Université Assane Seck de Ziguinchor

UFR Sciences et Technologies

Département Informatique

MEMOIRE DE FIN D'ÉTUDES

Pour l'obtention du diplôme de Master

Mention : Informatique

Spécialité : Génie Logiciel

Sujet :

Construction de Datasets : Vers un modèle de langage basé sur les langues locales sénégalaises (cas du Wolof, Sérère et du Pulaar)

Présenté par :

M. Boubacar Diallo

Soutenance le 02/08/2024

Membres du jury

- Pr. Youssou Dieng (**Président du jury**)
- Dr. El hadji Malick Ndoye (**Rapporteur**)
- Dr. Elodie Gauthier (**Examinatrice**)
- Pr. Abdoulaye Guissé (**Encadrant**)
- Pr. Ousmane Diallo (**Co-encadrant**)

Sous la direction de :

- Pr. Abdoulaye Guissé

Co-encadrant :

- Pr. Ousmane Diallo

Année Universitaire : 2022 – 2023

Dédicaces

À ma défunte Mère, Oumy Diallo, que ton âme repose en paix dans le paradis céleste.

À mon Père, Moussa Diallo, vous incarnez la force et la bienveillance d'un père dévoué.

À mon grand-père Mamadou Oury Barry, votre sagesse continue de guider mes pas.

Remerciements

- ✓ *Je rends grâce à Allah, le tout miséricordieux, le très miséricordieux.*
- ✓ *Je remercie le Pr. Abdoulaye Guissé pour sa disponibilité et son soutien pour ce travail.*
- ✓ *Je tiens à exprimer ma profonde gratitude à mon co-encadrant, le Pr. Ousmane Diallo, également pour sa disponibilité et son soutien. Ses efforts et son dévouement ont grandement contribué à mon parcours et je lui en suis profondément reconnaissant.*
- ✓ *Nous souhaitons adresser nos remerciements distingués au Pr. Youssou Dieng pour avoir accepté de présider mon jury de soutenance, ainsi qu'aux autres membres du jury, le Dr. El Hadji Malick Ndoye et Mme Elodie Gauthier, pour leur précieuse évaluation de mon travail.*
- ✓ *Je souhaite également exprimer ma gratitude à tous les enseignants du département d'Informatique de l'UASZ, pour la valeur et la qualité de leur enseignement, ainsi que pour les efforts déployés pour garantir une excellente formation à leurs étudiants.*
- ✓ *Je remercie Aminata Ndiaye Diallo de Jokalante, et à toute l'équipe de Jokalante pour leur soutien, leurs précieux conseils et leur expertise partagée.*
- ✓ *Je tiens à exprimer ma reconnaissance envers M. Ibrahima Ndao pour ses soutiens et ses suggestions qui ont enrichi ce travail.*
- ✓ *Je n'oublie pas mes camarades de classe pour leur collaboration et leur soutien tout au long de mon parcours académique.*
- ✓ *Enfin, un immense merci à ma famille pour leur soutien et leurs encouragements constants tout au long de cette aventure académique.*

Résumé

La diversité linguistique au Sénégal est confrontée à un obstacle majeur en raison du faible taux d'alphabétisation, avec 54,6% de la population ayant peu ou pas de compétences en lecture et écriture. Cette situation limite l'accès aux services numériques et à des secteurs vitaux comme la santé, l'éducation et l'agriculture. Pour pallier ce problème, le projet Kallaama mobilise des linguistes et des informaticiens pour créer des données audios transcrites et annotées, collecter des ressources textuelles et développer des dictionnaires de prononciation dans les principales langues sénégalaises (Wolof, Sérère, Pulaar). Ces données sont utilisées pour entraîner des systèmes de reconnaissance vocale, facilitant ainsi le développement d'agents conversationnels vocaux (voicebots, callbots). Kallaama est soutenu par l'entreprise Jokalante, qui souhaite offrir des services vocaux et conversationnels personnalisés en langue locale pour conseiller les petits producteurs et entreprises agricoles. Ce mémoire de fin d'études a contribué à la collecte de ressources textuelles en ligne et hors ligne, au prétraitement (nettoyage et normalisation) de ces données, puis à la construction de jeux de données textuels et de lexiques de prononciation pour les trois principales langues vernaculaires du Sénégal. Ces datasets sont utilisés à des fins d'apprentissage automatique (Machine Learning) et d'apprentissage profond (Deep Learning) en vue de créer des modèles de langage et de prononciation, avec pour finalité la mise en place d'agents conversationnels vocaux, utiles pour les populations peu ou pas lettrées.

