2010.
- E. Mendoza, N. Valencia, J. Muñoz, and H. Trujillo. Differences in voice quality between men and women : use of the long-term average spectrum (ltas). *Journal of Voice*, 10(1) :59–66, 1996.
- B. T. Meyer, S. H. Mallidi, H. Kayser, and H. Hermansky. Predicting error rates for unknown data in automatic speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5330–5334. IEEE, 2017.

- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013a.
- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.
- A.-r. Mohamed, G. Hinton, and G. Penn. Understanding how deep belief networks perform acoustic modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4273–4276. IEEE, 2012.
- P. J. Moreno, B. Logan, and B. Raj. A boosting approach for confidence scoring. In *Seventh European Conference on Speech Communication and Technology*, 2001.
- M. Negri, M. Turchi, J. G. de Souza, and D. Falavigna. Quality estimation for automatic speech recognition. In *COLING*, pages 1813–1823, 2014.
- D. Palaz, M. M. Doss, and R. Collobert. Convolutional neural networks-based continuous speech recognition using raw speech signal. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4295–4299. IEEE, 2015.
- J. Pennington, R. Socher, and C. Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- G. Perennou and M. d. Calmes. Bdex lexical data and knowledge base of spoken and written french. In *European conference on Speech Technology*, 1987.
- D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, et al. The subspace gaussian mixture model structured model for speech recognition. *Computer Speech & Language*, 25(2) : 404–439, 2011a.
- D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hanemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011b.

## BIBLIOGRAPHIE

---

- C. Quirk. Training a sentence-level machine translation confidence measure. In *LREC*. Citeseer, 2004.
- L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.
- F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386, 1958.
- R. San-Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. M. Pardo. Confidence measures for spoken dialogue systems. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 393–396. IEEE, 2001.
- R. Sarikaya, Y. Gao, M. Picheny, and H. Erdogan. Semantic confidence measurement for spoken dialog systems. *IEEE Transactions on Speech and Audio Processing*, 13(4) :534–545, 2005.
- H. Schmid. Treetagger| a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43 :28, 1995.
- F. Sébastien, F. Jérôme, P. Julien, and R. Stéphane. Prédiction a priori de la qualité de la transcription automatique de la parole bruitée. In *Proc. XXXIIIe Journées d'Études sur la Parole*, pages 249–257, 2018. doi : 10.21437/JEP.2018-29. URL <http://dx.doi.org/10.21437/JEP.2018-29>.
- 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.
- X. Shi, I. Padhi, and K. Knight. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, 2016.
- 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.

- G. Stemmer, S. Steidl, E. Nöth, H. Niemann, and A. Batliner. Comparison and combination of confidence measures. In *International Conference on Text, Speech and Dialogue*, pages 181–188. Springer, 2002.
- A. Stolcke. Srilm - an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, USA, 2002.
- A. Stolcke et al. Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002, 2002.
- I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1) :9–40, 2007.
- V. Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.
- S. Wang, Y. Qian, and K. Yu. What does the speaker embedding encode? In *Interspeech*, volume 2017, pages 1497–1501, 2017.
- P. Wiggers and L. J. Rothkrantz. Using confidence measures and domain knowledge to improve speech recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.
- C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1) :79–82, 2005.
- I. H. Witten and T. C. Bell. The zero-frequency problem : Estimating the probabilities of novel events in adaptive text compression. *Ieee transactions on information theory*, 37(4) :1085–1094, 1991.

## BIBLIOGRAPHIE

---

- Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*, 2016.
- Z. Wu and S. King. Investigating gated recurrent neural networks for speech synthesis. *CoRR*, abs/1601.02539, 2016. URL <http://arxiv.org/abs/1601.02539>.
- S. R. Young. Recognition confidence measures : Detection of misrecognitions and out-of-vocabulary words. *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 1 :21–24, 1994.
- M. D. Zeiler. ADADELTA : an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL <http://arxiv.org/abs/1212.5701>.
- R. Zhang and A. I. Rudnicky. Word level confidence annotation using combinations of features. In *Seventh European Conference on Speech Communication and Technology*, 2001.

