



RECONNAISSANCE AUTOMATIQUE DE LA PAROLE POUR L'APPRENTISSAGE DE LA LANGUE YEMBA DANS LES ÉCOLES PRIMAIRES

Mémoire de fin d'études

Présenté et soutenu par :

KANA AZEUKO SHERELLE

En vue de l'obtention du :

Diplôme d'Ingénieur de Conception, Génie Informatique

Année académique 2023-2024

Le xx Septembre 2024

DÉDICACE



REMERCIEMENTS

Je tiens à exprimer toute ma gratitude envers ceux qui ont contribué à mon éducation, à ma formation, et qui ont rendu possible la réalisation de ce projet. Leur soutien indéfectible a été essentiel à chaque étape de mon parcours. C'est grâce à eux que je peux aujourd'hui présenter ce travail avec fierté :

- ✿ **Pr. ,** pour l'honneur qu'il me fait en acceptant de présider ce jury.
- ✿ **Dr. TIOGNING DJIOGUE Lauraine**, pour l'honneur qu'elle me fait en acceptant d'examiner scientifiquement ce travail.
- ✿ **Dr. NGOUNOU Guy Merlin** mon encadreur académique qui a été mon guide attentif tout au long de ce stage, m'offrant disponibilité, conseils, remarques et une confiance inestimable. Sa rigueur exemplaire a été pour moi une source constante d'inspiration et d'apprentissage.
- ✿ **Dr MELATAGIA YONTA Paulin** mon co-encadreur et encadreur professionnel pour son suivi, ses conseils et son expertise technique et scientifique tout au long de notre travail.
- ✿ **UMMISCO** par le biais du Codirecteur du Centre Afrique Centrale et de l'Est, **Dr MELATAGIA YONTA Paulin**, qui nous a fourni un cadre de travail agréable pour la réalisation de ce projet.
- ✿ Le chef de département du Génie Informatique **Pr Thomas BOUETOU BOUETOU** et le corps enseignant de l'ÉNSPY, pour la formation et le savoir qu'ils se sont dévoués à me transmettre.
- ✿ Mes parents, **M. AZEUKO Georges** et **Mme KANA Edwige Michèle**, pour leur soutien indéfectible, leur éducation exemplaire et leurs conseils avisés qui ont toujours éclairé mon chemin avec amour. Leur présence constante et leurs multiples formes de soutien ont été pour moi une source inestimable de force et d'inspiration tout au long de la réalisation de ce mémoire. Sans oublier mes frères et sœurs pour leur chaleur fraternelle.

Enfin, un grand merci à tous ceux qui, de près ou de loin, m'ont apporté leurs soutiens moral, spirituel et intellectuel tout au long de ma démarche.

TABLE DES MATIÈRES

Dédicace	i
Remerciements	ii
Résumé	vi
Abstract	vii
Sigles et abréviations	viii
Glossaire	ix
Liste des tableaux	x
Table des figures	xi
Introduction Générale	1
1 Généralités et État de l'art	6
1.1 Qu'est ce que la parole?	7
1.1.1 Éléments fondamentaux de la parole	7
1.1.2 Phonétique	8
1.1.3 Langues tonales	8
1.2 Concepts de base du traitement de la parole	9
1.2.1 Définition du signal audio	9
1.2.2 Numérisation du signal audio	9
1.3 Reconnaissance automatique de la parole	11
1.3.1 Définition	11
1.3.2 Architecture de base d'un modèle de reconnaissance de la parole	12
1.3.3 Méthodologies de Reconnaissance Vocale	14
1.3.4 Types de Reconnaissance Vocale	14
1.3.5 Quelques applications de la reconnaissance vocale	15
1.3.6 Métriques d'évaluation des ASR	15

1.3.7	Exemples d'outils de reconnaissance automatique de la parole	16
1.4	Réseaux de neurones graphes GNN	20
1.4.1	Architecture d'un GNN	22
1.4.2	Variante des GNN	23
1.5	Présentation de la langue Yemba	25
1.6	Étude de l'existant	28
1.6.1	Architectures basées sur les caractéristiques acoustiques	28
1.6.2	Réseau neuronal convolutif pour la reconnaissance de la parole arabe . .	29
1.6.3	Reconnaissance des émotions de la parole basée sur le réseau neuronal Graph-LSTM	30
1.7	Bilan et positionnement	31
2	Méthodologie	32
2.1	Présentation de l'Architecture méthodologique	33
2.2	Compréhension du projet et collecte des données	35
2.2.1	Compréhension de l'apprentissage et l'évaluation des LCN	35
2.2.2	Formation du corpus	36
2.2.3	Collecte et préparation des données	37
2.2.4	Traduction du Corpus en Langue Yemba	38
2.2.5	Mise sur pied du protocole de collecte	38
2.2.6	Choix des Locuteurs	39
2.2.7	Collecte sur le terrain	40
2.2.8	Nettoyage et Étiquetage des données	41
2.3	Entraînement avec Kaldi	44
2.4	Mise en place du modèle GraphSAGE	45
2.4.1	Extraction des MFCC	46
2.4.2	Construction des graphes	46
2.4.3	Entraînement de GraphSAGE	46
2.4.4	Extraction des embeddings	46
2.4.5	Formatage des embeddings au format Kaldi	46
2.5	Description de l'évaluation	46
2.5.1	Évaluation avec des Métriques	46
2.5.2	Évaluation Empirique	46
3	Expérimentations et résultats	48
3.1	Présentation du jeu de données	49
3.2	Expérimentations avec Kaldi	49
3.2.1	Présentation de l'environnement matériel	49
3.2.2	Configuration expérimentale	50



TABLE OF CONTENTS



3.3	Expérimentations de l'architecture proposée	51
3.3.1	Présentation de l'environnement matériel	51
3.3.2	Configuration expérimentale	51
3.4	Résultat et interprétation	51
4	Conclusion Générale	52
4.1	Rappel du problème	52
4.2	Démarche et résultats	52
4.3	Limites	52
4.4	Perspectives	52
5	Annexe	53
5.1	Présentation de UMMISCO	53
5.2	Expérimentation avec Kaldi	53
5.2.1	Étape 1 : Configuration de l'environnement de travail	53
5.2.2	Étape 2 : Préparation des données	54
5.2.3	Étape 3 : Extraction des caractéristiques MFCC et calcul des statistiques CMVN	57
5.2.4	Étape 4 : Entraînement et alignement du modèle acoustique	58
5.2.5	Création du modèle de langage	58
5.2.6	Décodage et Évaluation	58
	Références bibliographiques	60



RÉSUMÉ

L

Mots clés :

ABSTRACT

T

Keywordss :

SIGLES ET ABRÉVIATIONS

ANACLAC Association Nationale des Comités des Langues Camerounaises

ASR Automatic Speech Recognition

API Application Programming Interface

CNN Convolutional Neural Networks

DL Deep Learning

ENSPY École Nationale Supérieure Polytechnique de Yaoundé

GNN Graph Neural Network

LCN Langue et Culture Nationale

HMM Modèles de Markov Cachés

ML Machine Learning

MFCC Mel-Frequency Cepstral Coefficients

MINEDUB Ministère de l'Éducation de Base

SIL Société Internationale de Linguistique

GLOSSAIRE

Un centre d'intérêt est un champ thématique autour duquel les apprentissages, à travers toutes les disciplines, doivent se faire sur une période d'un mois.

Une langue peu dotée, fait référence à une langue présentant certains (sinon tous) des aspects suivants : absence d'un système d'écriture unique ou d'une orthographe stable, présence limitée sur le web, manque d'expertise linguistique, manque de ressources électroniques pour le traitement de la parole et du langage, telles que des corpus monolingues, des dictionnaires électroniques bilingues, des données de parole transcrites, des dictionnaires de prononciation, des listes de vocabulaire, etc.

LISTE DES TABLEAUX

1.1	Tableau récapitulatif	31
2.1	Répartition des nombres mots par centre d'intérêts	37
2.2	Répartition des mots par centre d'intérêts	37
2.3	Tableau de quelques mots après traduction	38
2.4	Tableau statistique de l'échantillonnage des locuteurs	40
2.5	Détails de quelques locuteurs et des emplacements respectifs où leurs enregistre- ments ont eu lieu	40
2.6	Tableau statistique des locuteurs descente 1	41
2.7	Tableau statistique des locuteurs descente 2	41
3.1	Répartition des fichiers par session de collecte	49
3.2	Répartition des mots par centre d'intérêts	51

TABLE DES FIGURES

1.1	Les organes vocaux, présentés en vue latérale.[27]	7
1.2	La courbe audiométrique de l'oreille humaine [6]	10
1.3	Architecture de base d'un modèle de reconnaissance de la parole[16]	12
1.4	Une vue simplifiée des différents composants de Kaldi..[11]	16
1.5	Un aperçu d'un script de formation de base de SpeechBrain[11]	19
1.6	Composants des CNN [37]	20
1.7	Comparaison entre un CNN et un GNN	21
1.8	Architecture de base d'un GNN.[44]	22
1.9	Les régions où la langue Yemba est parlée et sa distribution géographique. [42]	26
1.10	Ton bas pour la lettre "a"[19]	27
1.11	Ton haut pour la lettre "a"[19]	27
1.12	Ton moyen pour la lettre "a"[19]	27
1.13	Architecture d'un Système de Reconnaissance Vocale[2]	28
1.14	Architecture CNN pour un ASR[12]	29
1.15	Architecture du GLNN[20]	30
2.1	Architecture méthodologique	33
2.2	Diagramme méthodologique : vue éclatée du bloc bleu	35
2.3	Illustration d'un fichier audio sur Audacity	42
2.4	Illustration de l'étiquetage d'un fichier audio avec Audacity	43
2.5	Diagramme méthodologique : vue éclatée du bloc rouge	44
2.6	Diagramme méthodologique : vue éclatée du bloc vert	45
5.1	Arborescence dossier data : préparation des fichiers audio	54
5.2	Arborescence dossier data : préparation des dossiers et fichiers de langage	56
5.3	Arborescence dossier data : extraction des caractéristiques	57
5.4	Arborescence dossier exp	57

INTRODUCTION GÉNÉRALE

CONTEXTE

Dans les années qui ont suivi l'accession à l'indépendance des pays africains, une préoccupation majeure a émergé : comment préserver l'identité culturelle des peuples africains face à l'héritage colonial qui risquait de les acculturer? Cette question, d'une importance capitale, s'est rapidement concentrée autour des axes clés parmi lesquels la sauvegarde des langues maternelles, d'autant plus lorsque celles-ci sont peu dotées en ressources et menacées de disparition. Le Yemba, langue parlée dans la région de l'Ouest Cameroun, en est une. Cependant, comme de nombreuses langues africaines, le Yemba est un pilier de l'identité culturelle et de la transmission intergénérationnelle de connaissances. Sa disparition pourrait avoir des conséquences néfastes pour la richesse culturelle et linguistique du Cameroun. Face à cette réalité, des efforts plus concertés ont été déployés, notamment à travers des initiatives internationales telles que celles de l'UNESCO et de l'Union africaine. Cette loi a été un premier pas crucial dans la reconnaissance et la préservation des langues locales.

Cependant, la route vers la préservation des langues maternelles était semée d'embûches. La mondialisation croissante et la prédominance des langues internationales menaçaient davantage la diversité linguistique et culturelle. Face à cette réalité, des efforts plus accrus ont vu le jour. En 1953, la Conférence de Yaoundé sur l'enseignement en Afrique centrale, organisée par l'UNESCO, a marqué le début d'une prise de conscience internationale de l'importance de l'enseignement des langues maternelles en Afrique [29]. Cette rencontre historique a posé les bases d'un mouvement qui s'est amplifié au fil des décennies. En 1998, conscient de cette menace imminente, le Cameroun a posé les premières pierres en adoptant une loi d'orientation sur l'éducation, encourageant explicitement l'enseignement des langues maternelles dans le système éducatif [38]. En 2019, le Cameroun a franchi une nouvelle étape dans cette entreprise de préservation linguistique en adoptant la Politique nationale de promotion et de développement des langues maternelles. Cette politique ambitieuse vise à encourager l'utilisation des langues maternelles dans tous les domaines de la vie, y compris l'éducation [30].

Dans un monde de plus en plus connecté, les avancées technologiques offrent des opportunités pour la préservation et la promotion des langues maternelles. La reconnaissance de la



parole en est un exemple. Grâce aux progrès rapides dans le domaine de l'intelligence artificielle et du traitement automatique du langage naturel, il est désormais possible de développer des outils d'ASR¹ qui sont des implémentations existantes de l'architecture de base des ASR, adaptés à des langues jusqu'ici sous-représentées sur la scène technologique mondiale. Ce travail tire parti de ces avancées technologiques pour le cas de la langue Yemba. C'est dans ce contexte de préservation et de promotion des langues maternelles que s'inscrit notre projet de reconnaissance de la parole pour l'apprentissage de la langue Yemba à l'école primaire.

1. Automatic Speech Recognition





PROBLÉMATIQUE

Les défis que nous rencontrons avec la langue Yemba incluent plusieurs limitations pour sa reconnaissance automatique. Tout d'abord, il y a la non existence des ressources linguistiques numériques disponibles, ce qui complique la création de corpus de données de haute qualité nécessaires pour entraîner les modèles de reconnaissance vocale. De plus, il est crucial de développer des outils adaptés aux contextes d'apprentissage spécifiques, comme celui de l'école primaire, où les variations linguistiques et les besoins pédagogiques doivent être pris en compte. Aussi la plupart des outils de reconnaissance automatique de la parole (ASR) sont développés et testés principalement sur des langues largement dotées en ressources, telles que l'anglais, le français ou l'espagnol. L'adaptation de ces technologies pour les faire fonctionner efficacement avec des langues ayant peu de ressources apparaît comme un défi supplémentaire. De se fait ces outils ne sont pas optimisés pour des langues moins dotées comme le Yemba, ce qui peut entraîner une diminution significative de la précision de ceux-ci. Cependant, nous estimons pouvoir surmonter ce défi en tirant parti de la capacité de représentation des données qu'ont les modèles d'apprentissage profond, tels que les GNN.

QUESTION DE RECHERCHE

Etant donné que la langue Yemba est une langue peu dotée en ressource, il en ressort la question suivante :

Question

Comment pallier l'absence de ressources pour la langue Yemba dans le cadre de la reconnaissance automatique de la parole pour son apprentissage à l'école primaire?

HYPOTHÈSES DE TRAVAIL

Nous posons comme hypothèse qu'un outil classique d'ASR sur lequel on a intégré un GNN peut améliorer les prédictions de celui-ci et ainsi apporter une solution à notre problématique.

- Nous pensons que les outils d'ASR existant s ne sont pas optimisés pour les langues peu dotées en ressources.
- Nous pensons que coupler un GNN á un outil d'ASR permettra d'améliorer ses performances.





OBJECTIFS

Notre travail s'inscrit dans le cadre d'une initiative sociétale visant à pérenniser et sauvegarder les langues camerounaises, en particulier la langue Yemba, en utilisant Kaldi, un outil de reconnaissance automatique de la parole (ASR)² appliqué à l'apprentissage de celle-ci à l'école primaire. Pour mener à bien ce projet nous nous sommes fixés les objectifs spécifiques suivants :

- Elaborer un protocole de collecte de données et collecter ces données pour fournir une base nécessaire au développement de l'outil ASR.
- Utiliser les données recueillies pour configurer et entraîner un outil de reconnaissance automatique de la parole.
- Améliorer les prédictions de cet outil sur nos données en Yemba grâce aux réseaux de neurones graphes (GNNs).

MÉTHODOLOGIE

Afin de mener à bien ce travail et atteindre les objectifs fixés, nous commencerons par comprendre l'apprentissage des LCN³ ainsi que leur évaluation. Ensuite, nous procéderons à la formation du corpus, que nous traduirons en langue Yemba. Une fois le corpus établi, nous mettrons en place un protocole de collecte et sélectionnerons les locuteurs appropriés. La collecte des données sur le terrain sera suivie d'une phase de nettoyage et d'étiquetage des données pour les préparer à l'utilisation avec l'outil Kaldi. Nous extrairons ensuite les caractéristiques MFCC avec Kaldi et entraînerons le modèle acoustique $P(O|W)$ ainsi que le modèle de langage $P(W)$. Après le décodage $P(W|O)$, nous construirons les graphes nécessaires et développerons l'architecture GraphSAGE. Cette architecture sera ensuite entraînée, et les embeddings seront extraits et formatés pour être compatibles avec Kaldi. Enfin, nous procéderons à l'évaluation, à la production des résultats, et à leur interprétation.

PLAN DU MÉMOIRE

La suite de ce mémoire comprends : trois chapitres principaux, suivis d'une conclusion générale :


- ✎ **Chapitre 1 : Généralités et État de l'art** Dans ce chapitre, nous allons présenter et définir les concepts pertinents pour la compréhension de notre projet. Ensuite, nous réaliserons une étude des travaux existants, en exposant les avantages et les limites des solutions actuelles. Enfin, nous conclurons par un bilan et une présentation de notre positionnement par rapport à ces solutions.


2. Automatic Speech Recognition

3. Langues et Culture Nationale





 **Chapitre 2 : Méthodologie** Dans ce chapitre nous présentons notre contribution à la résolution de ce problème. Nous y détaillons les approches méthodologiques adoptées pour mettre en œuvre : la collecte de données, la configuration des outils ASR et l'amélioration de ces outils.

 **Chapitre 3 : Expérimentation et Résultats** Dans ce chapitre, nous décrivons les outils et technologies utilisés pour développer notre solution. Nous évaluons l'approche mise en œuvre, présentons le système développé et illustrons son fonctionnement à l'aide de scénarios d'utilisation concrets. Nous discutons également des résultats obtenus.