Mots clés **:** Reconnaissance vocale, Langues vernaculaires, Agents conversationnels vocaux, Voicebots, Callbots, Apprentissage automatique, Apprentissage profond, Datasets, Dictionnaires de prononciation

Abstract

Linguistic diversity in Senegal faces a major obstacle due to the low literacy rate, with 54.6% of the population having little or no reading and writing skills. This situation limits access to digital services and vital sectors such as health, education and agriculture. To alleviate this problem, the Kallaama project mobilizes linguists and computer scientists to create transcribed and annotated audio data, collect textual resources and develop pronunciation dictionaries in Senegal's main languages (Wolof, Serer, Pulaar). These data are used to train speech recognition systems, facilitating the development of conversational voice agents (voicebots, callbots). Kallaama is supported by the Jokalante company, which aims to offer personalized voice and conversational services in the local language to advise small producers and agricultural businesses. This thesis contributed to the collection of online and offline text resources, the pre-processing (cleaning and normalization) of this data, and then the construction of text datasets and pronunciation lexicons for the three main vernacular languages of Senegal. These datasets are used for Machine Learning and Deep Learning purposes to create language and pronunciation models, with a view to setting up conversational voice agents, useful for populations with little or no literacy.

Keywords **:** Speech recognition, Vernacular languages, Conversational voice agents, Voicebots, Callbots, Machine learning, Deep learning, Datasets, Pronunciation dictionaries.

Table des matières

Liste des figures

Liste des Tableaux

Liste des abréviations

AED : Attention Encoder-Decoder

ALFFA : African Languages in the Field: Speech Fundamentals and Automation

API : Alphabets Phonétiques Internationaux

BERT : Bidirectional Encoder Representations from Transformers

CER : Character Error Rate

CNN : Convolutional Neural Network

CTC : Connectionist Temporal Classification

CSS : Cascading Style Sheets

DEC : Digital Emerging Countries

DNN : Deep Neural Network

E2E : End-to-End

EPT : Ecole Polytechnique de Thiès

G2P : Grapheme-to-Phoneme

GMM : Gaussian Mixture Model

GPT : Generative Pre-trained Transformer

HMM : Hidden Markov Model

HTML : HyperText Markup Language

IFAN : Institut Fondamental d'Afrique Noir

IPA : International Phonetic Alphabet

MEA : Middle-East and Africa

MITLM : Massachusetts Institute of Technology Language Modeling Toolkit

NLP : Natural Language Processing

ONG : Organisation Non Gouvernementale

PAS : Programme Algorithme et Solution

RASR : RWTH ASR (Automatic Speech Recognition)

RNN : Recurrent Neural Network

SAMPA : Speech Assessment Methods Phonetic Alphabet

SER : Syllable Error Rate

SaaS : Software as a Service

SRAP : Système de Reconnaissance de la Parole

SRILM : Stanford Research Institute Language Modeling Toolkit

TALN : Traitement Automatique du Langage Naturel

TTS : Text-to-Speech

UCAD : Université Cheikh Anta Diop

UASZ : Université Assane Seck de Ziguinchor

WER : Word Error Rate

WFSA : Weighted Finite-State Acceptor

WFST : Weighted Finite-State Transducer

WFT : Weighted Finite Transducer

INTRODUCTION GENERALE

Au cours des deux dernières décennies, les avancées significatives dans le domaine des Technologies de l'Information et de la Communication (TIC) ont été particulièrement marquées par le développement constant du traitement du langage humain. Cette évolution, notamment dans la reconnaissance automatique de la parole, a joué un rôle majeur dans la promotion et le développement des langues à faible ressources. Actuellement, la reconnaissance automatique de la parole est intégrée à de nombreuses applications variées. Elle est utilisée dans les systèmes d'apprentissage des langues pour améliorer la prononciation des apprenants[1], les applications téléphoniques telles que les serveurs vocaux pour l'accès aux services[2], et la recherche dans des bases de données vocales, particulièrement bénéfique pour les personnes à besoins spécifiques et les analphabètes, surtout dans les régions rurales [3] [4]. De plus, elle est utilisée dans les applications de transcription automatique des documents radio et télédiffusés. Cependant, il convient de noter que cette technologie présente une limite majeure : l'efficacité d'un moteur de reconnaissance vocale dépend largement de la disponibilité d'une quantité substantielle de données (écrites et orales) pour son entraînement.

