**ASR pour le Yemba : Une Approche Seq2Seq avec GRU et Mécanisme d’Attention**

***Implémentation d’une architecture RNN pour la transcription syllabique et tonale de la langue Yemba***

|  |  |
| --- | --- |
| **Matricule** | **Noms** |
| **19m2315** | **Noubissi fopa christian junior** |
| **24F2456** | **Essuthi Mbangue ange Armel** |
|  | **Nguemtchueng tsamo bibiane** |
| **21T2635** | **Moukeki indjandja dave kevin** |
|  | **Abanda** |

**Sous la supervision de : Pr. Paulin Maletagia**

Sommaire

1. Presentation

* Contexte général : langues peu dotées et enjeux de l’ASR.
* Objectifs du projet : développer un modèle de reconnaissance vocale pour le yemba.
* Brève description des données utilisées (corpus YembaTones).

2. État de l’art

* Brève revue des approches existantes en ASR.
* Technologies adaptées aux langues peu dotées.
* Modèles classiques vs architectures modernes (CTC, attention, RNN, GRU...).

3. Données

* Description du corpus (structure, taille, type de transcription).
* Nettoyage et traitement :
  + Gestion des valeurs manquantes.
  + Nettoyage des libellés (ex. : "haut" vs "haut ").
  + Équilibrage éventuel des classes (syllabes, tons).
* Encodage des syllabes et tons (format final utilisé).

4. Prétraitement audio

* Méthodes utilisées (extraction de spectrogrammes, normalisation, padding).
* Paramètres audio : fréquence d’échantillonnage, durée moyenne, etc.

5. Architecture du modèle

* Description du modèle GRU Seq2Seq avec attention.
* Justification du choix de cette architecture.
* Détails des couches, activation, dimensionnalités, etc.

6. Entraînement

* Split du dataset (train/val/test).
* Hyperparamètres (learning rate, epochs, batch size...).
* Métriques de suivi pendant l'entraînement (loss, WER, etc.).

7. Évaluation

* Méthodologie d’évaluation (WER, CER, SER).
* Résultats obtenus.
* Analyse des erreurs (exemples de prédictions correctes/incorrectes).

8. Discussion

* Performances obtenues vs contraintes (taille du corpus, tonalité).
* Limites du modèle actuel.
* Difficultés rencontrées (déséquilibre des classes, manque de données, etc.).

9. Perspectives d’amélioration

* Suggestions pour améliorer le modèle (data augmentation, autres archis...).
* Intégration possible dans une application réelle (mobile, web...).

10. Conclusion

* Bilan global du travail.
* Apports du projet pour le yemba et les langues peu dotées.

11. Annexes (facultatif)

* Extraits du vocabulaire.
* Exemples de transcriptions.
* Schémas d’architecture, courbes de perte, etc.

1. Presentation
   1. Contexte

Le Cameroun est souvent qualifié d’**Afrique en miniature** en raison de sa diversité linguistique exceptionnelle : on y recense pres de **300 langues locales**, réparties en plusieurs familles (bantoïdes, nigéro-congolaises, etc.). Pourtant, malgré cette richesse, la majorité de ces langues sont considérées comme **peu dotées**, c’est-à-dire qu’elles disposent de peu de ressources linguistiques, éducatives ou technologiques. Il se heurte donc a plusieurs probleme notament :

* **Une faible intercompréhension** : Aucune langue ne fait l’unanimité au Cameroun, ce qui limite leur usage dans les sphères officielles ou intercommunautaires
* **Manque de standardisation** : Beaucoup de langues n’ont pas de norme écrite stable, ce qui freine leur enseignement et leur informatisation
* **Ressources limitées** : Peu de dictionnaires, grammaires, corpus numériques ou outils de traitement automatique existent pour ces langues.
* **Pression des langues officielles** : Le français et l’anglais, langues officielles, dominent l’espace public, l’éducation et les médias, reléguant les langues locales à un usage domestique ou communautaire.
* **Vitalité menacée** : Certaines langues sont en danger d’extinction, faute de transmission intergénérationnelle ou de valorisation institutionnelle

Ces langues véhiculent des savoirs, des cultures et des identités uniques. De plus des travaux récents, comme ceux utilisant l’intelligence artificielle pour évaluer la vitalité linguistique, ouvrent la voie à des outils de documentation et de revitalisation. C’est donc dans ce cadre que l’ASR (Automatic Speech Recognition) permet de documenter oralement les langues menacées, en automatisant la transcription de corpus parlés. Cela facilite donc la création de ressources linguistiques numériques permettant ainsi de préserver les tons, les rythmes et les structures syllabiques, essentiels dans des langues comme le Yemba.