Nous terminons par une **conclusion générale**, qui présente le bilan de ce travail, dégage ses limites et quelques perspectives.



GÉNÉRALITÉS ET ÉTAT DE L'ART

Dans ce chapitre, nous commencerons par introduire divers éléments et concepts théoriques pour mieux comprendre notre sujet. Ensuite, nous réaliserons une étude de l'existant, décrivant quelques architectures d'ASR de la littérature, avec leurs avantages et leurs inconvénients. Enfin, nous conclurons par un positionnement par rapport à ces éléments.

1.1 Qu'est ce que la parole ?

1.1.1 Éléments fondamentaux de la parole

La parole est le langage articulé humain destiné à communiquer la pensée. La parole est à distinguer des communications orales diverses comme les cris, les alertes ou les gémissements. « Articuler la parole » consiste à former des signes audibles, les syllabes, formant les mots qui constituent des symboles [41]. Elle se réfère à la façon de produire et percevoir les consonnes et les voyelles de toutes les langues du monde [35] [27].

Les éléments de la parole sont :

- La Voix : Utilisation des cordes vocales et de la respiration pour produire des sons, avec variations d'intensité et de hauteur.
- L' Articulation : Usage des lèvres et de la langue pour former les sons spécifiques de la parole.
- La Résonance : Modification des sons par les cavités du pharynx, de la bouche et du nez, influençant la qualité sonore.
- La Fluidité : Rythme de la parole, affecté par les hésitations et répétitions, lié au bégaiement.
- La Perception : Capacité à détecter et interpréter les variations subtiles du signal acoustique de la parole.

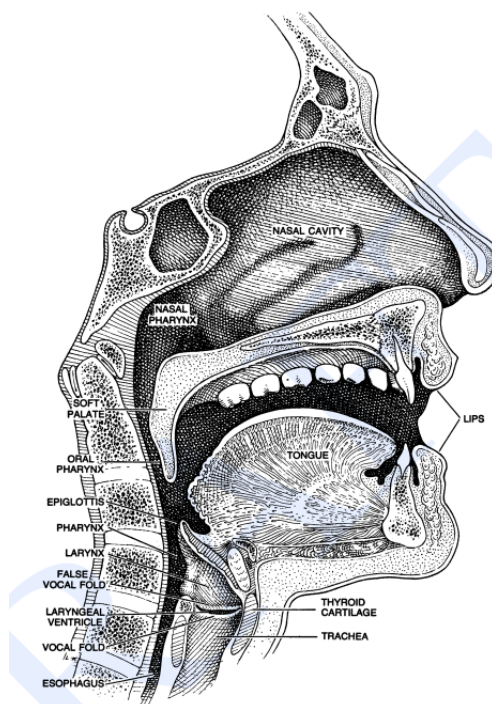


FIGURE 1.1 – Les organes vocaux, présentés en vue latérale.[27]



1.1.2 Phonétique

La phonétique est l'étude des sons linguistiques, de leur production par les articulateurs du tractus vocal humain, de leur réalisation acoustique et de la façon dont cette réalisation acoustique peut être numérisée et traitée [27]. L'étude de la prononciation des mots relève de la phonétique, qui est la discipline consacrée à l'analyse des sons de la parole dans les langues du monde. La prononciation d'un mot est modélisée comme une chaîne de symboles qui représentent des phones ou des segments.

Un phone est un son de la parole, les phones sont représentés par des symboles phonétiques qui ressemblent à des lettres d'une langue alphabétique comme l'anglais.

1.1.3 Langues tonales

Un ton est une hauteur appliquée à une certaine syllabe. La signification d'un mot peut varier en fonction des tons utilisés pour la prononciation des syllabes le constituant. Par exemple, si la syllabe est prononcée avec une hauteur plus élevée, le mot peut avoir une signification différente de celle lorsqu'elle est prononcée avec une hauteur plus basse.

Les langues tonales sont celles où le ton et la prononciation travaillent ensemble pour communiquer le sens. En d'autres termes, il y a une couche supplémentaire de complexité dans la prononciation de chaque mot qui affecte sa signification. Dans certaines langues tonales, le ton appliqué à une seule syllabe peut complètement changer le sens du mot ! [33]





1.2 Concepts de base du traitement de la parole

1.2.1 Définition du signal audio

Par définition, le signal audio constitue la partie audible du spectre des vibrations acoustiques. Un tel mécanisme se fonde principalement sur deux paramètres : la fréquence et l'amplitude de la vibration. Il peut être analogique ou numérique.

Un signal audio analogique peut être défini comme un ensemble continu d'informations.

Un signal audio numérique est un ensemble discret d'informations.

Un signal audio numérique est caractérisé par :

- Nombre de canaux sonores codés : mono, stéréo, multicanaux.
- Fréquence d'échantillonnage : nombre d'échantillons par seconde utilisés pour décrire numériquement le signal qui représente l'onde sonore pour chaque canal. La bande passante dépend étroitement de cette caractéristique.
- Résolution de chaque échantillon en bits
- Débit numérique : taille du fichier par rapport à la durée du son.

1.2.2 Numérisation du signal audio

La numérisation du signal audio est un procédé qui permet de transformer le signal audio analogique en un signal numérique c'est-à-dire une série de 0 et 1 qui peuvent être traités par un ordinateur ou d'autres dispositifs électroniques.

Les étapes de numérisation du signal audio sont les suivantes :

a. Échantillonnage

Il s'agit ici de découper le signal en échantillons d'une durée T_e , Fréquence d'échantillonnage correspond au nombre d'échantillons par seconde $F_e = 1/T_e$.

Théorème de Shannon *Pour numériser convenablement un signal, il faut que la fréquence d'échantillonnage soit au moins 2 fois supérieure à la fréquence du signal à numériser.*

Échantillonnage de la voix humaine

- Infrasons (< 20 Hz) : Ces fréquences sont en dessous du seuil de perception humaine et ne contribuent pas de manière significative au contenu audible de la voix humaine.
- Ultrasons (> 20 000 Hz) : Ces fréquences sont au-dessus du seuil de perception humaine et n'ont pas d'impact sur la qualité perçue de la voix humaine.



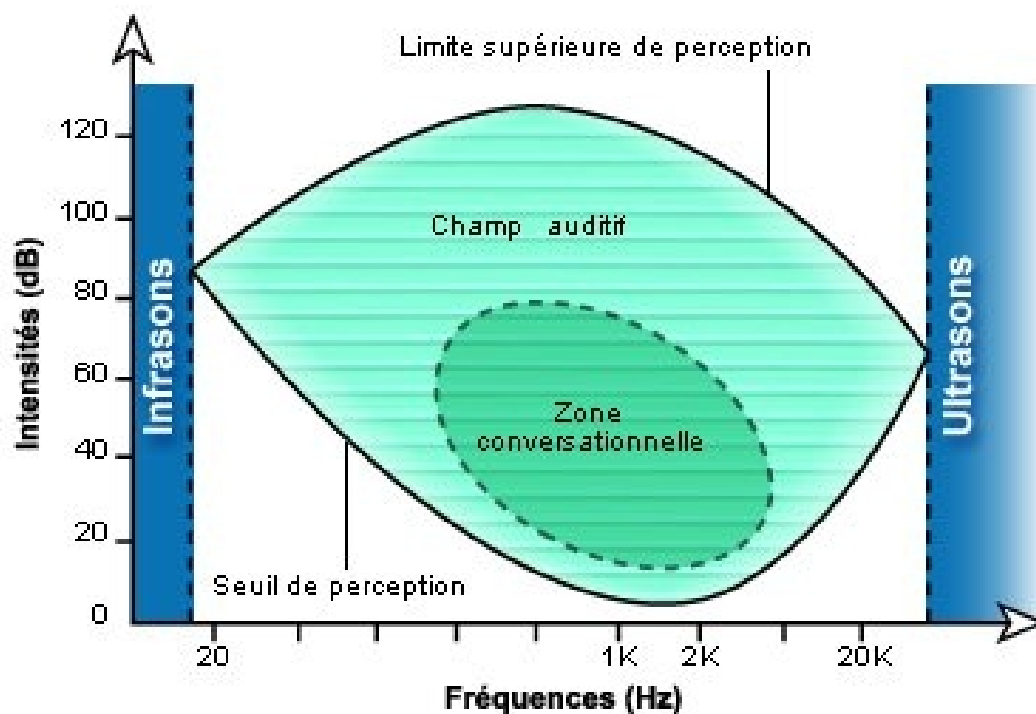


FIGURE 1.2 – La courbe audiométrique de l'oreille humaine [6]

La plage de fréquences de 250 Hz à 8 000 Hz représente le contenu fréquentiel essentiel de la voix humaine [26]. Selon le théorème de Shannon-Nyquist, pour échantillonner correctement un signal contenant des fréquences allant jusqu'à 8 000 Hz, la fréquence d'échantillonnage doit être d'au moins **16 000 Hz (16 kHz)**. Cela garantit que toutes les nuances importantes de la voix humaine sont capturées et peuvent être reproduites fidèlement lors de la restitution du signal.

b. Quantification La quantification consiste, pour chaque échantillon, à lui associer une valeur d'amplitude. Cette valeur de l'amplitude s'exprime en «**bit**».

c. Codage C'est l'action qui permet de transformer la valeur numérique de l'amplitude en valeur binaire.

Les méthodes de numérisation existantes sont les suivantes :

- PCM (Pulse Code Modulation) / ADPCM
- FLAC (Free Lossless Audio Codec)
- WAV (WaveForm Audio File Format)
- MP3 : MP3, MPEG-1 Audio Layer III
- uLAW / aLAW for telephone speech



1.3 Reconnaissance automatique de la parole

1.3.1 Définition

La reconnaissance vocale (également appelée reconnaissance automatique de la parole (RAP) ou reconnaissance vocale par ordinateur) est le processus de conversion d'un signal vocal en une séquence de mots, à l'aide d'un algorithme implémenté sous forme de programme informatique (Anusuya and Katti, 2009). [22] [34]

Théorie des probabilités de la reconnaissance de la parole

L'objectif principal d'un système ASR est de formuler l'hypothèse de la séquence de symboles discrets la plus probable parmi toutes les séquences valides de la langue L, à partir de l'entrée acoustique donnée. L'entrée acoustique ou signal audio est traitée comme un ensemble d'observations discrètes, de sorte que :

$$O = o_1, o_2, o_3, \dots, o_t$$

De même, la séquence de symboles à reconnaître est définie comme :

$$W = w_1, w_2, w_3, \dots, w_n$$

L'objectif fondamental du système ASR peut alors être exprimé comme suit :

$$W^* = \operatorname{argmax}_{W \in L} P(W | O)$$

Cette équation implique que, pour une séquence donnée W et une séquence d'entrée acoustique O, la probabilité $P(W | O)$ doit être déterminée. L'objectif est alors de décoder la chaîne de mots, basée sur la séquence d'observations acoustiques, de sorte que la chaîne décodée ait la probabilité maximale. Le théorème de Bayes peut être appliqué à cette probabilité pour obtenir l'équation suivante :

$$P(W | O) = \frac{P(O | W)P(W)}{P(O)}$$

Les quantités du côté droit de l'équation sont plus faciles à calculer que $P(W | O)$. $P(W)$ est définie comme la probabilité a priori de la séquence elle-même. Cette probabilité est calculée en utilisant la connaissance a priori des occurrences de la séquence W. Puisque $P(O)$ est la même pour chaque phrase candidate W, l'équation précédente peut donc être simplifiée comme suit :

$$W^* = \operatorname{argmax}_W P(O | W)P(W)$$





1.3.2 Architecture de base d'un modèle de reconnaissance de la parole

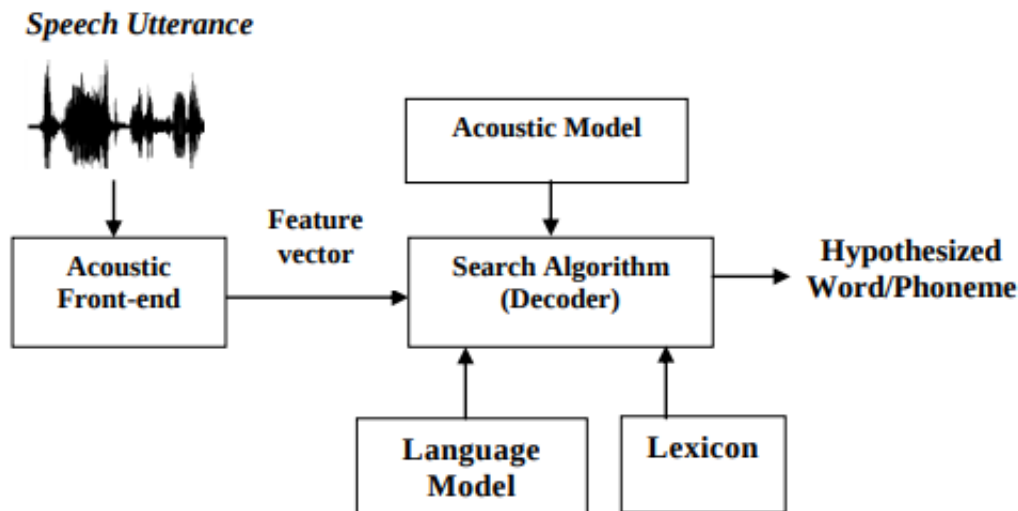


FIGURE 1.3 – Architecture de base d'un modèle de reconnaissance de la parole[16]

Dans la figure 1.3, qui présente les différents composants de l'architecture de base d'un ASR nous avons :

Frontal acoustique (Acoustic Front-end) : La frontale acoustique, souvent appelée front-end acoustique ou traitement du signal d'entrée, est la première étape cruciale dans un système de Reconnaissance Automatique de la Parole (ASR). Elle prépare le signal audio brut à être analysé par les modules suivants du système. Ses fonctions sont les suivantes :

❖ **Acquisition du signal**

- Choix du microphone : La qualité du microphone est essentielle. Il doit capturer le signal de manière claire, en minimisant le bruit ambiant.
- Amplification : Le signal peut être amplifié pour atteindre un niveau optimal pour le traitement numérique.

❖ **Prétraitement du signal**

- Enlèvement du silence : Les portions de silence sont identifiées et supprimées pour ne conserver que les parties parlées.
- Réduction du bruit : Des algorithmes de réduction de bruit sont appliqués pour atténuer les bruits parasites (bruit de fond, souffle du microphone, etc.).
- Normalisation : Le signal est normalisé pour avoir une amplitude moyenne constante, facilitant ainsi les comparaisons.

❖ **Extraction des caractéristiques :** Calcul des coefficients MFCC (Mel-Frequency Cepstral Coefficients) : Les coefficients MFCC sont les caractéristiques les plus couramment





utilisées en ASR. Ils capturent les informations les plus pertinentes pour la reconnaissance de la parole en tenant compte de la perception humaine. Certaines méthodes d'extraction de caractéristiques comprennent l'Analyse en Composantes Principales (PCA), l'Analyse Discriminante Linéaire (LDA), l'Analyse en Composantes Indépendantes (ICA), le Codage Prédicatif Linéaire (LPC), l'Analyse Cepstrale et l'Analyse à Échelle de Fréquence Mel.

Modèle acoustique

Le modèle acoustique est une sorte de dictionnaire statistique qui associe des sons (représentés par des coefficients acoustiques) à des phonèmes. Il permet de faire face à la variabilité de la parole humaine et de reconnaître les mots prononcés par différents locuteurs, même dans des conditions bruyantes ou avec des accents différents.

Le rôle du modèle acoustique :

- Apprentissage des variations : Le modèle acoustique est entraîné sur un grand corpus de données audio. Il apprend à associer chaque phonème à une multitude de réalisations acoustiques différentes. En effet, un même phonème peut être prononcé de manière légèrement différente selon les locuteurs, leur accent, le contexte, etc.
- Création d'un modèle probabiliste $P(O|W)$: Le modèle ne mémorise pas simplement des exemples sonores, mais il construit un modèle probabiliste. Ce modèle lui permet d'estimer la probabilité qu'une séquence donnée de coefficients acoustiques corresponde à un phonème particulier.
- Adaptation aux locuteurs : Grâce à cet apprentissage probabiliste, le modèle est capable de généraliser et de reconnaître de nouvelles réalisations de phonèmes, même s'il n'a jamais rencontré exactement la même prononciation auparavant.

Modèle de langage

Le modèle de langage est un modèle statistique qui prédit la probabilité $P(W)$ d'une séquence de mots W . Il a été entraîné sur d'énormes quantités de textes, ce qui lui permet d'apprendre les règles de la grammaire, les structures de phrases typiques, ainsi que les associations sémantiques entre les mots. En d'autres termes, un modèle de langue spécifie la distribution de probabilité des mots que le locuteur peut prononcer ensuite, en tenant compte d'un historique des mots prononcés. Les modèles de langue courants sont les modèles **bigram et trigram**.

- Un modèle bigram regarde les deux derniers mots pour prédire le suivant.
- Un modèle trigram regarde les trois derniers mots.

Recherche (Search) ou Décodage : Ce module combine les probabilités des modèles acoustique et de langue pour trouver la séquence de mots la plus probable (ou les meilleures options) à partir des observations acoustiques, produisant ainsi l'énoncé reconnu.

Mot ou phonème reconnu : qui est le résultat de l'ASR.

En résumé, Le modèle acoustique analyse le signal audio et propose une liste de phonèmes ou de





mots potentiels. Le modèle de langage évalue la probabilité de chaque séquence de mots proposée par le modèle acoustique en se basant sur sa connaissance de la langue. Le décodeur combine les informations du modèle acoustique et du modèle de langage pour trouver la transcription (mot ou phonème) la plus probable.

1.3.3 Méthodologies de Reconnaissance Vocale

Les méthodologies de reconnaissance vocale (ASR) sont généralement classées en trois approches : l'approche acoustique-phonétique, l'approche de reconnaissance de motifs et l'approche d'intelligence artificielle.