Au Sénégal, le français est la langue officielle. Néanmoins, comme dans de nombreux pays africains, la langue officielle du pays n’est parlée que par une minorité de la population (moins de 25 % de la population sénégalaise). La majorité de la population sénégalaise parle et comprend le wolof, et à coté nous avons le pulaar ensuite le sérère. Même si le wolof est bien documenté et décrit dans les études de linguistique [5] [6] [7], la langue souffre encore d’un manque de données numériques. Ce déficit est également observé pour le pulaar et le sérère, qui sont respectivement moins documentés que le wolof et cela pose de réels problèmes en termes d’inclusion et d’accès à l’information numérique. Par ailleurs, 54,6% de la population sénégalaise est peu ou pas alphabétisée. Ainsi l'accès aux services digitaux, basés largement sur la modalité écrite, devient difficile. Cette situation entrave la participation de la société à l'information et limite l'accès à des services essentiels tels que la santé, l'éducation et l'agriculture. Des recherches récentes et le succès des services vocaux, indiquent que les services conversationnels vocaux en langues locales sont prometteurs pour améliorer l'accessibilité.

C'est dans ce contexte que le projet Kallaama a été lancé, mobilisant des linguistes et des informaticiens pour combler ces lacunes. L'initiative consiste à produire des audios transcrites et annotées, de collecter des corpus textuels dans les trois (3) principales langues sénégalaises qui sont le wolof, le sérère et le pulaar. Ces données seront utilisées pour entraîner des systèmes de reconnaissance vocale robustes, avec l'objectif de développer des services vocaux et conversationnels pour les personnes peu ou pas lettrées, notamment dans les activités agricoles touchant 55% de la population sénégalaise. L'initiative prévoit également la création d'un centre pérenne d'acquisition de données langagières à des fins de traitement automatique, en collaboration avec des partenaires tels qu'Orange Innovation, l'École Polytechnique de Thiès et l'Université Assane Seck de Ziguinchor qui en assurent la caution académique. Le consortium formé a pour objectif de construire des données linguistiques open-source pour le développement de technologies d'interaction vocale et conversationnelle, contribuant ainsi à des avancées significatives dans le domaine de la recherche linguistique.

Le stage a joué un rôle dans la construction de datasets (audios transcrites, corpus textuels et dictionnaire de prononciation) destinée à des fins d'apprentissage automatique, en vue de créer des modèles phonétiques, de langues et de prononciations. Ces modèles sont essentiels pour la construction des systèmes vocaux et conversationnels pour les langues vernaculaires du Sénégal.

Le plan du mémoire se déroule comme suit :

- dans le premier chapitre, le focus est mis sur le contexte du projet, débutant par la présentation détaillée de l'entreprise d'accueil, Jokalante, incluant ses services et son organigramme. Cette introduction à l'environnement professionnel est suivie d'une analyse approfondie du contexte et des enjeux du projet Kallaama, exposant les détails du projet, les membres du consortium impliqués dans le projet, et le bailleur de fonds ;
- le deuxième chapitre concentre sur la modélisation de la parole, revisitant les principes généraux tels que le modèle acoustique, le modèle de langue, et le dictionnaire de prononciation. Ce volet est enrichi d'une revue littéraire sur les défis actuels et les approches récentes des systèmes de reconnaissance automatique de la

parole (SRAP) à faible ressources, tout en mettant en lumière les ressources nécessaires pour la mise en place effective d'un SRAP ;