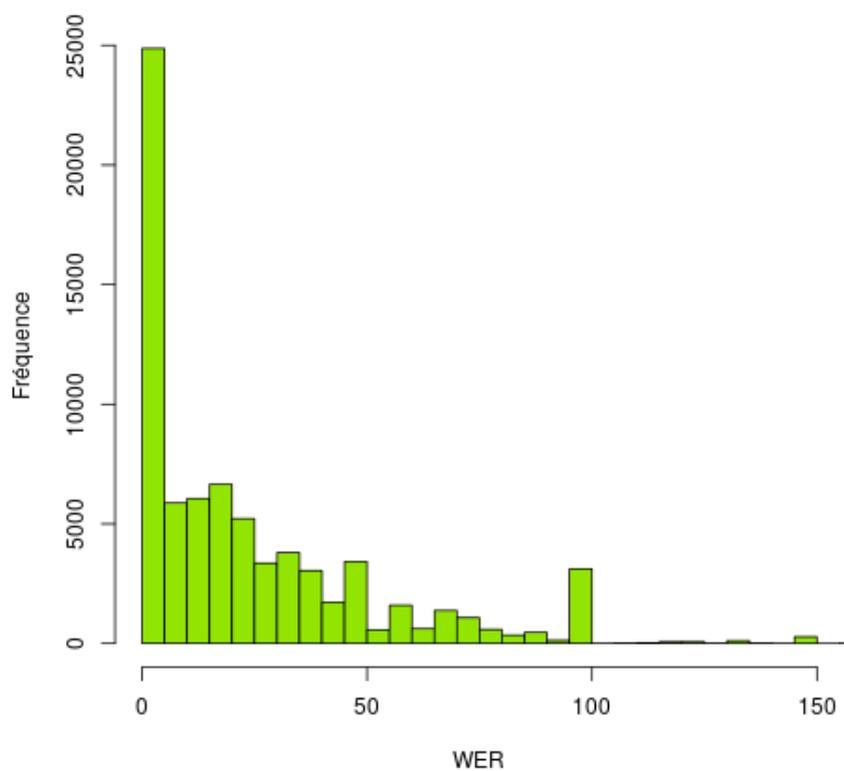


FIGURE A.1 – Distribution des tours de parole du corpus  $Test_{p^{red}}$  en fonction de leurs WER