L'approche acoustique-phonétique repose sur l'identification des unités phonétiques distinctes dans le signal de parole. Elle comprend trois étapes principales : l'analyse spectrale et la détection des caractéristiques, la segmentation et l'étiquetage des régions acoustiques stables, et enfin la détermination des mots valides à partir des étiquettes phonétiques.[22]

L'approche de reconnaissance de motifs se compose de deux étapes : l'entraînement des modèles et la comparaison des modèles. Elle repose sur un cadre mathématique formel pour établir des représentations fiables des modèles de parole, permettant de comparer les discours inconnus avec des modèles appris. Cette approche, qui utilise des modèles statistiques comme les Modèles de Markov Cachés, est la méthode principale de reconnaissance vocale depuis six décennies.

L'approche d'intelligence artificielle combine les méthodes acoustique-phonétique et de reconnaissance de modèles. Elle utilise deux principales techniques de reconnaissance de modèles : le Dynamic Time Warping (DTW), qui compare les séquences de parole avec des modèles de référence en ajustant la durée, et les Hidden Markov Models (HMMs), qui sont désormais préférés pour leur meilleure généralisation et leurs besoins en mémoire réduits.

1.3.4 Types de Reconnaissance Vocale

Les systèmes de reconnaissance vocale peuvent être divisés en plusieurs classes différentes en décrivant les types d'énoncés qu'ils ont la capacité de reconnaître. Ces classes sont classées comme suit :

- **Mots Isolés** : Les systèmes de reconnaissance de mots isolés requièrent des pauses avant et après chaque mot, acceptant un seul mot ou une seule phrase à la fois.
- **Mots Connectés** : Les systèmes de mots connectés permettent de relier des phrases avec une pause minimale entre elles.





- **Parole continue** : Les systèmes de parole continue permettent aux utilisateurs de parler naturellement, comme pour la dictée, mais ils sont plus difficiles à créer car ils doivent déterminer les limites des phrases.
- **Parole spontanée** : Les systèmes de parole spontanée peuvent gérer un discours naturel avec des mots enchaînés, des hésitations ("euh", "ah") et des légers bégaiements.

1.3.5 Quelques applications de la reconnaissance vocale

Quelques applications du domaine de la reconnaissance vocale sont :

- Traduction : Application de traduction d'une langue à une autre, utilisant une forme d'onde vocale comme entrée et des modèles de mots parlés.
- Secteur éducatif : Apprentissage des langues avec reconnaissance vocale, utilisant des formes d'onde vocale et des modèles de mots parlés.
- Service client/Call centers : Réponse automatique et gestion des appels, utilisant une forme d'onde vocale comme entrée et des modèles de mots parlés.
- Secteur médical : Transcriptions médicales pour convertir la parole en texte, avec une entrée sous forme d'onde vocale et des modèles basés sur des mots parlés.
- Secteur automobile : Commandes vocales pour contrôler les fonctions du véhicule, avec une entrée sous forme d'onde vocale et des modèles basés sur des mots parlés.

1.3.6 Métriques d'évaluation des ASR

Word Error Rate (WER)

Le WER est une mesure de la précision du système ASR en termes de mots. Il est calculé en comparant la séquence de mots transcrits par le système avec la séquence de mots de référence (vérité terrain).

Le WER est calculé en utilisant l'édition minimale (insertion, suppression, substitution) nécessaire pour convertir la sortie du système en la vérité terrain.

Avec :

- S est le nombre de substitutions.
- D est le nombre de suppressions.
- I est le nombre d'insertions.
- N est le nombre total de mots dans la référence.

La formule est :

$$WER = \frac{S + D + I}{N}$$





Sentence Error Rate (SER)

Le SER mesure la proportion de phrases (ou d'énoncés) qui contiennent au moins une erreur (substitution, insertion ou suppression). Le SER est calculé comme suit :

$$SER = \frac{\text{Nombre d'énoncés incorrects}}{\text{Nombre total d'énoncés}}$$

1.3.7 Exemples d'outils de reconnaissance automatique de la parole

1.3.7.1 KALDI

Kaldi est un kit d'outils gratuit et open-source pour la recherche en reconnaissance vocale. Kaldi fournit un système de reconnaissance vocale basé sur des transducteurs à états finis (utilisant OpenFst, disponible gratuitement), ainsi qu'une documentation détaillée et des scripts pour construire des systèmes de reconnaissance complets. Kaldi est écrit en C++ et la bibliothèque principale prend en charge la modélisation de tailles de contexte phonétique arbitraires, la modélisation acoustique avec des modèles de mélange gaussien en sous-espace (SGMM) ainsi que des modèles de mélange gaussien standard, ainsi que toutes les transformations linéaires et affines couramment utilisées. Kaldi est publié sous la licence Apache v2.0, qui est très peu restrictive, ce qui le rend adapté à une large communauté d'utilisateurs.

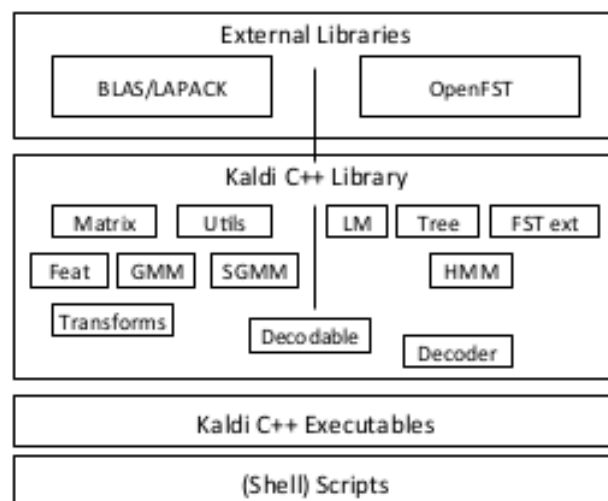


FIGURE 1.4 – Une vue simplifiée des différents composants de Kaldi..[11]

Présentation générale du fonctionnement de Kaldi

Pour mettre en place un système de reconnaissance automatique de la parole (ASR) avec l'outil Kaldi, les étapes suivantes doivent être respectées :

Étape 1 : Configuration de l'environnement de travail





La configuration de l'environnement de travail avec Kaldi implique la création d'un dossier de projet dans le répertoire 'egs' de l'installation de Kaldi, suivie de la création de liens symboliques vers les dossiers nécessaires ('steps', 'utils', 'src'). Ensuite, il faut modifier le fichier 'path.sh' pour inclure le chemin absolu vers l'installation de Kaldi, et créer un fichier 'cmd.sh' pour définir les commandes de formation et de décodage. Ces étapes permettent de préparer l'environnement pour l'exécution des scripts de traitement audio.

Étape 2 : Préparation des données

La préparation des données pour Kaldi consiste à créer manuellement et automatiquement les dossiers et fichiers nécessaires dans le dossier 'data' du projet. Cette étape comprend la création de fichiers essentiels comme 'utt2spk', 'segments', 'wav.scp', et 'text', qui décrivent les relations entre les fichiers audio, les locuteurs, et les transcriptions textuelles. Elle inclut également la préparation de fichiers de langage tels que 'lexicon.txt', 'nonsilence_phones.txt', 'optional_silence.txt', et 'silence_phones.txt', qui sont ensuite utilisés pour générer les dossiers de langage des données de test et d'entraînement.

Étape 3 : Extraction des caractéristiques MFCC et calcul des statistiques CMVN

L'extraction des caractéristiques MFCC et le calcul des statistiques CMVN dans Kaldi impliquent la configuration du fichier 'mfcc.conf' pour définir les paramètres d'extraction, puis l'exécution de scripts (voir annexe) pour traiter les données d'entraînement et de test. Les fichiers résultants 'feats.scp' et 'cmvn.scp' répertorient respectivement les chemins vers les fichiers de caractéristiques audio et les fichiers de normalisation. Le dossier 'exp' est utilisé pour stocker les journaux et résultats de ces opérations, ainsi que des étapes ultérieures comme l'entraînement du modèle acoustique et le décodage.

Étape 4 : Entraînement et alignement du modèle acoustique

Kaldi propose plusieurs modèles acoustiques qui peuvent être entraînés en tant que mono-phones ou triphones. Quelques exemples de modèles acoustiques sont :

- **HMM-GMM (Hidden Markov Model - Gaussian Mixture Model)** : Les mono-phones/tri-phones utilisent des HMM pour modéliser les séquences de phonèmes, avec des mélanges gaussiens pour modéliser les distributions d'observations acoustiques.
- **Modèles DNN-HMM (Deep Neural Network - Hidden Markov Model)** : Utilise des réseaux de neurones profonds (DNN) pour modéliser les observations acoustiques à la place des mélanges gaussiens.
- **Modèles TDNN (Time-Delay Neural Network)** : C'est un type spécifique de réseau de neurones conçu pour capturer des dépendances temporelles à long terme dans les données acoustiques.

Étape 5 : Création du modèle de langage





Kaldi offre un support pour les modèles de langage n-gram et l'intégration avec d'autres outils pour les modèles de langage plus avancés. La création du modèle de langage dans Kaldi avec un autre outil consiste à générer un fichier '.arpa', puis à le compresser en utilisant 'gzip' pour en réduire la taille. Ensuite, le fichier compressé est formaté avec la commande 'utils/format_lm.sh', qui transforme le modèle de langage en un format utilisable par Kaldi, permettant ainsi son intégration dans le pipeline de reconnaissance vocale.

Étape 6 : Décodage et Évaluation

Cette étape comprend :

- Génération du graphique de décodage qui prend en compte les données de langage de test et le modèle acoustique.
- Décodage qui prend en compte le graph de décodage et les données de test pour créer des fichiers contenant les résultats de chaque itération de décodage.
- Évaluation se fait en parcourant les fichiers résultats de décodage pour obtenir les meilleurs scores sur les métriques.

Kaldi présente deux principaux avantages qui le distinguent dans le domaine de la reconnaissance automatique de la parole (ASR). Tout d'abord, sa modularité et sa flexibilité permettent une personnalisation poussée des pipelines ASR, offrant aux chercheurs et aux développeurs la liberté de choisir et d'adapter chaque composant selon leurs besoins spécifiques. Ensuite, Kaldi bénéficie d'une communauté active et d'un support étendu, avec une abondance de ressources, de tutoriels et de forums, facilitant l'apprentissage et l'implémentation pour les nouveaux utilisateurs, aussi il offre des performances de pointe dans de nombreuses tâches de reconnaissance vocale. Cependant, sa principale limite réside dans sa complexité d'utilisation, qui peut représenter un défi pour les débutants en raison de la nécessité de comprendre et de configurer de nombreux scripts et paramètres techniques.

1.3.7.2 SpeechBrain

SpeechBrain est un kit d'outils de reconnaissance vocale open-source et tout-en-un disponible sur PyTorch[31]. Il est conçu en 2021 pour faciliter la recherche et le développement des technologies de traitement de la parole par réseaux neuronaux en étant simple, flexible, convivial et bien documenté. Il a été développé en tenant compte des principes de conception suivants : **Accessibilité** : SpeechBrain est conçu pour être facilement compréhensible par un large éventail d'utilisateurs, y compris les débutants et les praticiens. **Facilité d'utilisation** : SpeechBrain utilise une pile logicielle simple (c'est-à-dire, Python → PyTorch → SpeechBrain) afin d'éviter de gérer trop de niveaux d'abstraction. **Répliquabilité** : SpeechBrain promeut une science ouverte et transparente. Nous avons entraîné la plupart de nos modèles avec des données accessibles au public.



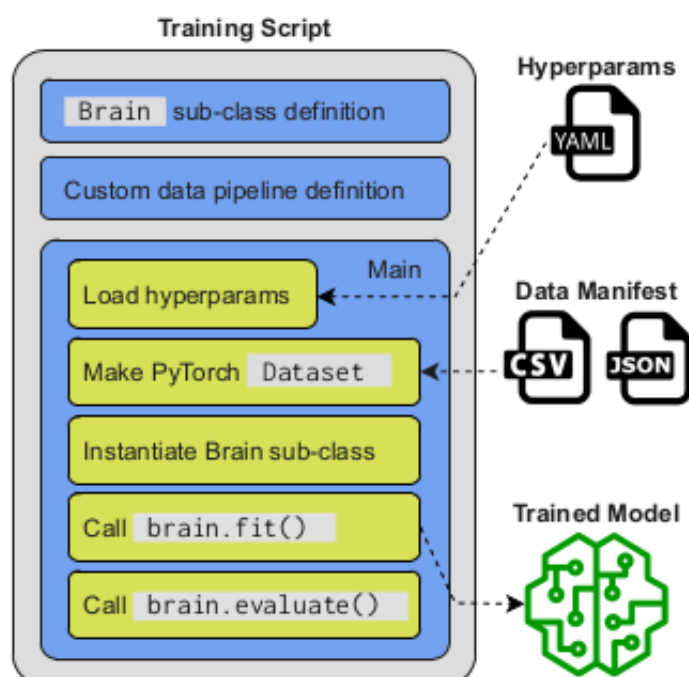


FIGURE 1.5 – Un aperçu d'un script de formation de base de SpeechBrain[11]

SpeechBrain a été implémenté pour supporter plusieurs tâches de traitement de langage et de la parole tels que : détection d'activité vocale, reconnaissance vocale, compréhension du langage parlé.

D'un point de vue architectural, SpeechBrain se situe entre une bibliothèque et un framework. La figure ci dessus illustre l'architecture d'un script d'entraînement dans SpeechBrain prend en entrees :

- Hyperparamètres (YAML) : Un fichier YAML contient les hyperparamètres nécessaires à l'entraînement.
- Manifeste de données (CSV/JSON) : Un ou plusieurs fichiers CSV ou JSON spécifient les emplacements et les métadonnées des données d'entraînement.

Et produit en sortie Modèle entraîné de reconnaissance vocale.



1.3.7.3 Whisper

Whisper est un système de reconnaissance automatique de la parole (ASR) entraîné sur 680 000 heures de données supervisées multilingues et multitâches collectées sur le web. Nous montrons que l'utilisation d'un ensemble de données aussi vaste et diversifié améliore la robustesse face aux accents, au bruit de fond et au langage technique. De plus, cela permet la transcription dans plusieurs langues ainsi que la traduction de ces langues vers l'anglais. Nous mettons en open source les modèles et le code d'inférence pour servir de base à la création d'applications utiles et pour poursuivre les recherches sur le traitement robuste de la parole.

1.4 Réseaux de neurones graphes GNN

Réseaux de Neurones Convolutifs

Les réseaux de neurones convolutifs (CNN) sont l'un des principaux types de réseaux de neurones utilisés pour la reconnaissance et la classification d'images.

Un CNN est généralement composé de quatre types de couches :

- Couche Convolutionnelle : Applique des filtres pour extraire les caractéristiques de l'image, comme les bords, les textures et les motifs.
- Pooling : Réduit la dimension des caractéristiques extraites, en conservant les informations essentielles, ce qui diminue la complexité et les besoins en calcul.
- Fonction d'Activation : Introduit des non-linéarités dans le modèle, permettant au réseau de capturer des relations complexes entre les caractéristiques.
- Fully Connected. : Connecte tous les neurones des couches précédentes, intégrant les caractéristiques extraites pour effectuer la classification finale.

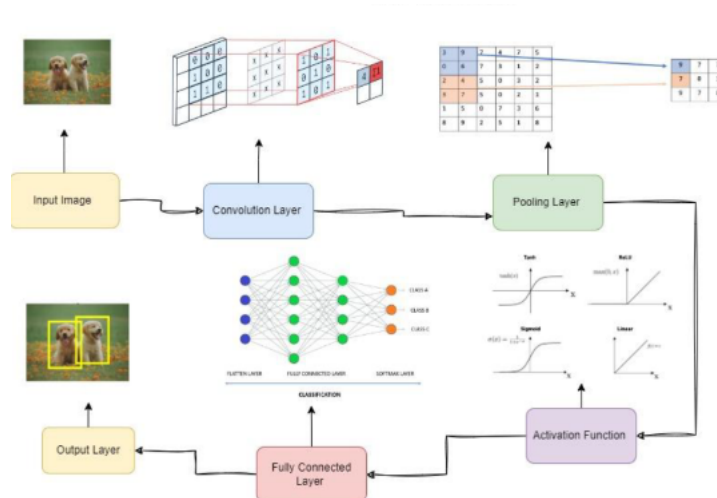


FIGURE 1.6 – Composants des CNN [37]



Les CNN sont appliqués à un graphe dans un espace euclidien tandis que les GNN sont appliqués à un graphe dans un espace non euclidien. L'espace non euclidien indique un espace plus arbitraire que l'espace euclidien en raison de ses connexions arbitraires entre les nœuds.

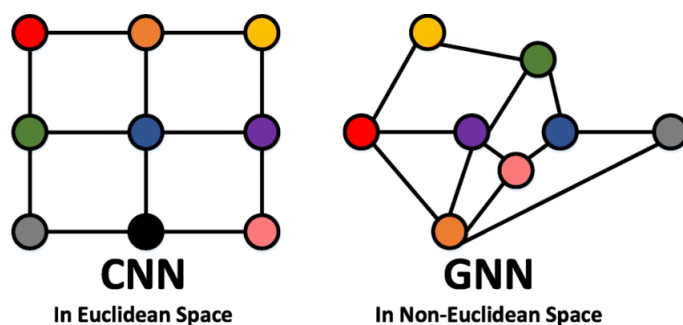


FIGURE 1.7 – Comparaison entre un CNN et un GNN

Les GNN sont une amélioration des CNN dans ce sens qu'ils peuvent accomplir des tâches que les CNN ne peuvent pas, notamment sur des données en forme de graphes. Alors que les CNN excellent dans l'identification d'objets et la catégorisation d'images, ils sont limités par la complexité et l'absence de localité spatiale des graphes, ainsi que par l'ordre non fixe des nœuds. Les GNN, en revanche, sont conçus pour traiter ces structures de données irrégulières de manière efficace.