- le troisième chapitre présente nos réalisations techniques dans le projet Kallaama. Il débute par la définition de la méthodologie, mettant en avant le workflow, suivi de la discussion sur la collecte de données, le prétraitement des données, la construction d'un dictionnaire de prononciation, et enfin, le dépôt des données. Cette approche méthodique permettra une compréhension détaillée des étapes concrètes de la mise en œuvre du projet ;
- le quatrième et dernier chapitre explore les perspectives et rétrospectives du projet Kallaama. Il met en avant les futurs impacts sur les langues vernaculaires et le développement de services vocaux inclusifs, il fournit également une rétrospective critique évaluant les contributions du projet aux recherches académiques linguistiques. En soulignant le rôle important du consortium et du financement dans la réalisation du projet, ce chapitre clôture le mémoire en offrant une vision complète et réfléchie du chemin parcouru par le projet Kallaama.

Chapitre 1 : CONTEXTE DU STAGE

Introduction

Ce chapitre explore le contexte du projet débutant par une présentation de l'entreprise d'accueil, Jokalante, une entreprise sociale dédiée à l'utilisation des technologies pour diffuser des informations aux populations rurales. Ensuite, nous examinons la diversité linguistique du Sénégal et les défis d'alphabétisation, mettant en lumière l'importance de développer des solutions technologiques adaptées. Nous nous penchons ensuite sur le projet coopératif Kallaama, ses objectifs, ses partenaires et son financement.

1.1 Présentation de l'entreprise d'accueil

Jokalante est une entreprise sociale sénégalaise de 10 personnes créée en 2016 qui se consacre à l'utilisation des nouvelles technologies pour diffuser des prévisions climatiques et météorologiques. Elle complète ces informations par des conseils agricoles et nutritionnels délivrés vocalement et dans des langues locales, visant ainsi à mieux informer les populations rurales pour faciliter leurs prises de décision. L'entreprise s'engage également dans l'accompagnement des femmes afin de favoriser leur accès au crédit dans le secteur agricole. En effet, Jokalante produit des contenus ciblant les petits producteurs, les pêcheurs et les éleveurs, qui subissent les premiers les effets des changements climatiques et en particulier les femmes, qui représentent 70% de la main-d'œuvre agricole au Sénégal. L'accès à des conseils agricoles adaptés et aux ressources financières reste une préoccupation quotidienne pour les populations spécifiquement visées.

1.1.1 Produits et services proposés par Jokalante

Jokalante a développé une plateforme SaaS[1] (Software as a Service) permettant aux organisations de diffuser des informations adaptées aux profils de leurs membres, incluant des données telles que leur identité, leur localité, leur langue parlée, et leur mode de communication privilégié. Ces contenus variés sont diffusés grâce à un système de gestion

[1] https://tic.jokalante.com/

d'appels téléphoniques interactif (m-alert) qui facilite les échanges entre les membres et leurs organisations en personnalisant les informations en fonction de leurs besoins. Étant donné que les populations ciblées sont largement illettrées, l'utilisation de la langue locale à l'oral est essentielle. Dans le cadre de son engagement en faveur de l'agriculture durable et de l'autonomisation des communautés, Jokalante offre une palette diversifiée de services novateurs. Cela englobe des prévisions climatiques précises pour permettre aux agriculteurs d'anticiper les conditions atmosphériques, des conseils agro météorologiques basés sur des données météorologiques pour optimiser les pratiques agricoles, et des conseils en agro-nutrition adaptés aux cultures locales pour améliorer la qualité des récoltes et la santé des communautés agricoles. Pour assurer une accessibilité maximale à ces informations, Jokalante utilise divers canaux tels que des messages vocaux, un CALL Center, USSD, SMS, et d'autres services. Un solide mécanisme de feedback garantit une amélioration continue des services en fonction des besoins changeants des communautés agricoles. En outre, l'entreprise utilise des outils avancés pour analyser les données météorologiques et agricoles, fournissant des rapports détaillés pour une meilleure compréhension des conditions locales. Grâce à des solutions numériques innovantes, Jokalante collecte des données pertinentes, renforçant ainsi son engagement constant envers l'amélioration continue de ses services.