* 1. Problematique

Malgré les avancées significatives dans le domaine de la reconnaissance vocale, ces technologies restent **largement concentrées sur les langues à forte dotation**, telles que l’anglais, le mandarin ou le français. En revanche, les langues locales camerounaises comme le Yemba, caractérisées par une **structuration tonale complexe**, demeurent à la marge des systèmes de transcription automatique, faute de ressources linguistiques, de corpus annotés et d’outils adaptés.

Ainsi, la problématique que ce projet entend aborder peut-être formulée comme suit : **Comment concevoir un système de reconnaissance vocale capable de transcrire automatiquement les énoncés oraux en Yemba en prenant en compte la structure syllabique et tonale de cette langue peu dotée, à partir d’une architecture de type RNN ?**

* 1. **Motivations du projet**

Le Cameroun, pays à forte diversité linguistique, compte plus de 280 langues nationales, dont la grande majorité est considérée comme **peu dotée** sur le plan technologique et scientifique. Parmi celles-ci, le **Yemba**, langue tonale du groupe bantoïde méridional, ne dispose que de **très peu de ressources numériques**, en particulier dans le domaine du traitement automatique de la parole (TAP). Dans ce contexte

* Le développement d’un système de **reconnaissance vocale automatique (ASR)** pour le Yemba contribuerait à **préserver et valoriser le patrimoine linguistique** camerounais.
* Ce projet s’inscrit dans une dynamique d’**inclusion technologique**, en explorant les possibilités d’intégrer les langues locales aux technologies vocales émergentes.
* L’approche proposée, fondée sur une **architecture RNN**, se veut adaptée aux **spécificités phonologiques du Yemba**, en particulier la **tonalité** et la structure **syllabique**, souvent ignorées dans les modèles généralistes.
* Enfin, la mise en œuvre d’un tel système représente un **défi scientifique stimulant**, à la croisée du traitement du signal, de la linguistique computationnelle et de l’apprentissage automatique.
  1. Objectifs du projet

L’objectif principal de ce projet est de développer un système de reconnaissance vocale automatique (ASR) adapté à la langue Yemba, une langue tonale parlée dans l’Ouest du Cameroun. Plus précisément, il s’agit de concevoir un modèle basé sur une architecture RNN (Réseaux de Neurones Récurrents), capable d’effectuer une transcription syllabique et tonale fidèle des énoncés oraux en Yemba. Ce projet vise à :

* Explorer les possibilités des RNN pour capturer les dynamiques temporelles propres aux langues à tons, notamment en ce qui concerne la modulation syllabique et la variation tonale.
* Automatiser la transcription des données audio en prenant en compte les spécificités phonologiques du Yemba, telles que les tons lexicaux et les structures syllabiques complexes.
* Contribuer à la numérisation et à la préservation des langues camerounaises peu dotées, en enrichissant les ressources disponibles pour le traitement automatique du langage dans un contexte multilingue.
* Fournir un prototype fonctionnel pouvant servir de base à des applications concrètes (éducation, traduction, interfaces vocales, documentation linguistique).
* Réduire la fracture linguistique numérique, en facilitant l’intégration du Yemba dans les technologies vocales émergentes.
  1. Methodologie generale

Le présent projet adopte une approche expérimentale pour développer un système de reconnaissance vocale automatique (ASR) spécialisé dans la transcription **syllabique et tonale** de la langue **Yemba**. L’architecture proposée repose sur un modèle **Seq2Seq** basé sur des unités **GRU (Gated Recurrent Units)**, renforcé par un **mécanisme d’attention**, afin de capturer efficacement la dynamique temporelle et les spécificités tonales de cette langue peu dotée. L’architecture du modele repose sur un encodeur, une attention un decodeur et une fonction de perte base sur la difference entre les differentes positions des caracteres et des syllabes.

* 1. Plan

Dans la suite du documents, nous detaillerons les etapes qui nous ont permis de passer d’obtenir ce model depuis le telechargement des donnees sur le site <https://data.mendeley.com/datasets/cx268tmrwn/3> jusqu’au deploiement de l’application en passant par le nettoyage, l’analyse, et le pretraitement des donnees, la conception, l’entrainement, l’optimisation, et l’evaluation.

1. Revues des solutions existantes
2. Description des donnees

Le jeu de données **YembaTones** repose sur un dictionnaire bilingue (Yemba–Français) de 344 mots, sélectionnés en fonction de leur pertinence linguistique et regroupés en paires contrastives selon leurs variations tonales. Chaque mot est enregistré de manière isolée par 11 locuteurs natifs du Yemba, totalisant environ 3 420 fichiers audio au format **.wav**, tous associés à des annotations détaillées.

Chaque fichier audio est accompagné d’un fichier **.TextGrid** (format compatible avec le logiciel Praat) indiquant la segmentation syllabique et la tonalité (haut, bas, moyen) de chaque syllabe. Les enregistrements ont été réalisés dans des environnements naturels (domicile, lieu de travail, campus), puis nettoyés et segmentés manuellement à l’aide d’Audacity pour isoler chaque mot de manière claire et exploitable.