Qu'est ce qu'un graphe ?

Un graphe est un ensemble de nœuds et de liens. Formellement un graphe est le couple $G = (V, E)$, où $V = 1, \dots, n$ est l'ensemble des n nœuds, et $E = (i, j) \mid i, j \in V$ est l'ensemble des liens entre eux. Un graphe peut également être représenté par une matrice d'adjacence A de taille $n \times n$ telle que : $A_{i,j} = 1$ si $(i, j) \in E$, 0 sinon

Un graphe peut représenter diverses structures de données, telles que les réseaux sociaux, les graphes de connaissances et les réseaux d'interaction protéine-protéine. Les graphes sont des espaces non euclidiens, ce qui signifie que la distance entre deux nœuds dans un graphe n'est pas nécessairement égale à la distance entre leurs coordonnées dans un espace euclidien. Cela rend l'application des réseaux neuronaux traditionnels aux données de graphe difficile, car ils sont généralement conçus pour des données euclidiennes.

Les GNN (Graph Neural Networks) utilisent un mécanisme de passage de messages pour agréger des informations provenant des nœuds voisins, leur permettant ainsi de capturer les relations complexes dans les graphes. Les GNN sont efficaces pour diverses tâches, y compris la classification des nœuds, la prédiction des liens et le clustering. Pour une tâche donnée on peut considérer : les caractéristiques des nœuds, des liens ou du graphe tout entier.

Une couche GNN décrit un procédé pour représenter chaque nœud dans un espace embedding. Considérons pour notre graphe en entrée :



- On associe a chaque nœud un vecteur x_i de taille d contenant les caractéristiques du nœud.
- La matrice X de taille (n, d) contenant les x_i des n nœuds du graphe
- La matrice A , matrice d'adjacence de taille (n, n)

Un GNN est une fonction f qui, il prend en entrée les matrices X et A et renvoie une matrice H d'embeddings des nœuds du graphe tel que :

$$H = f(X, A)$$

1.4.1 Architecture d'un GNN

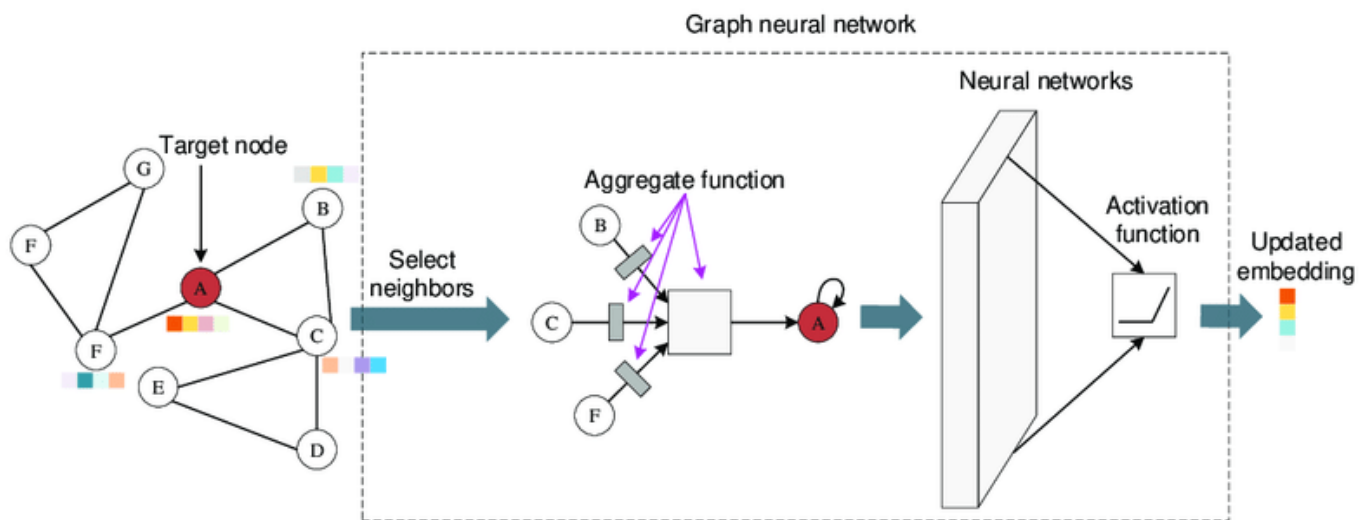


FIGURE 1.8 – Architecture de base d'un GNN.[44]

Fonctionnement

Pour produire ces embeddings les mécanismes suivants sont mis en oeuvre :

1. Passage de Message ou Message passing

Dans le mécanisme de passage de messages d'un réseau neuronal, chaque nœud stocke son message sous forme de vecteurs de caractéristiques, et à chaque itération, le voisin met à jour l'information sous forme de vecteur de caractéristiques.

Ce mécanisme englobe deux étapes qui sont :

- Sélection des voisins : Pour chaque nœud cible (par exemple, le nœud A dans l'image), les nœuds voisins sont identifiés à partir de la matrice d'adjacence A .
- Agrégation : Les caractéristiques des nœuds voisins sélectionnés sont combinées à l'aide d'une fonction d'agrégation (moyenne, maximum ou somme). Cela permet de regrouper l'information des nœuds voisins pour enrichir la représentation du nœud cible.





2. Transformation par Réseau de Neurones

Qui se deroule comme suit :

- Application d'un réseau de neurones : Les informations agrégées sont ensuite passées à travers un réseau de neurones.
- Fonction d'activation : Une fonction d'activation est appliquée pour produire une nouvelle représentation (embedding) mise à jour du nœud cible.

De maniere plus formel, le fonctionnement d'un GNN se resume suivant cette équation

$$h_u^{(k+1)} = \text{update}^{(k)} \left(h_u^{(k)}, \text{agg}^{(k)}(\{h_v, \forall v \in N(u)\}) \right) \quad (1)$$

Dans cette équation, $h_u^{(k)}$ est l'embedding actuel du nœud u où les embeddings (h_v) des nœuds voisins seront envoyés; $N(u)$, le voisinage du nœud u ; et $\text{update}^{(k)}$ et $\text{agg}^{(k)}$, des fonctions invariantes par permutation.

Il existe divers modèles de GNN qui diffèrent dans leur approche de la fonction d'agrégation ou de mise à jour exprimée dans l'équation (1) et dans leur capacité à effectuer des tâches de prédiction au niveau des nœuds, des arêtes ou du réseau. La théorie des trois modèles de GNN les plus utilisés est présentée ci-dessous.

1.4.2 Variantes des GNN

On distingue plusieurs types ou variantes de GNN. Nous allons presenter chacune d'elle en mettant en avant leurs forces et faiblesses.

● Graph convolution neural network (GCN)

Les GCNs sont une amélioration des CNNs qui peuvent être appliqués à des données graphes. Le mécanisme de "convolution" appliqué ici est le même que celui des Convolutional Neural Network (CNN), la valeur d'un nœud est obtenue en appliquant un filtre sur les nœuds voisins.

Ils utilisent un mécanisme de "message passing" basé sur une somme pondérée fixe des caractéristiques des nœuds voisins. Ce modèle est largement utilisé pour des tâches telles que la classification de nœuds et les systèmes de recommandation.

Les GCNs imposent des auto-connexions en définissant $\tilde{A} = A + I$ et empilent plusieurs couches convolutionnelles suivies de fonctions d'activation non linéaires.

$$H^{(k+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(k)} W^{(k)} \right)$$

Dans cette équation, H est la matrice des caractéristiques contenant les embeddings des nœuds comme lignes, et \tilde{D} désigne la matrice des degrés du graphe, qui est calculée comme





$\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$. De plus, $\sigma(\cdot)$ est une fonction d'activation, et W est une matrice de poids entraînable.

Parmi ses avantages, on trouve sa simplicité et son interprétabilité, ainsi que sa capacité à être entraîné de manière stable avec souvent moins d'époques. Cependant, les GCN ont des limites en termes de pouvoir expressif, une incapacité à capturer les structures locales complexes, et ne prennent pas en compte les caractéristiques des arêtes.

- **Graph Attention Networks (GAT)** Le Graph Attention Network (GAT) est une architecture de réseau neuronal conçue pour traiter et analyser les données structurées en graphe. Les GAT sont une variation des Graph Convolutional Networks (GCN) qui intègrent le concept de mécanismes d'attention.

Les Graph Attention Networks (GAT) utilisent un mécanisme de "message passing" basé sur une somme pondérée avec des poids appris, grâce à un mécanisme d'attention auto-appliquée. Ces réseaux sont particulièrement adaptés aux tâches de classification de nœuds et à toute application nécessitant des informations localisées.

L'expression pour les GATs est présentée dans l'Équation (4).

$$h_u^{(k+1)} = \sigma \left(\sum_{v \in N(u)} \alpha_{uv} W^{(k)} h_v^{(k)} \right)$$

Dans cette équation, α_{uv} représente les coefficients d'attention des voisins du nœud u , $v \in N(u)$, concernant l'agrégation des caractéristiques à ce nœud. Ces coefficients sont calculés comme

$$\alpha_{uv} = \frac{\exp(a^\top \text{LeakyReLU}(W[h_u, h_v]))}{\sum_{j \in N(u)} \exp(a^\top \text{LeakyReLU}(W[h_u, h_j]))}, \quad (5)$$

avec a désignant un vecteur d'attention entraînable

Les GAT offrent l'avantage de capturer des relations fines et d'améliorer les performances sur les tâches nécessitant une attention particulière à certains voisins spécifiques. Cependant, ils sont plus coûteux en termes de calcul que les GCN et peuvent être plus sensibles aux hyperparamètres.

- **Graph SAGE**

Apprentissage inductif Ce type d'apprentissage désigne la capacité d'un modèle à extraire des représentations significatives et généralistes sur les graphes (nœuds, liens, graphes), tout en étant capable de généraliser à de nouveaux (nœuds, liens, graphes) non observés lors de l'entraînement.





GraphSage est une variante de GNN, il est basé sur l'apprentissage inductive. Il est adapté aux gros graphes qui contiennent beaucoup d'informations au niveau des nœuds. Il utilise un mécanisme de "message passing" basé sur une somme pondérée fixe des caractéristiques des nœuds voisins. Ce modèle est particulièrement adapté à des tâches telles que la classification de nœuds et les applications où la scalabilité est cruciale.

GraphSAGE, un framework construit sur le modèle GCN original, met à jour l'information d'embedding de chaque nœud en échantillonnant le nombre de voisins à différentes valeurs de saut et en agrégeant leurs informations d'embedding respectives.

Ce processus itératif permet aux nœuds d'acquérir de plus en plus d'informations provenant de différentes parties du graphe. La principale différence entre le modèle GCN et GraphSAGE réside dans la fonction d'agrégation. Alors que les GCN utilisent un agrégateur moyen, GraphSAGE emploie une fonction d'agrégation généralisée. De plus, dans GraphSAGE, les caractéristiques propres ne sont pas agrégées à chaque couche. Au lieu de cela, les caractéristiques de voisinage agrégées sont concaténées avec les caractéristiques propres, comme le montre l'Équation (3).

$$h_u^{(k+1)} = \sigma \left(\left[W_{agg}^{(k)} \left(\{h_v^{(k)}, \forall v \in N(u)\} \right), B^{(k)} h_u^{(k)} \right] \right) \quad (3)$$

Dans cette équation, B est une matrice de poids entraînable, et agg désigne une fonction d'agrégation généralisée, telle que la moyenne, le pooling ou LSTM.

Ses avantages incluent sa capacité à s'adapter à des graphes de grande taille, des stratégies d'échantillonnage flexibles et une adéquation pour les graphes avec des degrés de nœuds variables. Cependant, les GraphSAGE ont une capacité limitée à capturer les structures globales du graphe et peuvent nécessiter plus d'époques pour un entraînement efficace.

1.5 Présentation de la langue Yemba

La langue yemba est une langue africaine parlée principalement dans la région de l'ouest Cameroun département de la Menoua. Cette région est située dans le groupe des Grassfields [13]. Le nom, "Yémba", signifie "je dis que", et a été proposé pour la première fois en 1983 par le professeur **Maurice Tadadjeu** et le pasteur **Fabien Wamba**. [43]

La langue yemba appartient à la famille des langues bantoïdes, plus spécifiquement au sous-groupe des langues bantoïdes méridionales. Cette famille linguistique fait partie des langues nigéro-congolaises, qui sont elles-mêmes classées dans la famille hypothétique des langues voltaïco-congolaises [24] [28].

Origines et Répartition :



La langue Yemba a une histoire riche et est répartie en cinq aires linguistiques distinctes [8] :

- **Yemba central** : Foto, Foréké, Fongo-Tongo, Fongo Ndeng, Fossong Eleleng, Fotetsa, Fotoumena et Fokoué
- **Yemba Est** : Bafou, Baleveng, Bamendou
- **Yemba Ouest** : Fossong Wecheng, Fondonerra, Fombap, Fontsa Toula
- **Yemba Sud** : Fomopea
- **Yemba SuD-Est** : Baloum

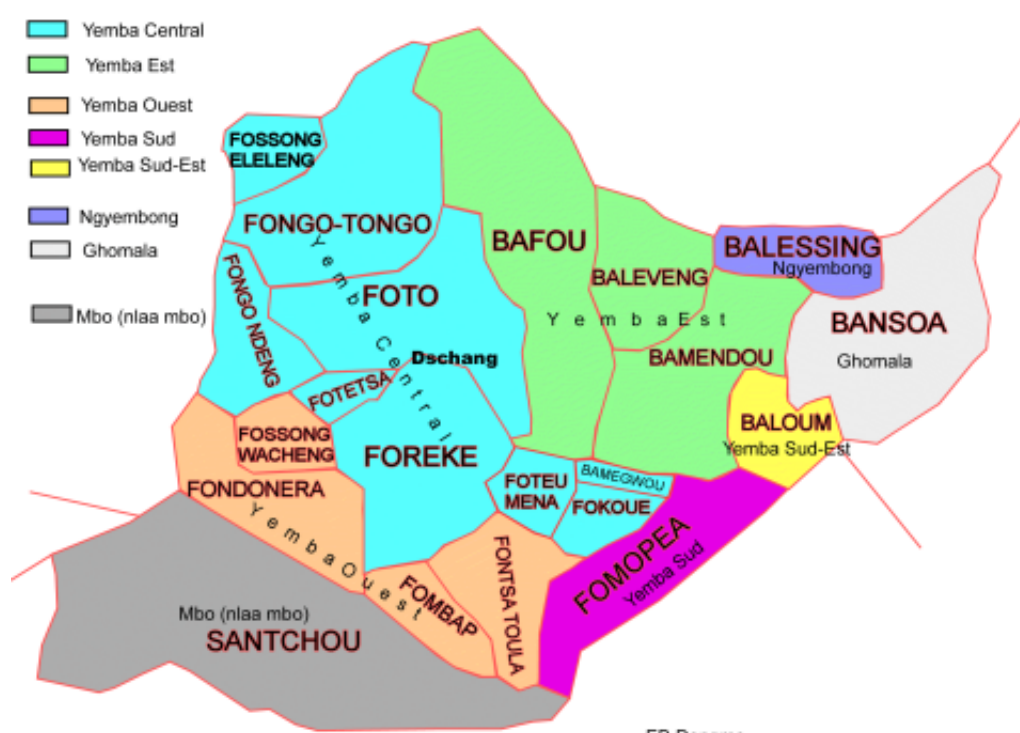


FIGURE 1.9 – Les régions où la langue Yemba est parlée et sa distribution géographique. [42]

Chacune de ces régions présente des variations linguistiques au niveau des tons, bien que les différences à l'intérieur de chaque aire soient minimales.

Caractéristiques Linguistiques

L'alphabet Yemba se compose de 33 lettres et est conforme à l'Alphabet Général des Langues Camerounaises (AGLC) publié en 1978 et adopté en 1984 par l'Association Nationale des Comités de Langues Camerounaises (ANACLAC) [19].

Les caractéristiques linguistiques de la langue yemba comprennent un petit alphabet et un système tonal distinctif. Les tons, haut, moyen et bas, sont une composante importante de la



langue yemba, avec des implications significatives pour la prononciation et la signification des mots[19].

- Ton haut : accent aigu sur
- Ton moyen : trait d'union
- Ton bas : ne s'écrit pas

a

FIGURE 1.10 – Ton bas pour la lettre “a”[19]

á

FIGURE 1.11 – Ton haut pour la lettre “a”[19]

ā

FIGURE 1.12 – Ton moyen pour la lettre “a”[19]



1.6 Étude de l'existant

La revue intitulée "Automatic Speech Recognition for Under-Resourced Languages : A Survey" de Laurent Besacier et al.[17] présente plusieurs défis pour la reconnaissance automatique de la parole (ASR) dans les langues à faible ressource. Parmi ces défis, on note particulièrement la variabilité acoustique élevée, qui fait référence aux différences significatives dans les accents, les dialectes et les environnements de parole. Ces variations rendent plus difficile pour les modèles de reconnaissance de la parole de comprendre et de transcrire correctement les mots prononcés. Il est donc crucial d'accorder une attention particulière à la manière dont les données sont représentées dans ces contextes. Dans cette section, nous allons présenter les architectures d'ASR disponibles dans la littérature en décrivant les études qui les intègrent. Pour chaque architecture, nous exposerons les avantages et les limites, afin de mieux comprendre leurs applications et leur efficacité dans la reconnaissance de la parole pour les langues à faible ressource.