Jokalante a produit des centaines d’heures de contenus audios entre les émissions radios, les messages vocaux en langues locales et les messages reçus des producteurs et dispose de l’entièreté des droits sur ces contenus. Cependant ces contenus ne sont pas encore codifiés ni exploitables à des fins technologiques, en particulier pour de la reconnaissance de parole, mais sont juste traduits du « script » français vers les langues gérées par Jokalante : pulaar, wolof, diola, mandinké et sérère. Jokalante a commencé en 2020, sa montée en compétence sur la transcription de fichiers audios en Wolof, langue pour laquelle 10h ont été transcrites selon les règles de l’art pour les besoins d’Orange Innovation dans le cadre de ses travaux de recherche sur les langues subsahariennes.

Jokalante travaille avec une base de plus de 230 000 personnes dont 40 000 sont des femmes (l’une des contraintes est que les personnes doivent disposer d’un mobile personnel).

1.1.2 Organisation et organigramme de Jokalante

L'organisation de Jokalante est dirigée par un Conseil d'Administration, supervisant les opérations de l'entreprise. Sous la direction du Conseil d'Administration, se trouve la Directrice Générale, chargée de la gestion quotidienne et du leadership de l'entreprise. La structure hiérarchique comprend également des postes clés tels que le Responsable Administratif et Financier (RAF), le Responsable Informatique et Sécurité (RIS), le Responsable Marketing et Communication (RMC), ainsi que le Responsable des Ressources Humaines (RH). Chaque responsable occupe un rôle stratégique dans le bon fonctionnement et le développement de l'entreprise Jokalante.

La figure (1) ci-dessous illustre l'organigramme de l'entreprise.

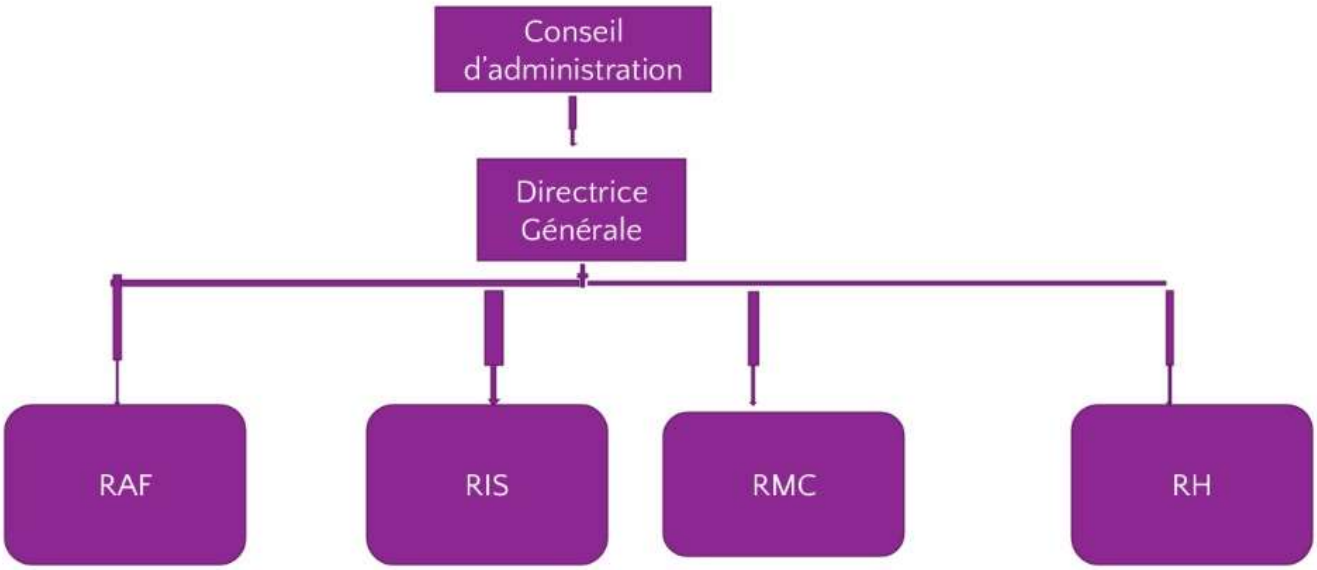

Figure 1 : Organigramme de Jokalante

1.2 Contexte et enjeux

1.2.1 Introduction du projet coopératif Kallaama

1.2.1.1 Problématique et objectif du projet

Le Sénégal est caractérisé par une riche diversité linguistique et la reconnaissance des langues nationales est mentionnée dès l'article premier de la constitution du 22 janvier 2001 « La langue officielle de la République du Sénégal est le français ». Les langues nationales sont le diola, le malinké, le pulaar, le sérère, le soninké, le wolof et toute autre langue nationale qui sera codifiée ». Cependant, comme dans de nombreux pays africains, la langue officielle n'est parlée que par une minorité de la population, représentant moins de 25 % des Sénégalais. La langue principale du pays est le wolof, largement utilisé et compris par la majorité de la population. D'autres langues importantes incluent le pulaar et le sérère.