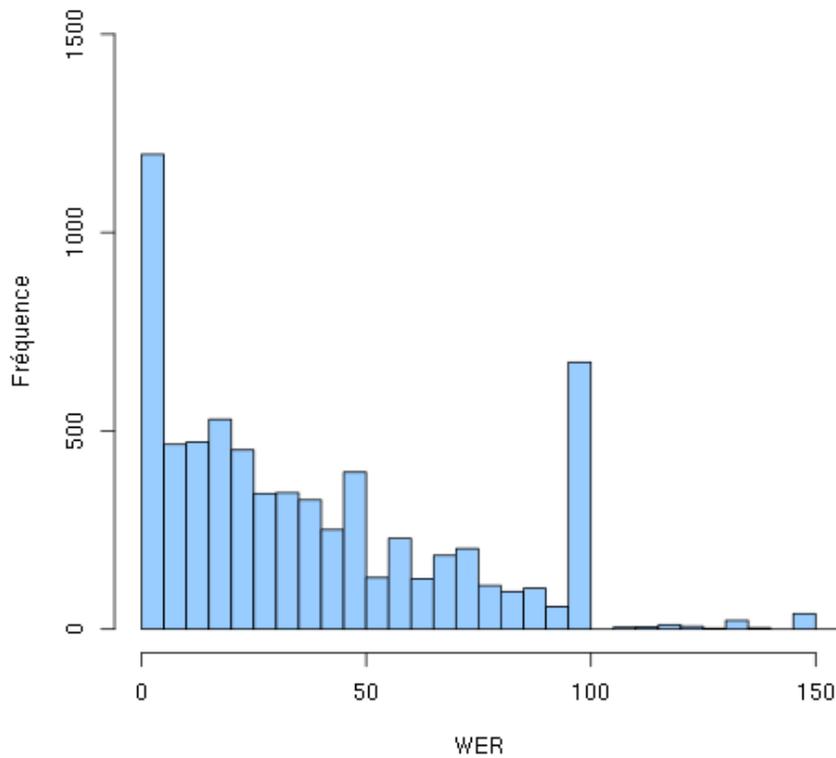


FIGURE A.2 – Distribution des tours de parole du corpus TEST en fonction de leurs WER

Traits	TEST
LM	<b>24,19</b>
POS	25,95
LEX	25,78
SIG	25,86
LM_POS_LEX	22,01
LM_POS_LEX_SIG	<b>21,99</b>

TABLE A.1 – Évaluation des modèles TranscRater sur le corpus de TEST en termes de MAE

Cross-validation	DEV	TEST
1	15.35	20.02
<b>2</b>	<b>15.24</b>	<b>19.24</b>
3	15.64	20.5
4	15.42	19.75
5	15.37	20.12
6	15.53	19.71
7	15.27	20.05
8	15.83	19.67
9	15.26	19.73
10	15.56	19.71

TABLE A.2 – Évaluation des 10 modèles de prédiction  $\text{CNN}_{softmax}$  EMBED+RAW-SIG sur les corpus DEV et TEST en termes de MAE

Transcription	
Référence	dans ce contexte il a été décidé le rappel de l' ambassadeur du royaume au sénégal pour une période de trois jours
Hypothèse	dans ce contexte il a été décidé le rappel de l' ambassadeur du royaume au sénégal pour une période de trois jours
Taux d'erreur de mots	
Référence	0
TranscRater	10,58
$\text{CNN}_{softmax}$	0

TABLE A.3 – Exemple de transcription automatique (obtenue par  $\text{SRAP}_1$ ) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et  $\text{CNN}_{softmax}$

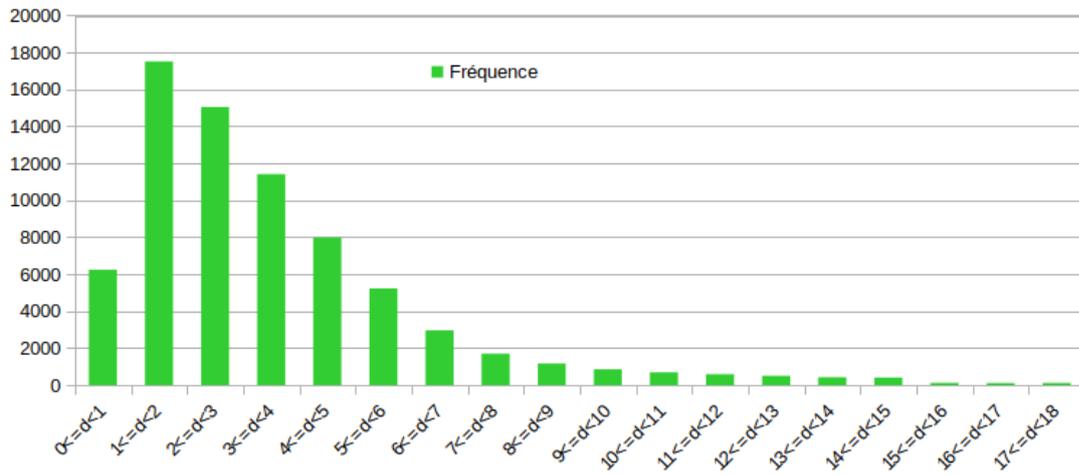


FIGURE A.3 – Distribution des tours de parole du corpus TRAIN en fonction de leurs durées (en seconde)