1.6.1 Architectures basées sur les caractéristiques acoustiques

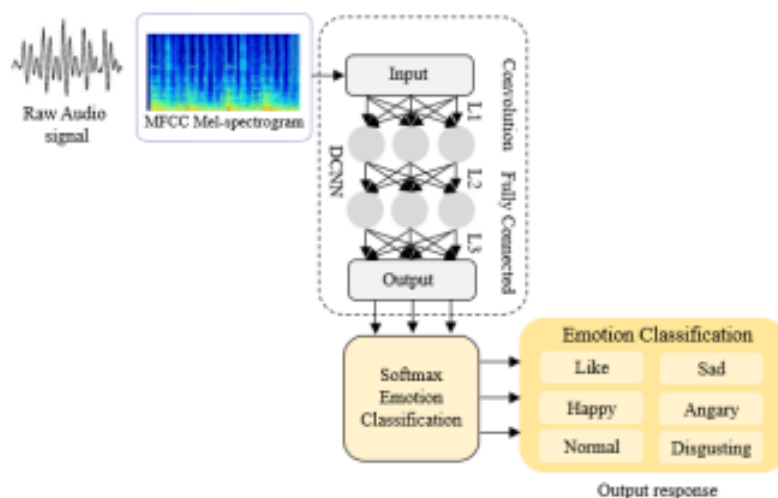


FIGURE 1.13 – Architecture d'un Système de Reconnaissance Vocale[2]

Cette étude [2] a proposé l'utilisation d'une technique bien connue d'extraction de caractéristiques basée sur la fréquence appelée MFCC, couramment utilisée pour améliorer les performances de classification des signaux de parole. L'auteur, Mahmood Alhlffee, a utilisé deux corpus différents de signaux vocaux bruts représentant cinq états émotionnels dans les bases de données RAVDSS et SAVEE. Les enregistrements étaient au format fichier WAV, totalisant entre 400 et 1000 fichiers avec des discours émotionnels en anglais. Le modèle proposé a atteint une précision de détection de 70% lorsqu'il a extrait des segments audio des fichiers et redimensionné les mots de parole en forme de spectrogrammes. Cependant, les méthodes de reconnaissance

MFCC-HMM sont sensibles aux conditions environnementales variées, limitant leur efficacité dans les applications réelles de systèmes de reconnaissance automatique de la parole.

1.6.2 Réseau neuronal convolutif pour la reconnaissance de la parole arabe

Ce travail, mené par Engy R. Rady et al., se concentre sur la reconnaissance automatique de la parole arabe (AASR) pour des mots isolés. Deux techniques sont utilisées lors de la phase d'extraction des caractéristiques : les coefficients spectraux de fréquence logarithmique (MFSC) et les coefficients cepstraux de fréquence Gammatone (GFCC) avec leurs dérivés de premier et de second ordre. Le réseau de neurones convolutifs (CNN) est principalement utilisé pour l'apprentissage des caractéristiques et le processus de classification. Le CNN a permis d'améliorer la performance de la reconnaissance automatique de la parole (ASR) grâce à sa connectivité locale, son partage de poids et le pooling. Le modèle CNN a été testé en utilisant un corpus de parole arabe pour des mots isolés, augmenté de manière synthétique par diverses transformations telles que la modification de la hauteur, de la vitesse, de la plage dynamique, l'ajout de bruit, et le décalage temporel. Le corpus, enregistré avec une fréquence d'échantillonnage de 44100 Hz et une résolution de 16 bits, contient 9 992 énonciations de 20 mots prononcés par 50 locuteurs natifs masculins arabes. Les résultats montrent que l'utilisation des GFCC avec CNN a permis d'atteindre une précision maximale de 99,77 %, surpassant les performances des études précédentes. Cependant, une limite de cette étude réside dans la focalisation sur des locuteurs masculins, ce qui pourrait ne pas représenter l'ensemble de la population.

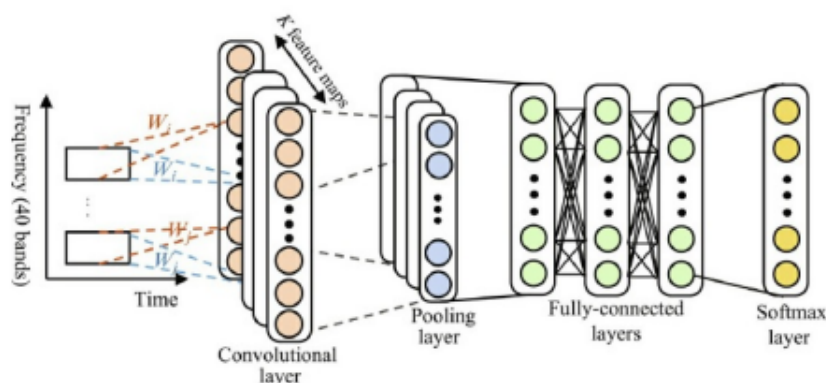


FIGURE 1.14 – Architecture CNN pour un ASR[12]

1.6.3 Reconnaissance des émotions de la parole basée sur le réseau neuronal Graph-LSTM

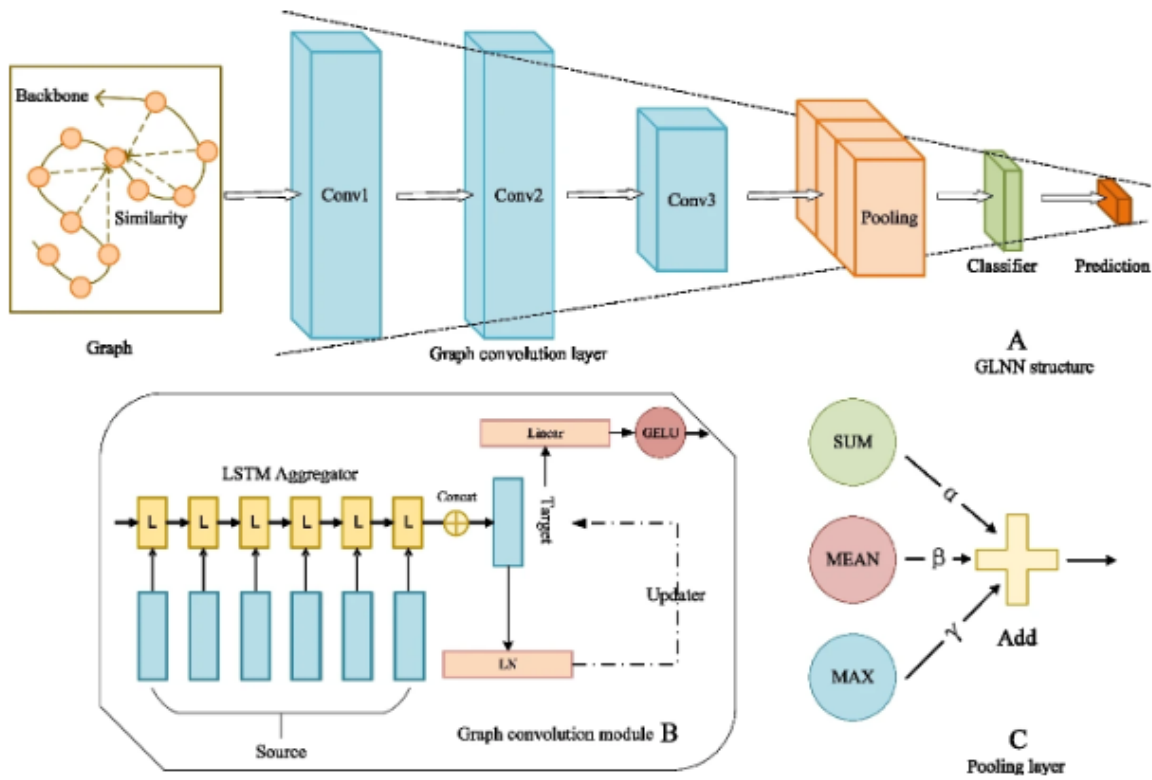


FIGURE 1.15 – Architecture du GLNN[20]

Dans cette étude [20], Yan Li et al. ont établi le réseau de neurones Speech-Graph GLNN Graph-LSTM, utilisant la similarité des caractéristiques et introduisant une architecture novatrice de réseau de neurones de graphe qui exploite un agrégateur LSTM et un pooling pondéré pour accomplir une tâche de reconnaissance des émotions dans la parole, incluant l'extraction des caractéristiques et la classification des émotions. Le jeu de données utilisé pour cette étude est la base de données Interactive Emotional Dyadic Motion Capture (IEMOCAP), contenant 12 heures de données audiovisuelles collectées à partir de dialogues situationnels entre deux personnes. Les graphes sont utilisés pour représenter les données, en construisant un graphe où chaque trame du signal vocal est considérée comme un nœud. Les résultats montrent que le modèle GLNN obtient une amélioration notable avec une précision pondérée (WA) de 71,83% et une précision non pondérée (UA) de 65,39%, surpassant les modèles de référence et démontrant une efficacité supérieure. Cependant, la recherche se concentre sur la parole des adultes, négligeant ainsi la reconnaissance des émotions dans la parole des enfants.



1.7 Bilan et positionnement

La revue de la littérature sur les architectures ASR nous a permis de comprendre que l'extraction des caractéristiques dans un ASR est une étape cruciale pour obtenir de bons résultats. Selon la tâche à accomplir, nous pouvons soit extraire des caractéristiques MFCC, soit utiliser un modèle de deep learning.

TABLE 1.1 – Tableau récapitulatif.

Critères	MFCC	GNN	CNN
Représentation des relations entre éléments	✗	✓	✗
Extraction des Caractéristiques Locales	✓	✓	✓
Adaptation aux structures de données complexes	✗	✓	✗
Sensibilité aux variations du signal	✓	✗	✓

Pour réaliser notre tâche de reconnaissance automatique de la parole (ASR), nous allons nous baser sur la capacité des GNN à représenter les relations complexes entre les données et à les modéliser de manière naturelle pour nos données audio.



MÉTHODOLOGIE

D *a ajouter , intro du chapitre*

2.1 Présentation de l' Architecture méthodologique

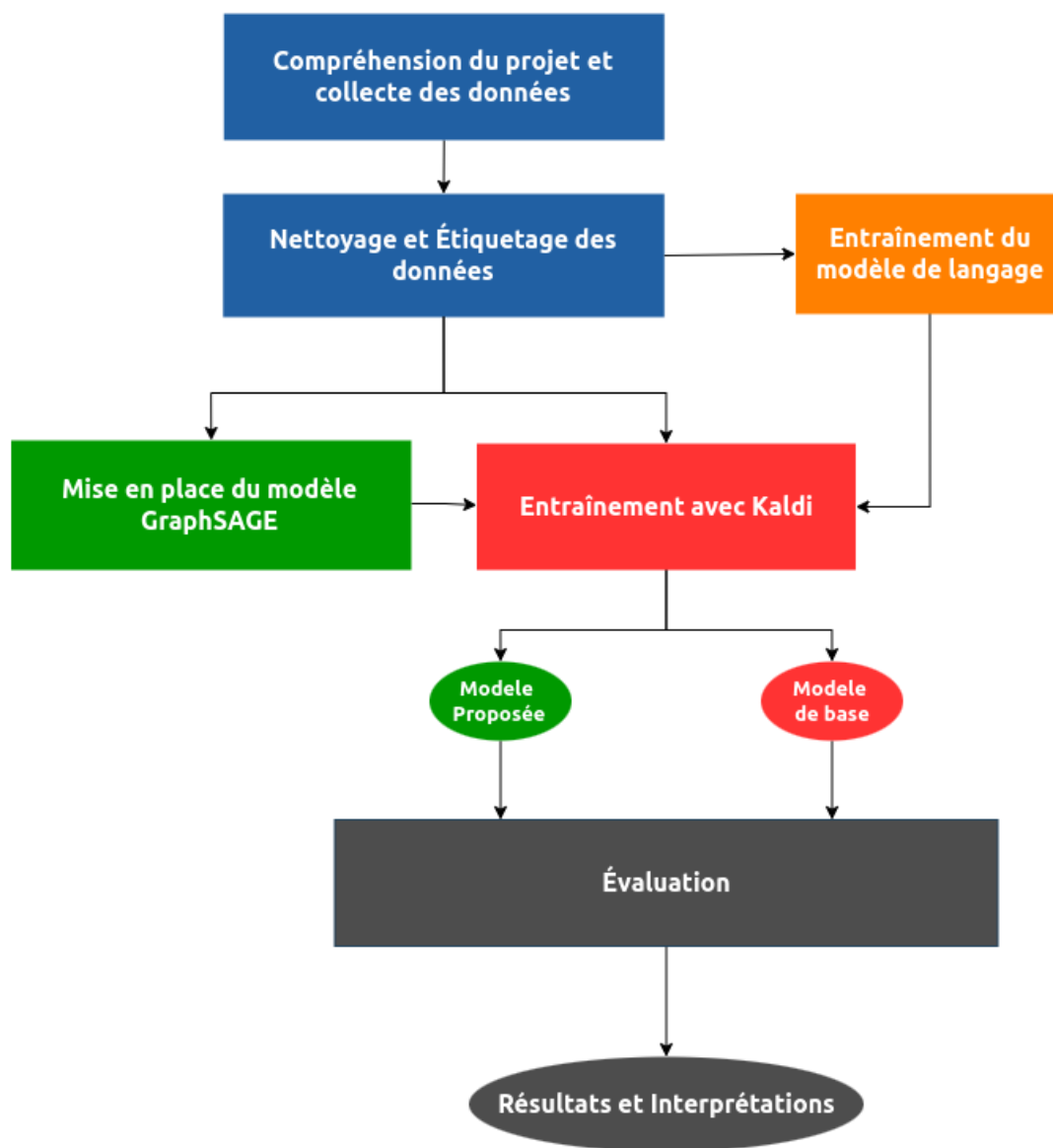


FIGURE 2.1 – Architecture méthodologique

La figure ci-dessus représente la méthodologie que nous avons élaborée pour mettre en œuvre notre solution. Elle illustre les différentes étapes impliquées dans la réalisation de celle-ci. Le diagramme est structuré en cinq blocs distincts, chacun identifié par une couleur spécifique :



- **Bloc bleu : Compréhension du projet et collecte des données** Ce bloc représente les étapes initiales du projet, où l'on se concentre sur la compréhension des exigences du projet et la collecte des données nécessaires. Ces données serviront de base pour les étapes ultérieures.
- **Bloc vert : Mise en place du modèle GraphSAGE** Ce bloc concerne le développement du modèle GraphSAGE, une méthode d'apprentissage basée sur des graphes, qui sera utilisée pour générer des embeddings à partir des données collectées. Cette étape est cruciale pour la création du modèle proposé.
- **Bloc orange : Entraînement du modèle de langage** Ce bloc est dédié à l'entraînement d'un modèle de langage.
- **Bloc rouge : Entraînement avec Kaldi** Ce bloc englobe toutes les expérimentations réalisées avec Kaldi, y compris l'entraînement du modèle acoustique et la mise en œuvre du modèle proposé (GraphSAGE) ainsi que du modèle de base. Cette étape permet de comparer les performances des différents modèles.
- **Bloc gris foncé : Évaluation et résultats** Ce bloc final regroupe l'évaluation des modèles expérimentés. Il inclut l'analyse des performances du modèle de base par rapport au modèle proposé, aboutissant aux résultats finaux et à leur interprétation.

Chaque bloc regroupe plusieurs sous processus. Les flèches unidirectionnelles entre les tâches indiquent la progression d'une tâche à une autre entre les différents blocs.

Une fois les données collectées, nettoyées et étiquetées, nous disposons de fichiers audio ainsi que d'un corpus de mots en Yemba. Le corpus de mots en Yemba est utilisé pour l'entraînement du modèle de langage. Parallèlement, les données audio sont exploitées pour la mise en place du modèle GraphSAGE et l'entraînement avec Kaldi, lequel utilise en entrée les résultats de l'entraînement du modèle de langage. L'entraînement avec Kaldi produit un modèle de base. Une fois le modèle GraphSAGE établi, ses résultats sont intégrés dans un nouvel entraînement avec Kaldi, aboutissant à la création de notre modèle proposé. Les deux modèles sont ensuite évalués simultanément, et les résultats obtenus sont analysés et interprétés.

Dans la suite nous allons présenter plus en détails les sous tâches de chaque bloc ainsi que les différentes interactions entre elles.



2.2 Compréhension du projet et collecte des données

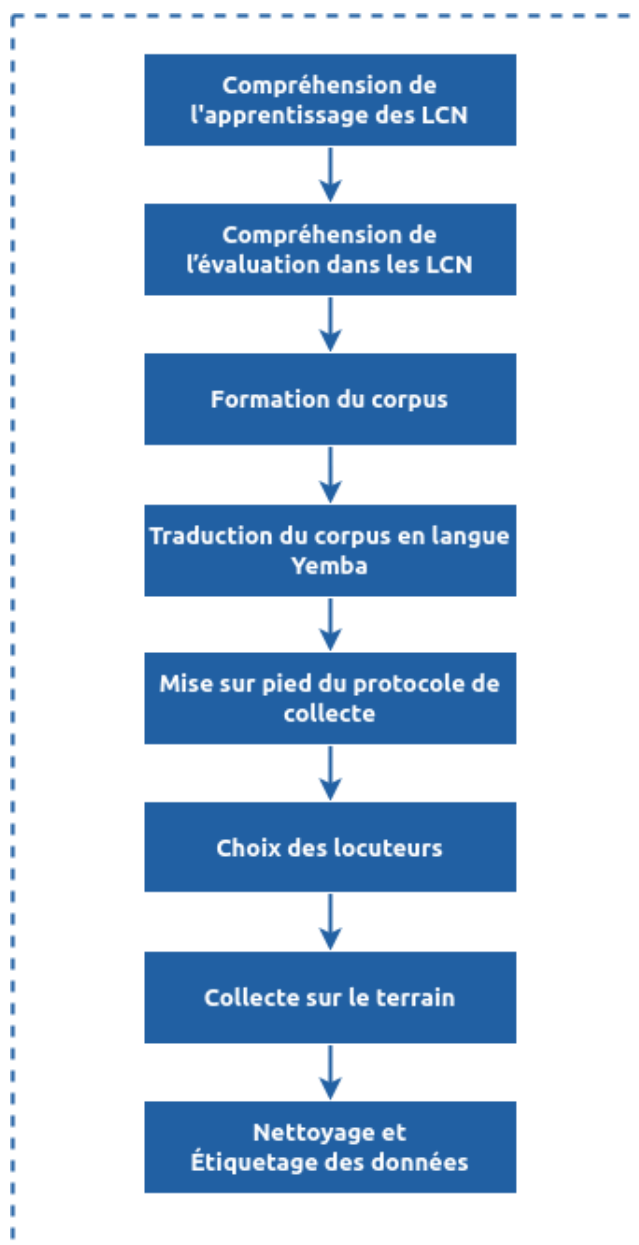


FIGURE 2.2 – Diagramme méthodologique : vue éclatée du bloc bleu

La figure ci-dessus est une vue éclatée du bloc Compréhension du projet et collecte des données de notre méthodologie, nous allons dans la suite explorer en détails chacun de ses sous blocs.