La figure 2 illustre la diversité des langues parlées au Sénégal.

Figure 2 : diversité linguistique du Sénégal

Cependant, le wolof est parlé par 76,6 % de la population sénégalaise, jouant ainsi un rôle de langue véhiculaire à travers l'ensemble du pays. À côté du wolof, on retrouve par ordre d'importance, le pulaar et le sérère avec respectivement 26 % et 13,5 % de locuteurs en 2013[8].

La figure 2 illustre l'évolution des langues parlées au Sénégal entre 1988 et 2013

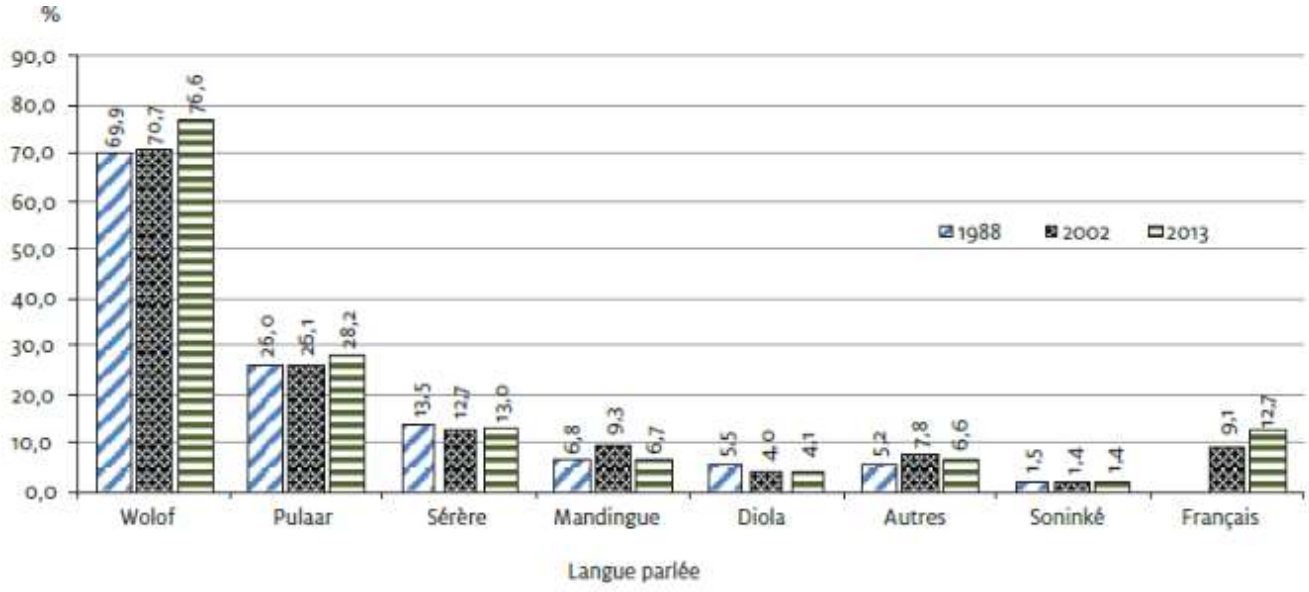

Figure 3 : Evolution des langues parlées au Sénégal [8]

Bien que le wolof soit bien documenté et étudié dans les recherches linguistiques, il souffre encore d'un manque de données numériques. Ce déficit est également présent pour le pulaar et le sérère, qui sont moins documentés que le wolof. Cette situation engendre des problèmes significatifs en matière d'inclusion et d'accès à l'information numérique.