Le corpus est structuré de manière hiérarchique à partir d’un répertoire principal nommé **audios**, contenant un dossier par locuteur **(speaker\_1, speaker\_2, …, speaker\_11).** Chaque dossier de locuteur est subdivisé en groupes **(group\_1, group\_2, …, group\_149)** représentant différentes séries de mots. À l’intérieur de chaque groupe se trouvent les fichiers audios **.wav** et leurs correspondants **.TextGrid**.

Un dossier **metadata** accompagne les données sonores et contient les informations linguistiques, tonales et syllabiques de chaque mot, permettant ainsi un traitement automatisé ou semi-automatisé pour les applications d’analyse ou d’apprentissage automatique.

1. Etat de l’art
   1. Approches existantes en ASR.

La reconnaissance automatique de la parole (ASR – *Automatic Speech Recognition*) vise à convertir un signal vocal en texte. Depuis les premières recherches dans les années 1950, les approches ASR ont connu une évolution rapide, passant de méthodes statistiques classiques à des architectures profondes entièrement neuronales. On distingue généralement trois grandes catégories d’approches : les méthodes basées sur des modèles probabilistes, les architectures hybrides HMM-DNN qui combine deux approches complémentaires pour la reconnaissance vocale à savoir : les modèles de Markov cachés (HMM) et les réseaux de neurones profonds (DNN), et les systèmes de bout-en-bout (*end-to-end*).

* 1. Les approches probabilistes classiques

Les premiers systèmes ASR reposaient principalement sur les modèles de Markov cachés (HMM) pour modéliser la séquence temporelle des phonèmes, combinés à des modèles de mélanges gaussiens (GMM) pour représenter les distributions acoustiques. Elle fonctionne comme suit :

 Avantages :

* Modélisation explicite des transitions temporelles.
* Facilité de décomposition en sous-problèmes (acoustique, lexique, langage).

 Inconvénients :

* Incapacité à capturer les dépendances longues.
* Sensibilité au bruit et à la variabilité des locuteurs.
* Performance limitée sur des langues à faible ressource ou à tons, comme le yemba.
  1. Systèmes hybrides HMM-DNN

Avec l’essor du deep learning, les GMM ont été progressivement remplacés par des réseaux de neurones profonds (DNN) qui améliorent la modélisation des unités acoustiques (phonèmes, sous-phonèmes). Ces systèmes hybrides conservent la structure HMM mais utilisent un DNN comme estimateur des probabilités d’émission. Techniques les plus courantes sont les DNN, CNN, RNN, LSTM, BiLSTM, etc. elles fonctionnent comme suit :

* **Extraction des caractéristiques acoustiques** (ex. MFCCs) à partir du signal audio.
* **Le DNN** prend ces caractéristiques en entrée et prédit la **probabilité de chaque état HMM** (appelés *senones*).
* Le **HMM** utilise ces probabilités pour modéliser la **structure séquentielle** de la parole (enchaînement des phonèmes).
* Le décodage final (souvent via un graphe WFST) permet de retrouver la **séquence de mots ou de syllabes** la plus probable.

Meilleure robustesse face aux variations acoustiques.

Amélioration substantielle de la performance par rapport aux GMM.

* 1. Approches de bout-en-bout (*End-to-End*)

Les approches modernes reposent sur des architectures neuronales capables de convertir directement un signal audio en texte, sans avoir besoin de modules séparés (prononciation, phonèmes, alignement forcé, etc.) comme les CTC (Connectionist Temporal Classification), les Attention-based Encoder-Decoder ou les Transducers (RNN-T, Conformer-Transducer). Leurs caracteristique principales reposent sur

* une **Architecture unifiée** : un seul réseau neuronal (souvent un modèle à base de RNN, Transformer ou Conformer) prend en entrée les caractéristiques acoustiques et produit directement la séquence de texte.
* **Moins de dépendances externes** : pas besoin de dictionnaire de prononciation ou de graphe HMM.
* **Apprentissage supervisé global** : le modèle apprend à prédire la transcription complète à partir de paires audio/texte.
* **Simplicité de conception** : un seul modèle à entraîner et à déployer.
* **Meilleure performance avec beaucoup de données** : surtout dans les langues bien dotées.
* **Facilité d’adaptation multilingue** : utile pour les langues peu dotées si on utilise du transfert d’apprentissage.

**⚠️ Limites dans les langues peu dotées**

* **Besoin de grandes quantités de données appariées** (audio + transcription).
* **Moins robuste aux variations de domaine** (accent, bruit, dialecte) sans adaptation spécifique.
* **Moins interprétable** que les systèmes modulaires.

1. Technologies adaptées aux langues peu dotées.
2. Modèles classiques vs architectures modernes (CTC, attention, RNN, GRU...).