2.2.1 Compréhension de l'apprentissage et l'évaluation des LCN

L'école primaire est divisée en trois niveaux :





- ✓ Niveau 1 (Cycle des Initiations : SIL-CP)
- ✓ Niveau 2 (Cycle des Apprentissages Fondamentaux : CE1-CE2)
- ✓ Niveau 3 (Cycle des Approfondissements : CM1-CM2)

Les programmes d'enseignement pour ces niveaux sont décrits dans un document fourni par le ministère de l'Enseignement de base, intitulé **Curriculum de l'Enseignement Primaire Francophone Camerounais** [30]. Ce document comprend une section sur les langues et cultures nationales, présentant les attentes en matière d'apprentissage des langues locales.

La compétence visée ici est celle de pratiquer au moins une langue nationale. L'apprentissage des langues et cultures nationales au niveau 3 vise à développer chez l'apprenant la compétence de s'enraciner dans sa culture singulière et de s'identifier à la diversité de la culture nationale. Ces pratiques s'articulent autour de :

- ✓ Pratiquer les us et coutumes de sa localité d'origine
- ✓ Pratiquer les valeurs du multiculturalisme
- ✓ **Parler sa langue maternelle**
- ✓ Pratiquer les arts de sa localité d'appartenance

Le document contient également un tableau détaillant les attentes de fin de niveau en langues et cultures nationales, ainsi que les critères d'évaluation. Un autre tableau concerne la distribution des ressources en langues et cultures nationales, présentant les unités d'enseignement ou centres d'intérêt associés. Notre travail se situe principalement dans la partie "parler sa langue maternelle".

Les centres d'intérêts ou unités d'enseignement du niveau 3 sont au nombre de huit : **le village, la ville, l'école, les métiers, les voyages, la santé, les sports et loisirs, l'espace, et la nature**. Ces centres d'intérêt permettent de structurer l'apprentissage en abordant des thèmes variés et pertinents pour les élèves, facilitant ainsi une immersion complète dans la culture et la langue Yemba.

2.2.2 Formation du corpus

Pour constituer notre jeu de données, il est essentiel de préparer une liste de mots qui sera soumise aux locuteurs. Comme mentionné précédemment, l'apprentissage des langues et cultures nationales au niveau 3 est basé sur huit centres d'intérêts. Notre corpus, ou liste de mots, a été élaboré par un Animateur Pédagogique de la région du Littoral, un expert du terrain travaillant pour le MINEDUB.

La formation du corpus s'est déroulée de la manière suivante : l'expert a proposé, pour chaque centre d'intérêt, une liste de mots appartenant au champ lexical correspondant. Le





corpus ainsi formé contient 60 mots soigneusement sélectionnés pour représenter chaque centre d'intérêt.

Cette liste de mots couvre divers aspects de la vie quotidienne et culturelle, permettant ainsi une compréhension profonde et nuancée de la langue. Ce processus de sélection a pris en compte la pertinence culturelle et linguistique des termes, garantissant que le corpus soit non seulement représentatif mais aussi utilisable pour des locuteurs natifs.

TABLE 2.1 – Répartition des nombres mots par centre d'intérêts

Centre d'intérêts	Nombre de mots
Le village, la ville	9
L'école	8
Les métiers	8
Les voyages	8
La santé	8
Sports, loisirs	8
Dans l'espace	4
La nature	8

Le tableau suivant contient quelques mots par centre d'intérêt proposés par l'expert.

TABLE 2.2 – Répartition des mots par centre d'intérêts

Centre d'intérêts	Mots
Le village, la ville	La plantation, Le semis, Les services administratifs
L'école	La salle de classe, Le cahier, L'enseignant
Les métiers	L'agriculture, La conduite, La couture
Les voyages	Le bus, L'aéroport, Le bateau
La santé	Un médecin, L'ordonnance, La vaccination
Sports, loisirs	Le handball, Une paire de godasse, La piste de course
Dans l'espace	La Terre, Les étoiles, La lune
La nature	La pluie, La forêt, La rivière

2.2.3 Collecte et préparation des données

Les données étant l'élément central de tout projet d'apprentissage automatique, la mise en place du jeu de données doit se faire de manière progressive et rigoureuse. Ainsi, pour mettre sur pied notre jeu de données, nous avons suivis les étapes suivantes :

- ✓ Traduction du Corpus en Langue Yemba
- ✓ Mise sur pied du protocole de collecte
- ✓ Choix des locuteurs
- ✓ Collecte sur le terrain





2.2.4 Traduction du Corpus en Langue Yemba

Une fois le corpus de mots formé, nous l'avons remis à un linguiste spécialiste de la langue Yemba. Grâce à ses connaissances approfondies, il a traduit chaque mot de notre corpus. Ce processus de traduction a permis de garantir l'exactitude et la pertinence des termes dans le contexte culturel et linguistique spécifique aux locuteurs Yemba. La collaboration avec M. Tafoueme a été essentielle pour assurer que notre jeu de données reflète fidèlement la langue et les nuances culturelles des locuteurs natifs. Chaque mot est identifié par un numéro unique de 1 à 60.

TABLE 2.3 – Tableau de quelques mots après traduction

Id_word	Yemba	Français
1	Mberɲ	La pluie
2	Mbīŋ	La forêt
3	Míá ntshi	La rivière
4	Lekwēt	La montagne
5	Merɲwé'tsāɲ	Le chat
6	Nzenzhe	la mouche
7	Nu	Le soleil
8	Ŋgāp	La poule
9	Ŋkā'	La plantation
10	Aphíe ntsō	Le semis

2.2.5 Mise sur pied du protocole de collecte

Le protocole de collecte est constitué tel que suit :

- ✎ **Définition l'objectif de la collecte** Notre objectif principal pour cette collecte était de faire prononcer chaque mot de notre corpus en Yemba par les élèves sélectionnés comme locuteurs.
- ✎ **Identification des écoles pour la collecte** Pour garantir la qualité des prononciations, il était essentiel de choisir des écoles primaires publiques où les élèves sont natifs de la langue Yemba. Ainsi, nous avons identifié deux écoles primaires d'un village situé dans l'aire linguistique Yemba Est :
 - * École publique de Melah
 - * École publique de Toudjoua

Ces écoles sont situées dans la région de l'Ouest du Cameroun, département de la Menoua, arrondissement de Penka Michel, village de Bamendou.


- ✎ **Planification la collecte** Pour mener à bien notre collecte, nous avons planifié plusieurs descentes :





- La première avait pour but de prendre contact avec les administrations, les enseignants, et les élèves, ainsi que de recenser les élèves qui seraient nos locuteurs.
- Les deuxième et troisième descentes étaient dédiées à la collecte proprement dite.

La deuxième descente s'est déroulée du 15 mai au 18 mai 2024, tandis que la troisième descente a eu lieu du 24 mai au 27 mai 2024. La collecte a été réalisée avec l'autorisation des différents directeurs d'école et avec l'assistance du personnel enseignant. Pour mener à bien cette collecte, nous avons utilisé un appareil professionnel d'enregistrement audio, le Zoom H6 Audio Handy Recorder, afin d'assurer une qualité d'enregistrement optimale.

 **Conception d'une fiche de collecte** Pour assurer le bon déroulement de la collecte sur le terrain à chaque descente, nous avons conçu une fiche de collecte contenant les éléments suivants : **Date(début et fin), Nom de l'établissement, Description de l'école(avec localisation), tableau des personnes ressources, la classe, la période d'heure, le nombre d'enfants prévus, le nombre d'enfant effectifs, la série de mots utilisée, les photos, le nombre de fichiers audios enregistrés, les difficultés rencontrées et les recommandations.**

2.2.6 Choix des Locuteurs

Nous nous sommes rendu une première fois dans chaque école, nous avons visité les classes du niveau 3 (CM1-CM2). Le choix des élèves a été guidé par les conseils des maîtres et des maîtresses, en se basant sur les critères suivants :

- (1) Être né au village.
- (2) Résider avec un grand-parent.
- (3) Appréciation du maître ou de la maîtresse.

Les critères 1 et 2 garantissent que l'élève est immergé dans la langue depuis sa naissance et qu'il est en contact avec des personnes plus âgées depuis son plus jeune âge, ce qui favorise sa capacité à bien parler. Pour la sélection des élèves, nous avons utilisé l'une des combinaisons suivantes : 1 et 2, 1 et 3, ou 2 et 3, en veillant à choisir ceux qui répondent à ces critères de manière équilibrée.

Nous avons recensé **80 élèves** par école, répartis de manière équilibrée entre les classes, avec 20 élèves par classe, soit 10 filles et 10 garçons. Cette répartition a permis de garantir une représentation équitable des genres et de diversifier les échantillons linguistiques. Chaque élève a été évalué selon les critères de sélection définis, ce qui nous a permis d'identifier les locuteurs les plus aptes à participer à notre étude. En travaillant avec un nombre significatif d'élèves, nous avons assuré la diversité et la richesse des données collectées, renforçant ainsi la validité de notre corpus linguistique.





Les informations recueillies comprennent les caractéristiques démographiques et linguistiques des locuteurs, offrant ainsi une vue d'ensemble complète et détaillée de notre corpus. Ces caractéristiques incluent l'âge, le niveau scolaire, le genre et l'école fréquentée par chaque locuteur, le lieu d'enregistrement. Chaque enfant est identifié par un numéro unique.

TABLE 2.4 – Tableau statistique de l'échantillonnage des locuteurs

Nombre de locuteurs	80
Nombre de filles	40
Nombre de garçons	40
Age moyen	11,42
Age maximal	15
Age minimal	8

TABLE 2.5 – Détails de quelques locuteurs et des emplacements respectifs où leurs enregistrements ont eu lieu

IdLocuteur	Genre	Age	Locuteur Natif	Lieu de collecte	Niveau Scolaire
15	F	9	Oui	Classe d'école calme	CM1
16	F	9	Oui	Classe d'école calme	CM1
17	F	11	Oui	Classe d'école calme	CM1
18	F	8	Oui	Classe d'école calme	CM1
19	M	12	Oui	Classe d'école calme	CM1
20	M	11	Oui	Classe d'école calme	CM1
21	M	12	Oui	Classe d'école calme	CM2
22	M	14	Oui	Classe d'école calme	CM2
23	M	11	Oui	Classe d'école calme	CM2
24	M	15	Oui	Classe d'école calme	CM2
25	M	13	Oui	Classe d'école calme	CM2
26	M	14	Oui	Classe d'école calme	CM2
45	F	11	Oui	Salon calme d'une maison à côté de l'école	CM1
46	F	11	Oui	Salon calme d'une maison à côté de l'école	CM1
37	F	11	Oui	Classe d'école calme	CM2
38	M	12	Oui	Salon calme d'une maison à côté de l'école	CM1

2.2.7 Collecte sur le terrain

Dans une salle calme, nous avons enregistré chaque locuteur en train de prononcer chaque mot en Yemba de notre corpus. Les enregistrements ont été réalisés soit individuellement, soit en groupe de deux locuteurs. Chaque session consistait en la prononciation de 1 ou 2 mots à





la fois. Cette approche nous a permis de capturer des échantillons de haute qualité tout en minimisant les distractions et en assurant la clarté des enregistrements. Nous avons également veillé à ce que les locuteurs se sentent à l'aise et détendus, ce qui a contribué à obtenir des prononciations naturelles et fidèles à leur usage quotidien de la langue Yemba.

Les tableaux suivants présentent les distributions statistiques des locuteurs ayant participé à la collecte lors des descentes 1 et 2. Pour les analyses ultérieures, nous avons pris en compte uniquement les locuteurs qui ont participé aux deux descentes. Cette approche permet de garantir la cohérence et la fiabilité des données recueillies, en fournissant un échantillon représentatif de la population cible.

TABLE 2.6 – Tableau statistique des locuteurs descente 1

Nombre d'enfants prévu	80	40 CM1 (20 filles , 20 garçons) 40 CM2 (20 filles , 20 garçons)
Nombre d'enfants effectif	75	38 CM1 (20 filles , 18 garçons) 37 CM2 (18 filles , 19 garçons)
Série de mots	1&2	

TABLE 2.7 – Tableau statistique des locuteurs descente 2

Nombre d'enfants prévu	75	38 CM1 (20 filles , 18 garçons) 37 CM2 (18 filles , 19 garçons)
Nombre d'enfants effectif	69	32 CM1 (18 filles , 17 garçons) 37 CM2 (18 filles , 19 garçons)
Série de mots	2	

2.2.8 Nettoyage et Étiquetage des données

2.2.8.1 Nettoyage des Données

Les données recueillies ont été regroupées par locuteurs, puis chaque fichier audio a été écouté en utilisant le logiciel Audacity. Les fichiers qui ne contenaient rien, contenaient des erreurs ou n'étaient pas audibles ont été supprimés.



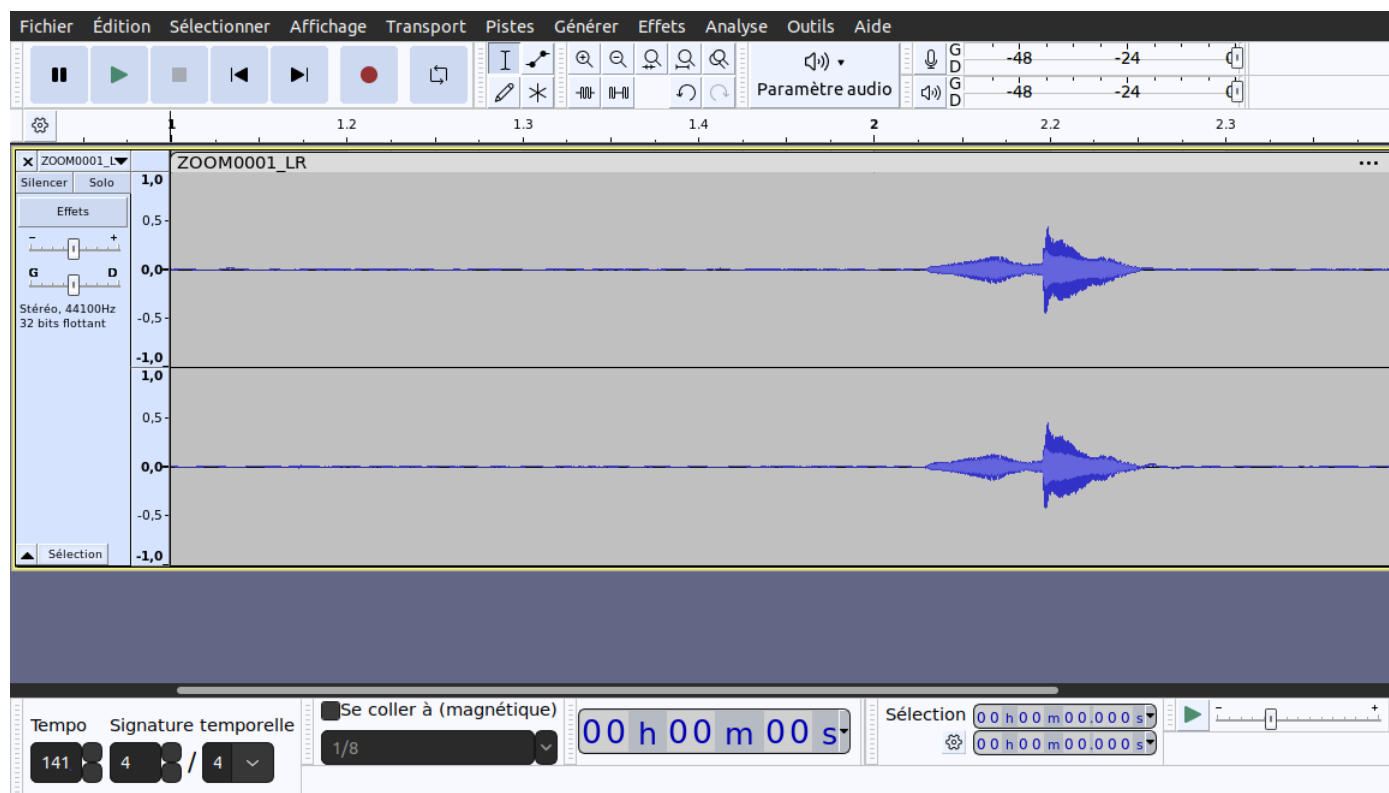


FIGURE 2.3 – Illustration d'un fichier audio sur Audacity

2.2.8.2 Étiquetage des Fichiers Audio

Les fichiers en bon état ont été renommés comme suit : spkr_X_word_Y_statement_i.wav, où :

- X représente l'identifiant du locuteur (allant de 1 à 69),
- Y représente l'identifiant du mot (allant de 1 à 60),
- i représente le numéro de la descente (1 ou 2).