Par ailleurs, malgré cette richesse linguistique, l'alphabétisation pose un défi majeur dans le pays. Selon la Direction de l'Alphabétisation et des Langues nationales, 54,6% d'analphabètes [2]sont recensés au Sénégal et donc en difficulté dans l'utilisation des services digitaux pour lesquels la modalité écrite est majoritaire. Cela limite leur participation à la société de l'information et restreint leur accès à des services essentiels, tels que l'information sur la santé, l'éducation, l'agriculture, et bien d'autres. Des travaux de recherche[9] et le succès de la communication vocale sous WhatsApp montrent que le développement de services conversationnels vocaux en langues locales constitue une piste aussi crédible que prometteuse pour l'accessibilité aux services. Pour avancer dans cette voie, il est nécessaire de développer des systèmes de reconnaissance de parole robustes sur ces langues.

[2] https://www.education.sn/fr/article/281

Alors que les technologies de reconnaissance automatique de la parole sont bien établies pour le français ou l'anglais, les langues vernaculaires du Sénégal, considérées comme peu dotées sur le plan informatique, manquent quasiment de solutions matures. Il est évident que les langues sénégalaises font face à un déficit notable en termes de ressources, que ce soit au niveau audio ou textuel, entravant ainsi l'application de ces technologies. Le projet Kallaama a pour objectif de générer 60 heures de données audio transcrites et annotées, des corpus textuels et des dictionnaires de prononciation afin de former des systèmes de reconnaissance de la parole dans trois des principales langues nationales du Sénégal : wolof, sérère et pulaar.

Le choix de ces trois langues est déterminé par le nombre significatif de locuteurs dans le pays. En effet, il s'agit des trois langues vernaculaires les plus largement représentées au Sénégal. Cette sélection est d'autant plus pertinente étant donné que ces langues traversent quelques frontières, ce qui renforce leur importance et leur impact dans la région.

1.2.2 Les membres du consortium

Kallaama est destiné à construire et à pérenniser un consortium fort entre Jokalante, Orange Innovation, l'EPT et l'UASZ.

L'engagement d'Orange Innovation dans le projet Kallaama s'inscrit dans le cadre du domaine DEC (Digital Emerging Countries). Établi en 2011, ce domaine de recherche est entièrement dévoué à soutenir l'ambition numérique d'Orange Middle-East and Africa (MEA). Son objectif est de concevoir des solutions numériques adaptées aux défis spécifiques des pays en développement, permettant ainsi d'offrir une connectivité abordable à un large public. De plus, il cherche à développer des services numériques répondant aux besoins locaux et compatibles avec les infrastructures existantes.

Sur le plan académique, l'UASZ garantit la validité académique du projet, tandis que l'EPT et Orange apportent leur savoir-faire dans la supervision de la production des données et la diffusion des résultats de recherche. De son côté, Jokalante contribue en fournissant des outils technologiques. Ces partenariats renforcent les liens entre des entreprises telles que Jokalante, Orange Innovation et le milieu universitaire, tout en assurant une rigueur dans le contrôle qualité.

1.2.3 Le bailleur de fonds et l'objectif de la subvention

Le projet Kallaama est financé par Lacuna Fund[3], une organisation qui se concentre sur le financement et le soutien de projets intégrant du traitement du langage naturel dans le but de remédier aux lacunes existantes dans la diversité des données linguistiques. Le Lacuna Fund a débuté en tant que bailleur de fonds participatif réunissant la Fondation Rockefeller, Google.org et le Centre de recherche pour le développement international du Canada, mais il a ensuite évolué pour devenir un engagement multipartite soutenu par un éventail d'organismes actifs dans le développement, la philanthropie et la recherche.

Le Lacuna Fund met à disposition des scientifiques spécialisés dans les données, des chercheurs et des entrepreneurs sociaux travaillant dans des contextes à revenus faibles et moyens à l'échelle mondiale, les ressources nécessaires. Ces ressources visent à les aider à produire de nouveaux ensembles de données pour répondre aux besoins d'une population mal desservie, à résoudre des problèmes non résolus, à étendre les ensembles de données existants pour plus de représentativité, ou à actualiser d'anciens ensembles de données pour les rendre plus durables.