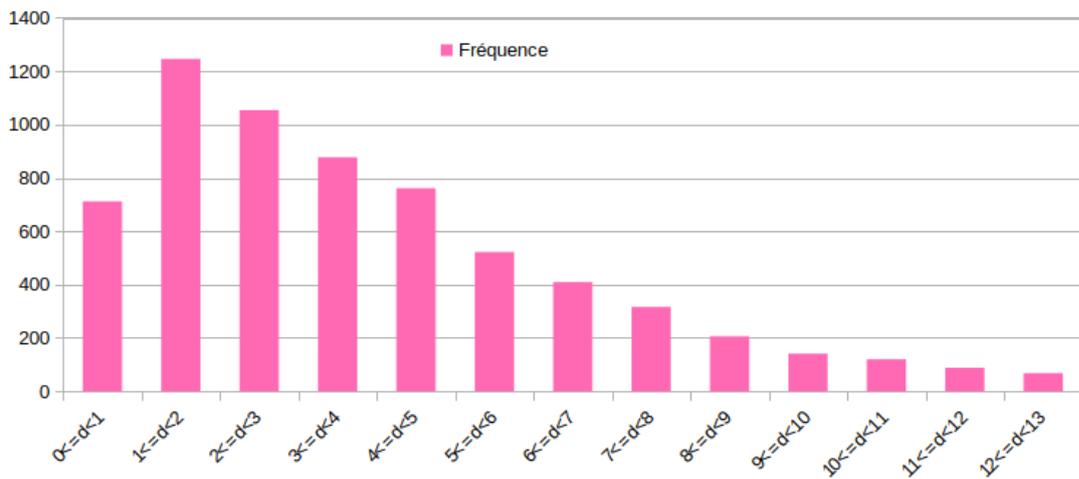


FIGURE A.4 – Distribution des tours de parole du corpus TEST en fonction de leurs durées (en seconde)

Transcription	
Référence	un mort et soixante trois blessés aujourd’hui lors d’affrontements à maputo au mozambique entre des manifestants protestant contre une hausse
Hypothèse	un mort et soixante trois blessés aujourd’hui lors d’affrontements à à maputo mozambique entre des manifestants protestant contre une hausse
Taux d’erreur de mots	
Référence	9.52
TranscRater	5.77
CNN <sub>Softmax</sub>	14.30

TABLE A.4 – Exemple de transcription automatique (obtenue par SRAP<sub>1</sub>) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et CNN<sub>Softmax</sub>

Transcription	
Référence	c’ était tout d’ un coup d’ où sort elle qu’ est ce que c’ est un ministre de droite a dit voilà la pompadour
Hypothèse	c’ était tout d’ un coup d’ où sortait le qu’ est ce que c’ est un ministre de droite a dit voilà la pompadour
Taux d’erreur de mots	
Référence	8,00
TranscRater	21,29
CNN <sub>Softmax</sub>	10,63

TABLE A.5 – Exemple de transcription automatique (obtenue par SRAP<sub>1</sub>) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et CNN<sub>Softmax</sub>

Transcription	
Référence	ça continue bonne bon été en tout cas et merci encore à vous
Hypothèse	c' est contenue monnaie venait merci encore à vous
Taux d'erreur de mots	
Référence	69,23
TranscRater	45,30
CNN <sub>Softmax</sub>	67,66

TABLE A.6 – Exemple de transcription automatique (obtenue par SRAP<sub>1</sub>) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et CNN<sub>Softmax</sub>

Transcription	
Référence	mais pas n'importe lequel le lit où michael jackson a poussé son dernier soupir oui c' est un peu glauque il fera partie avec des centaines d' autres objets d' une vente aux enchères le dix sept décembre prochain objets qui se trouvaient dans le manoir où le roi de la pop a passé les derniers mois de sa vie
Hypothèse	dans ce cas elle le liban études jackson a poussé son dernier soupir c' est un peu glauque il fera partie avec des centaines d' autres objets d' une vente aux enchères le dix sept décembre prochain objectif se trouvait dans le manoir où le roi de la pop a passé les derniers mois de sa vie
Taux d'erreur de mots	
Référence	19.67
TranscRater	14.39
CNN <sub>Softmax</sub>	22.26