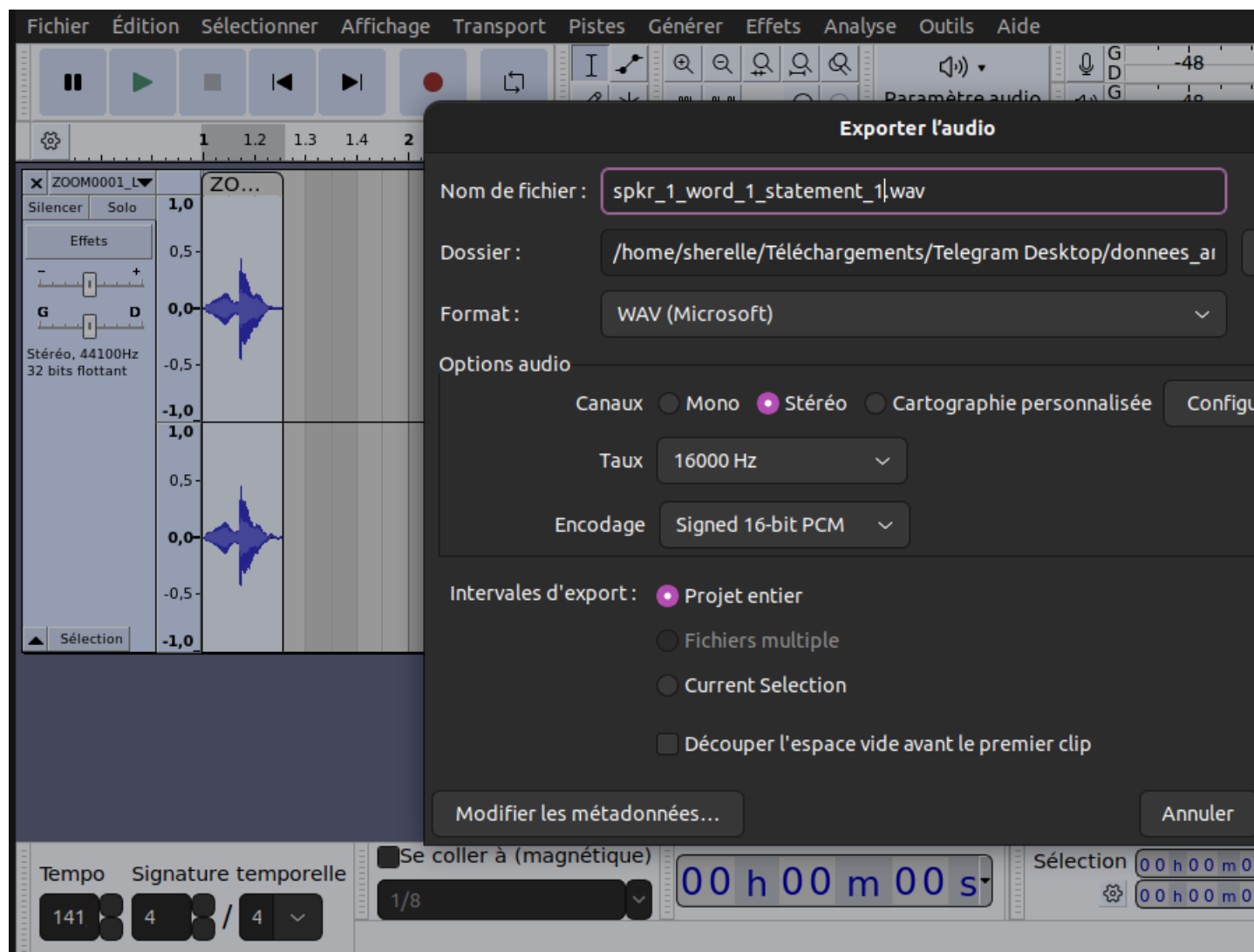


FIGURE 2.4 – Illustration de l'étiquetage d'un fichier audio avec Audacity

2.3 Entraînement avec Kaldi

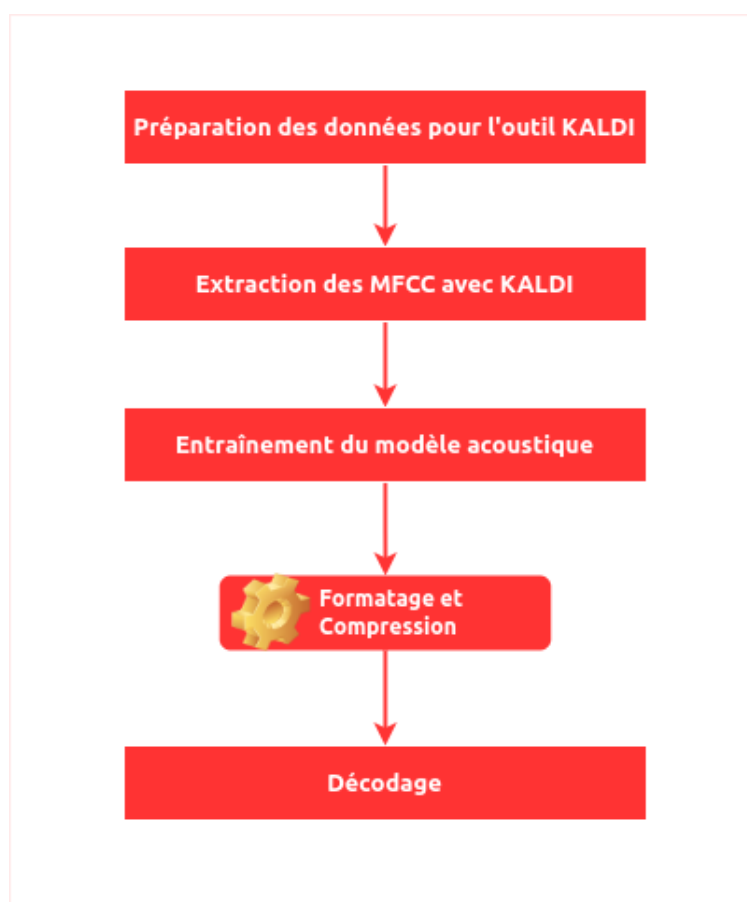


FIGURE 2.5 – Diagramme méthodologique : vue éclatée du bloc rouge

Ce schéma illustre les principales étapes de traitement des données dans le cadre de l'Entraînement avec l'outil KALDI.

- **Préparation des données** : Cette étape initiale consiste à organiser et formater les données brutes pour qu'elles soient compatibles avec les outils de KALDI. Cela inclut la création de fichiers de configuration, tels que wav.scp, text, utt2spk, et spk2utt, qui répertorient les fichiers audio et les transcriptions associées.
- **Extraction des MFCC** : Une fois les données préparées, KALDI extrait les coefficients cepstraux en fréquence de Mel (MFCC), qui sont des caractéristiques acoustiques cruciales. Ces caractéristiques servent de base pour l'entraînement du modèle acoustique.
- **Entraînement du modèle acoustique** : Cette étape consiste à utiliser les caractéristiques MFCC pour entraîner un modèle acoustique. Ce modèle est conçu pour capturer les relations entre les sons et leurs représentations numériques, en utilisant des algorithmes tels que les monophones et triphones.

- **Formatage et compression** : Une fois l'entraînement terminé, le modèle de langage de langage est compressé pour réduire la taille du fichier. Ensuite, le fichier compressé est formaté pour s'assurer qu'il est compatible avec les autres modules de KALDI et prêt pour le décodage.
- **Décodage** : Dans cette dernière étape, le modèle acoustique entraîné est utilisé pour décoder de nouvelles séquences audio, c'est-à-dire pour transformer les séquences MFCC en texte, en s'appuyant sur le modèle de langage formaté précédemment.

2.4 Mise en place du modèle GraphSAGE

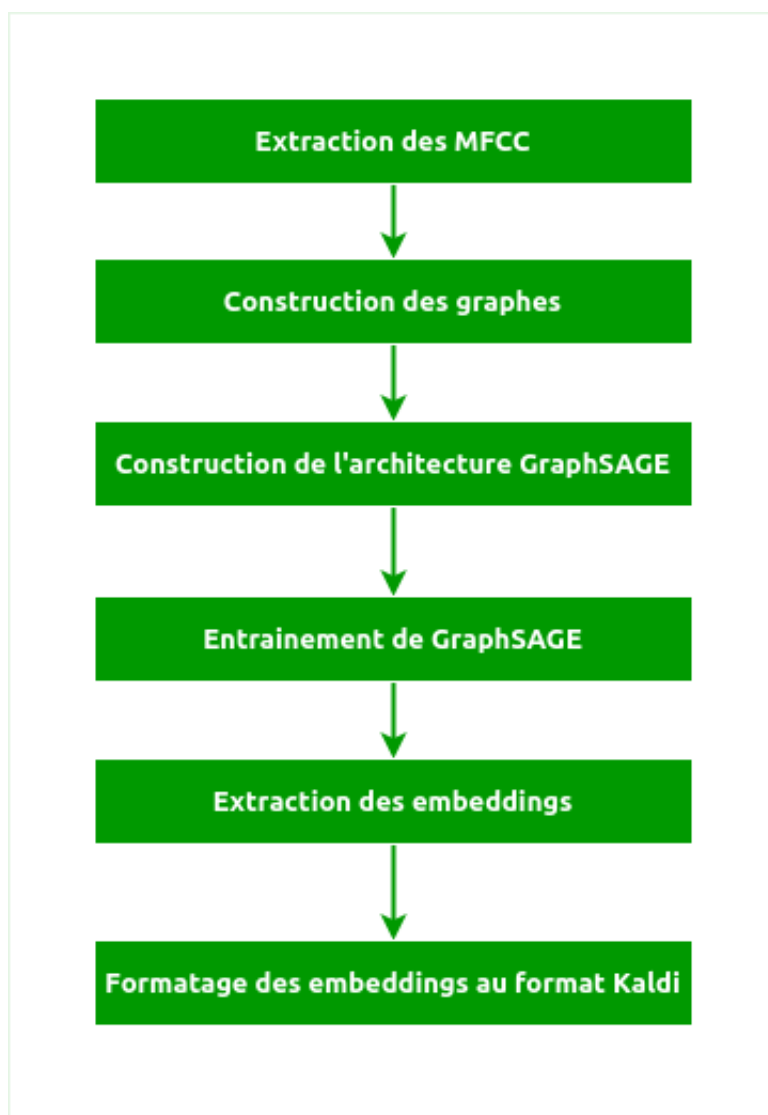


FIGURE 2.6 – Diagramme méthodologique : vue éclatée du bloc vert



2.4.1 Extraction des MFCC

La première étape consiste à extraire les coefficients cepstraux fréquentiels de Mel (MFCC) à partir des fichiers audio.

2.4.2 Construction des graphes

- Tous les audios correspondant à un même mot sont regroupés pour former un **cluster**.
- Chaque audio est représenté par un **nœud** dans le graphe.
- Au sein de chaque cluster, deux hyperparamètres sont définis :
 - k_i : nombre de voisins qu'un nœud doit avoir **à l'intérieur** du cluster.
 - k_o : nombre de voisins qu'un nœud doit avoir **en dehors** du cluster.
- Pour deux audios e_i et e_j appartenant au même mot, la pondération $W(e_i, e_j) = 1$. Cela indique une forte connexion entre ces nœuds.
- Pour chaque nœud e_i d'un cluster, k_o nœuds e_j sont choisis de manière aléatoire parmi les nœuds à l'extérieur du cluster et la pondération $W(e_i, e_j)$ est la similarité entre eux.

2.4.3 Entraînement de GraphSAGE

2.4.4 Extraction des embeddings

2.4.5 Formatage des embeddings au format Kaldi

Le processus de formatage consiste à convertir les vecteurs d'embeddings générés par le modèle GraphSAGE en des fichiers au format .ark, compatibles avec Kaldi.

2.5 Description de l'évaluation

Notre évaluation se fera à deux niveaux :

2.5.1 Évaluation avec des Métriques

2.5.2 Évaluation Empirique

Étant donné qu'il n'existe pas de benchmark pour notre cas, nous avons mis en place une plateforme d'évaluation.

Scénario d'Évaluation

- Un enseignant accède à la plateforme.
- Il enregistre un élève prononçant chaque mot de notre corpus en Yemba avec une prononciation correcte.
- Chaque enregistrement audio est soumis à notre solution ASR.





- La solution ASR retourne un audio à la plateforme.
- L'enseignant écoute et valide l'audio : s'il correspond à l'audio de départ, il note 1, sinon il note 0.

Après l'évaluation des 60 mots, les pourcentages de mots corrects et incorrects sont calculés comme suit :

$$(2.1) \quad \text{Pourcentage de mots corrects} = \left(\frac{\text{Nombre de 1}}{60} \right) \times 100$$

$$(2.2) \quad \text{Pourcentage de mots incorrects} = \left(\frac{\text{Nombre de 0}}{60} \right) \times 100$$



EXPÉRIMENTATIONS ET RÉSULTATS

C *e chapitre présente les environnements, outils de travail et configurations que nous avons utilisés pour la mise en œuvre de la solution décrite dans la méthodologie.*



3.1 Présentation du jeu de données

Le jeu de données YembaKidsVoice est composé de fichiers audio recueillis auprès de 69 enfants, garçons et filles, de niveau primaire (CM1-CM2). Ces données ont été collectées du 15 au 24 mai 2024 dans deux écoles primaires.

Les enregistrements ont été effectués à partir d'un corpus de 60 mots en langue Yemba, avec un taux d'échantillonnage de 16 kHz, en stéréo, et au format wav. Au cours de ces deux sessions de collecte, 5812 fichiers audio ont été enregistrés, répartis comme suit :

TABLE 3.1 – Répartition des fichiers par session de collecte

Série de mots	Session 1	Session 2	Total
1	2312	0	2312
2	1364	2136	3500
			5812

Après nettoyage et étiquetage, le jeu de données final contient un total de **8032** enregistrements, pour une durée totale de xx heures.

3.2 Expérimentations avec Kaldi

3.2.1 Présentation de l'environnement matériel

Matériel Utilisé

Pour cette expérimentation, nous avons utilisé le matériel suivant :

- **Processeur (CPU)** : Intel Core i5-6200U @ 2.30GHz × 4
- **Mémoire (RAM)** : 32 Go DDR4
- **Stockage** : 500Go SSD
- **Système d'exploitation** : Ubuntu 22.04.1 LTS

Logiciels Utilisés

Les logiciels et bibliothèques utilisés sont les suivants :

- **Kaldi** version 5.5
- **Python** 3.10.12
- **KenLM Language Model Toolkit**
- **GNU Bash**, version 5.1.16
- **GCC** 11.4.0





3.2.2 Configuration expérimentale

3.2.2.1 Préparation des Données

Renommage des fichiers audios

Pour préparer nos fichiers audio pour Kaldi, nous avons d'abord converti les fichiers stéréo en mono à l'aide d'un script Python, car Kaldi ne traite que les canaux mono. Ensuite, nous avons trié les fichiers de préparation dans l'ordre croissant basé sur les identifiants de speaker (spkr), de mot (word) et d'énoncé (statement).

Afin de faciliter ce tri, nous avons utilisé un script Python pour renommer les fichiers audio tel que, le fichier 'spkr_59_word_60_statement_2.wav' a été renommé en '0059_0060_0002.wav'.

Nous avons pris en compte que le nombre total de fichiers audio dans notre dataset nécessite une représentation sur 4 chiffres pour garantir une identification unique et ordonnée de chaque fichier. Cela facilite le tri des fichiers de préparation.

Création des Fichiers de Préparation pour le Dataset

Nous avons développé des scripts Python pour générer chacun de ces fichiers (segments, text, utt2spk, wav.scp).

Les données sont divisées en deux ensembles : 80% pour l'entraînement et 20% pour le test.

Les fichiers précédemment créés sont utilisés pour cette tâche. Le découpage consiste à séparer les données de chaque fichier en deux parties : une pour l'entraînement et une pour le test. Le script que nous avons écrit pour cette opération génère les fichiers segments, utt2spk, wav.scp et text dans les répertoires suivants : **data/train** et **data/test**

Le fichier **lexicon.txt** : nous l'avons créé nous même en coupant chaque mot en unité phonétique de longueur inférieure ou égale à deux. Une fois ces fichiers créés il faut valider les dossiers de langage d'entraînement et de test ce qui va permettre la création de manière automatique des fichiers supplémentaires liés au langage.

Nous avons extrait les caractéristiques MFCC des données de train et test. Pour le modèle acoustique nous avons entraîné plusieurs modèles :

- Le modèle de Markov caché HMM en mode monophone

Nous avons entraîné un modèle de langage de type trigramme basé sur notre corpus de mots en utilisant l'outil KenLm. Les métriques que nous avons utilisé sont : WER et SER.





3.3 Expérimentations de l'architecture proposée

3.3.1 Présentation de l'environnement matériel

3.3.2 Configuration expérimentale

exemples

https://journals.ekb.eg/article_160440_633339cb02bc485321dd14154e7c3f7b.pdf

<https://asmp-urasipjournals.springeropen.com/articles/10.1186/s13636-023-00303-9#abbreviations>

3.4 Résultat et interprétation

TABLE 3.2 – Répartition des mots par centre d'intérêts

	Modele Acoustique		WER	SER
MFCC	HMM monophonique	nj = 20	55,30%	70%
	HMM triphonique	nj = 10	xx	xx
	HMM	nj=2	xx	xx
MFCC+Pitch	HMM	nj=2	xx	xx
	HMM	nj=2	xx	xx



CONCLUSION GÉNÉRALE

C 4.1 Rappel du problème

4.2 Démarche et résultats

4.3 Limites

— Les niveaux 1 et 2 ne sont pas représentés dans notre jeu de données.

4.4 Perspectives

C 5.1 Présentation de UMMISCO

5.2 Expérimentation avec Kaldi

Nous présentons ici les commandes utilisées à chacune des étapes d'expérimentation avec Kaldi.

5.2.1 Étape 1 : Configuration de l'environnement de travail

- Créer le dossier du projet dans le dossier `egs` situé dans le dossier d'installation de Kaldi :

```
cd kaldi/egs
mkdir test1
```

- Dans le dossier du projet, créer des liens symboliques vers les dossiers `steps`, `utils` et `src` :

```
cd test1
ln -s ../wsj/s5/steps
ln -s ../wsj/s5/utils
ln -s ../../src
```

- Copier puis éditer le fichier `path.sh` dans le dossier de votre projet en ajoutant le chemin absolu vers le dossier d'installation de Kaldi :

```
cp ../wsj/s5/path.sh .  
nano path.sh # Ajouter la ligne suivante  
export KALDI_ROOT=$(pwd)/../..
```

- Créer et enregistrer `cmd.sh` dans le dossier du projet :

```
echo 'train_cmd="run.pl"' > cmd.sh  
echo 'decode_cmd="run.pl"' >> cmd.sh  
./cmd.sh
```

5.2.2 Étape 2 : Préparation des données

Préparation des données liées aux fichiers audios

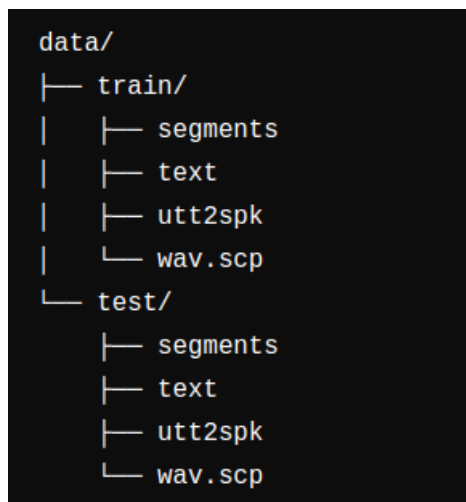


FIGURE 5.1 – Arborescence dossier data : préparation des fichiers audio

L'arborescence du dossier data contient les fichiers de préparation suivants, créés pour les données de test et d'entraînement :

- `utt2spk`
- `segments`
- `wav.scp`
- `text`

segments contient l'heure de début et de fin de chaque énoncé dans un fichier audio. Il est au format suivant :

```
utt_id fichier_id heure_début heure_fin
```

wav.scp fait le lien entre l'identifiant du fichier audio (file_id) et le chemin d'accès absolu vers ce fichier.

utt2spk contient la correspondance entre chaque énoncé et son locuteur correspondant.

text contient l'énoncé textuel. Il est sous la forme :

```
utt_id mot1 mot2...
```

Exemples du contenu des fichiers listés précédemment qui sont triés

segments

```
0001_0001_0001 0001_0001_0001 0.0 1.615
0001_0001_0002 0001_0001_0002 0.0 2.707
0001_0002_0001 0001_0002_0001 0.0 1.612
...
```

wav.scp

```
0001_0001_0001 /home/sherelle/Documents/yemba_dataset/0001_0001_0001.wav
0001_0001_0002 /home/sherelle/Documents/yemba_dataset/0001_0001_0002.wav
0001_0002_0001 /home/sherelle/Documents/yemba_dataset/0001_0002_0001.wav
...
```

text

```
0001_0001_0001 Mberɲ
0001_0001_0002 Mberɲ
0001_0002_0001 Mbinɲ
...
```

utt2spk

```
0001_0001_0001 0001
0001_0001_0002 0001
0001_0002_0001 0001
...
```

Préparation des dossiers et fichiers de langage



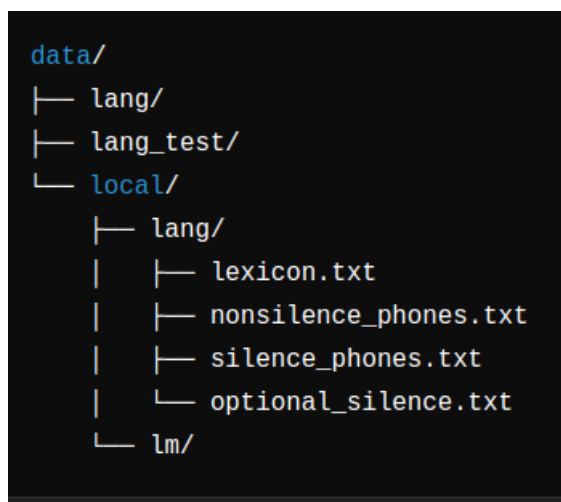


FIGURE 5.2 – Arborescence dossier data : préparation des dossiers et fichiers de langage

L'arborescence précédente contient :

- `lexicon.txt` : créé en coupant chaque mot en unité phonétique.
- `nonsilence_phones.txt` : ce fichier contient une liste de tous les phones qui ne sont pas silencieux. Il est créé à l'aide de la commande :

```

lexicon_file="data/local/lang/lexicon.txt"
nonsilence_phones_file="data/local/lang/nonsilence_phones.txt"
awk '{for(i=2;i<=NF;i++) print $i}' $lexicon_file | sort -u > $nonsilence_phones_file.txt
  
```

- `optional_silence.txt` : contient simplement un modèle sil :

```
echo 'sil' > optional_silence.txt
```

- `silence_phones.txt` : contient un modèle sil (silence) et spn (silence phone) :

```
echo -e 'sil'\n'spn' > silence_phones.txt
```

Une fois ces dossiers et fichiers préalables créés, les fichiers et dossiers de langage proprement dits sont créés à l'aide des commandes suivantes :

```

# Dossier de langage des données de train
utils/prepare_lang.sh data/local/lang 'oov' data/local/ data/lang

# Dossier de langage des données de test
utils/prepare_lang.sh data/local/lang 'oov' data/local/ data/lang_test
  
```





5.2.3 Étape 3 : Extraction des caractéristiques MFCC et calcul des statistiques CMVN

Le fichier `mfcc.conf` contient :

```
--use-energy=false  
--sample-frequency=16000
```

Les commandes pour le faire sont les suivantes :

Données de train

```
steps/make_mfcc.sh --cmd "run.pl" --nj nb_job data/train exp/make_mfcc/data/train mfcc  
steps/compute_cmvn_stats.sh data/train exp/make_mfcc/data/train mfcc
```

Données de test

```
steps/make_mfcc.sh --cmd "run.pl" --nj nb_job data/test exp/make_mfcc/data/test mfcc  
steps/compute_cmvn_stats.sh data/test exp/make_mfcc/data/test mfcc
```

`nb_job` : nombre de jobs définissant le nombre de lots de données qui seront traités à la fois.

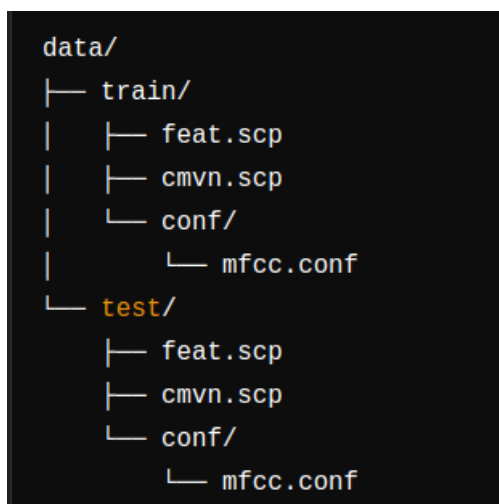


FIGURE 5.3 – Arborescence dossier data : extraction des caractéristiques



FIGURE 5.4 – Arborescence dossier exp





Le fichier feats.scp dans Kaldi contient les chemins vers les fichiers de caractéristiques (features) extraits des fichiers audio.

Le fichier cmvn.scp contient des lignes associant chaque identifiant de locuteur à son fichier de normalisation des moyennes et variances (CMVN)

Le dossier exp stock tous les logs et resultats des étapes d'extraction des caractéristiques, d'entraînement du modèle acoustique , du décodage et de l'évaluation.

5.2.4 Étape 4 : Entraînement et alignement du modèle acoustique

Cas du Modèle acoustique monophonique

Entraînement

```
steps/train_mono.sh --boost-silence 1.25 --nj 20 --cmd "run.pl" data/train data/lang exp/mono
```

Alignement

```
steps/align_si.sh --boost-silence 1.25 --nj 20 --cmd "run.pl" data/train data/lang exp/mono exp
```

5.2.5 Création du modèle de langage

5.2.5.1 Création du fichier arpa avec KenLM

- Ajouter le fichier du corpus(yemba_corpus.txt) de notre langage dans build/bin - Exécuter

```
bin/lmplz -o 3 < yemba\_corpus.txt > yemba.arpa --discount\_fallback
```

- Générer le fichier ARPA : build_binary trie yemba.arpa

5.2.5.2 Compression et formatage du fichier ARPA

compression

```
gzip -c data/local/lm/yemba.arpa > data/local/lm/yemba.arpa.kn.gz
```

formatage

```
utils/format_lm.sh data/lang data/local/lm/yemba.arpa.kn.gz data/local/lang/lexicon.txt data/la
```

5.2.6 Décodage et Évaluation

- Génération du graphique de décodage

```
utils/mkgraph.sh data/lang_test exp/mono exp/mono/graph
```

- Décodage

```
steps/decode.sh --nj 5 --cmd "run.pl" exp/mono/graph data/test exp/mono/decode_test
```





- Evaluation

```
for x in exp/*/decode*; do [ -d $x ] && grep WER $x/wer_* | utils/best_wer.sh; done
```



RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] *Proceedings of the Seventh Workshop on Culturally-Aware Tutoring Systems (CATS2024)*, July 2024. URL <https://hal.science/hal-04669404>.
- [2] Mahmood Alhlffee. *MFCC-Based Feature Extraction Model for Long Time Period Emotion Speech Using CNN*. Revue d'Intelligence Artificielle, 2020. DOI : <https://doi.org/10.18280/ria.340201>.
- [3] John Gibson Alicyn Warren. *Introduction to Digital Audio, Part 1*. Indiana University, 2013. adresse : <https://cecm.indiana.edu/361/digitalaudio1.html> (Visité le 8/02/2024).
- [4] G. Dimmendaal M. Lionel Bender. *Historical linguistics and the comparative study of African languages*. Amsterdam : John Benjamins Publishing Company. page 17-25.
- [5] O. Chaumette. *Numérisation d'un signal analogique*. Lycée JP Sartre.
- [6] controverses.sciences_po. *LE SON*. adresse : <https://controverses.sciences-po.fr/archive/implantscochleaires/son.html> (Visité le 04/07/2024).
- [7] Coursinfo. *Apprendre à maîtriser l'audio*. adresse : <https://www.coursinfo.fr/je-programme/apprendre-a-maitriser-laudio/> (Visité le 04/07/2024).
- [8] Edouard Dongmo. *Introduction langue Yemba*. Eveil Yemba, Oct 22, 2013. adresse : <https://eveilyemba.org/langue-yemba/introduction-langue-yemba/> (Visité le 17/05/2024).
- [9] Alec Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. OpenAI, 2022. adresse : <https://cdn.openai.com/papers/whisper.pdf>(Visité le 08/07/2024).
- [10] Bharti Khemani et al. *A review of graph neural networks : concepts, architectures,...* Journal of Big Data, 2024. <https://doi.org/10.1186/s40537-023-00876-4>.
- [11] Daniel Povey et al. The kaldı speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No. : CFP11SRW-USB.
- [12] Engy R. Rady et al. *Convolutional Neural Network for Arabic Speech Recognition*. Egyptian Journal of Language Engineering, 2021. DOI : https://journals.ekb.eg/article_160440_633339cb02bc485321dd14154e7c3f7b.pdf.

-
- [13] Hyman et al. *Grassfields Bantu*. 1984.
- [14] Jure Leskovec et al. *Graph Neural Networks*. KeAi, 2018. adresse : <https://arxiv.org/pdf/1812.08434>, DOI 1812.08434.
- [15] Karpagavalli et al. A review on automatic speech recognition architecture and approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9 : 393–404, 04 2016. doi : 10.14257/ijcip.2016.9.4.34.
- [16] Karpagavalli et al. A review on automatic speech recognition architecture and approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9 : 393–404, 04 2016. doi : 10.14257/ijcip.2016.9.4.34.
- [17] Laurent Besacier et al. Automatic speech recognition for under-resourced languages : A survey. *Speech Communication*, page 85–100, 2014. DOI 10.1016/j.specom.2013.07.008.
- [18] Mirco Ravanelli et al. *SpeechBrain : A General-Purpose Speech Toolkit*. arXiv, 2021. DOI 2106.04624v1.
- [19] Pierre-Marie Akenmo et al. *Petit Dictionnaire Yemba-Français*. CELY, 1997.
- [20] Yan Li et al. *Speech emotion recognition based on Graph-LSTM neural network*. SpringerOpen, 2023. DOI <https://doi.org/10.1186/s13636-023-00303-9>.
- [21] Jure Leskovec et Chenghan Jiang. *Representation Learning on Networks*. 2021.
- [22] M.A.Anusuya et S.K.Katti. *Speech Recognition by Machine : A Review*. (IJCSIS), 2009. adresse : <https://arxiv.org/pdf/1001.2267>, DOI 1001.2267.
- [23] Raphaël Fournier-S'niehotta. *Apprentissage sur graphes*. le cnam, 2021. HTTFOD RCP217 2020-2021.
- [24] J.H. Greenberg. *The Languages of Africa*. Bloomington : Indiana University, 1963.
- [25] Kenneth Heafield. *KenLM Language Model Toolkit*. 2013. adresse : <https://kheafield.com/code/kenlm/> (Visité le 24/06/2024).
- [26] Association lyonnaise des devenus sourds et malentendants. *Lire un audiogramme*. adresse : <https://www.aldsm.fr/2021/10/13/lire-un-audiogramme/> (Visité le 04/07/2024).
- [27] Daniel Jurafsky James H. Martin. *Speech and Language Processing : An introduction to natural language processing, computational linguistics, and speech recognition*. 25 Juin 2007.
- [28] P.M. Mbangwana. *Africa and the languages of the Atlantic Group*. 1996.
-





- [29] MINEDUB. *Loi d'orientation sur l'éducation : L'enseignement des langues maternelles est encouragé dans le système éducatif*. MINEDUB, 1998. adresse : <http://www.minedub.cm/>.
- [30] MINEDUB. *Politique nationale de promotion et de développement des langues maternelles*. MINEDUB, 2019. adresse : <http://www.minedub.cm/>, (Visité le : 16/04/2024).
- [31] Pytorch. *Pytorch Documentation*. Pytorch, 2021. adresse : <https://pytorch.org/docs/stable/index.html> (Visité le 08/07/2024).
- [32] S. Scardapane. *Neural Networks for Data Science Applications Master's Degree in Data Science Lecture 6 : Graph neural networks*. SAPIENZA UNIVERSITA DI ROMA, 2018.
- [33] Marisa Serrano. *What Are Tonal Languages?* Rosetta Stone, 27 Janvier 2022. adresse : <https://blog.rosettastone.com/what-are-tonal-languages/>, (Visité le : 03/07/2024).
- [34] Zhongzhi Shi. *Intelligence Science*. University Press., 2021. DOI <https://doi.org/10.1016/C2020-0-02066-9>.
- [35] SPEECHNEUROLAB. *Différences entre la parole, le langage et la communication*. SPEECHNEUROLAB, 25 Septembre 2020. adresse : <https://speechneurolab.ca/differences-entre-la-parole-le-langage-et-la-communication/>, (Visité le : 02/07/2024).
- [36] Romain Tavenard. *An introduction to dynamic time warping*. <https://rtavenar.github.io/blog/dtw.html>, 2021.
- [37] Mohammad Mustafa Taye. *Theoretical Understanding of Convolutional Neural Network : Concepts, Architectures, Applications, Future Directions*. 2023. <https://doi.org/10.3390/computation11030052>.
- [38] UNESCO. *La Conférence de Yaoundé sur l'enseignement en Afrique centrale est la première rencontre internationale organisée par l'UNESCO pour promouvoir l'enseignement des langues maternelles en Afrique*. UNESCO. adresse : <https://www.unesco.org/en/fieldoffice/yaounde>, (Visité le : 16/04/2024).
- [39] Petar Veličković. *Theoretical Foundations of Graph Neural Networks*. DeepMind, 2021. CST Wednesday Seminar.
- [40] Whisper. *Introducing Whisper*. OpenAI, 2022. adresse : <https://openai.com/index/whisper/> (Visité le 08/07/2024).
- [41] wikipedia. *Parole*. wikipedia. adresse : <https://fr.wikipedia.org/wiki/Parole>, (Visité le : 02/07/2024).
- [42] Wikipedia. *Yemba*. wikipedia. adresse : <https://fr.wikipedia.org/wiki/Yemba> (Visité le 02/07/2024).





- [43] YembaTV. *A Propos du Yemba*. YembaTV. adresse : <https://yembatv.org/a-propos-du-yemba/> (Visité le 07/04/2024).
- [44] Yufan et al. Zeng. *RLC-GNN : An Improved Deep Architecture for Spatial-Based Graph Neural Network with Application to Fraud Detection*. 18/06/2018. DOI - 10.3390/app11125656.
- [45] Éléonore Chodroff. *Tutoriel Kaldi*. 2018. adresse : <https://eleanorcho-droff.com/tutorial/kaldi/index.html> (Visité le 24/06/2024).