TABLE A.7 – Exemple de transcription automatique (obtenue par SRAP<sub>1</sub>) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et CNN<sub>Softmax</sub>

Transcription	
Référence	ça s' applique totalement à l' histoire du cinéma français qui joue à la fois sur la construction d' imaginaire et la construction d' un légendaire qui permet de regrouper la communauté autour d' une mythologie commune et cette mythologie joue aussi le role d' écran précisément laissant dans l' ombre certaines pages plus noires de la période de l' occupation
Hypothèse	ça s' explique totalement à l' histoire du cinéma français qui joue à la fois sur la construction d' imaginaire et la construction d' un légendaire qui permet de regroupés la communauté autour d' une mythologie commune et cette mythologie joue aussi le role d' écrans précisément hum laissant dans l' ombre certaines pages noires de la période de de l' occupation
Taux d'erreur de mots	
Référence	9,84
TranscRater	22,74
CNN <sub>Softmax</sub>	15,67

TABLE A.8 – Exemple de transcription automatique (obtenue par SRAP<sub>1</sub>) et de prédiction de WER en utilisant les meilleurs systèmes TranscRater et CNN<sub>Softmax</sub>

---

Référence	TranscRater	EMBED+RAW-SIG
0.0	15.04	29.15
0.0	9.75	0.0
15.38	18.04	23.55
0.0	18.93	0.0
0.0	60.13	0.89
3.85	17.70	24.86
48.0	32.47	4.84
10.0	39.32	28.21
9.78	15.56	16.03
87.5	32.83	36.32
0.0	24.97	38.59
32.35	18.98	41.27
26.32	29.42	45.22
90.0	64.49	74.09
100.0	55.45	100.0
14.29	12.86	9.25
12.5	19.11	22.88
26.67	33.84	14.64
0.0	15.50	9.42
150.0	52.87	100.03
57.14	50.17	39.44
63.33	38.08	68.78
0.0	23.33	0.0
25.0	32.80	27.25
4.55	27.41	7.63
0.0	22.85	0.0
33.33	26.37	40.42
33.33	28.43	29.44
150.0	45.92	100.0
19.67	14.39	22.26
88.89	29.40	78.38
15.38	23.78	20.43
75.0	28.63	54.84
21.43	19.04	20.18
41.18	14.41	32.47
33.33	42.31	37.21
10.0	28.45	2.36
100.0	64.61	99.99

---

TABLE A.9 – Exemples de WER prédits au niveau des tours de parole par les meilleurs systèmes de prédiction : TranscRater *vs* CNN<sub>Softmax</sub> EMBED + RAW-SIG

---

Référence	TranscRater	EMBED+RAW-SIG
11.11	26.56	11.29
17.65	15.16	12.04
0.0	18.84	0.0
27.27	18.43	12.97
0.0	12.74	5.32
22.73	25.63	29.06
9.09	11.66	6.41
0.0	45.08	0.0
56.25	20.22	14.71
20.0	34.30	33.73
39.13	20.29	27.07
30.77	26.11	25.28
133.33	53.52	95.06
66.67	37.26	51.77
80.0	49.52	37.75
100.0	45.24	100.0
0.0	18.58	25.98
0.0	12.60	0.0
53.33	25.70	15.65
100.0	37.35	92.89
18.18	31.30	27.98
94.44	36.28	38.83
56.52	37.03	51.41
84.21	33.00	21.83
15.38	22.10	28.99
10.53	10.95	0.0
44.0	27.16	31.33
7.14	25.40	13.43
81.82	27.08	32.97
64.29	31.78	98.1
25.86	19.55	15.64
9.52	23.35	22.86
61.9	19.23	62.94
57.14	43.35	37.44
100.0	56.71	100.0
35.71	29.17	26.39
0.0	41.73	43.87
0.0	6.27	0.0

TABLE A.10 – Exemples de WER prédits au niveau des tours de parole par les meilleurs systèmes de prédiction : TranscRater *vs* CNN<sub>Softmax</sub> EMBED+RAW-SIG