Guidé par des professionnels de l'apprentissage machine dans le monde entier, le Lacuna Fund est conçu pour et par les communautés qu'il entend servir. Tous les ensembles de données générés sont développés et détenus localement et mis ouvertement à la disposition de la communauté internationale tout en respectant les meilleures pratiques en matière d'éthique et de respect de la vie privée.

1.3 Présentation du stage

Dans le cadre de ce stage de quatre mois, la mission englobe diverses tâches, mettant particulièrement l'accent sur les langues locales du Sénégal, notamment le wolof, le pulaar et le sérère. L'objectif principal consiste à constituer des datasets en rassemblant des ressources textuelles et audios, traitées par des linguistes et informaticiens, qui seront exploitées pour le développement de SRAP. Ce processus englobe la transcription des enregistrements audio pour chaque langue, l'extraction de données textuelles du web et éventuellement à partir d'ouvrages disponibles dans les bibliothèques. Après la transcription des enregistrements

[3] https://lacunafund.org/fr/

audio, ces derniers sont vérifiés par un expert pour chaque langue avant d'être validés. Quant aux données extraites, elles sont soumises à un processus de nettoyage afin d'assurer leur qualité et leur cohérence. À partir de ces données récupérées et nettoyées, des corpus de textes sont construits, servant de base à la création de modèles de langue. Une composante significative de la mission consiste également à élaborer un lexique de prononciation pour chaque langue. Ce lexique, fonctionnant comme un dictionnaire, associe chaque mot à une ou plusieurs prononciations, représentées par des symboles décrivant les sons (phonèmes). Enfin, une étape du processus implique le dépôt des données dans des hébergeurs, assurant ainsi leur accessibilité et leur disponibilité pour d'autres chercheurs et développeurs intéressés par l'amélioration des études sur les langues locales du Sénégal.

1.3.1 Problématique du stage

La principale problématique de ce stage réside dans le prétraitement des données recueillies. Bien que ce processus soit primordial, il s'avère chronophage, posant ainsi des défis supplémentaires pour respecter les échéances du stage. La validation des données, en particulier, requiert une attention particulière afin d'assurer leur pertinence et leur fiabilité. La recherche de sites pertinents pour recueillir des données représente un défi pour le pulaar et le sérère en raison de leur disponibilité limitée sur le web. De plus, l'accès aux bibliothèques universitaires s'est avéré difficile pendant le stage, impactant la disponibilité des ressources papiers.

1.3.2 Encadrement

Dans le cadre de ce travail de mémoire, nous avons adopté une approche pluridisciplinaire, combinant les domaines de l'informatique et de la linguistique. Le co-encadrement est assuré par des membres éminents du consortium du projet Kallaama, alliant l'expertise académique et industrielle. Du côté académique, nous bénéficions de la collaboration de chercheurs renommés tels que Pr Abdoulaye Guissé, enseignant-chercheur à l'EPT de Thiès, ainsi que Pr Ousmane Diallo, enseignants-chercheurs à l’UASZ, apportant leur expertise en informatique et linguistique. Côté industriel, nous sommes encadrés par Elodie Gauthier, chercheuse à Orange Innovation en Traitement Automatique de la Parole, spécialisée dans l'application aux langues peu dotées, et Aminata Ndiaye Diallo, responsable du contrôle

interne chez Jokalante, qui enrichissent notre démarche par leur expérience pratique et leur compréhension des besoins industriels. Cette collaboration diversifiée garantit une approche complète, équilibrée et pertinente pour la réalisation du projet.

1.3.3 Matériels à disposition

Le stage bénéficie une infrastructure informatique complète, avec des équipements essentiels pour la collecte de données, le prétraitement et le stockage de celles-ci. En effet, Jokalante a mis à notre disposition un environnement propice en mettant à notre disposition des ordinateurs de bureau au sein de leur siège, ainsi que des ordinateurs portables pour le télétravail. Des dispositifs de stockage tels que des clés USB et des disques durs sont également disponibles, tout comme une connexion Internet. Cette mise à disposition d'équipements variés assure un accès pratique et efficace aux outils nécessaires à la réalisation des tâches liées au stage.

Conclusion

En conclusion de ce premier chapitre, nous avons parlé du contexte dans lequel s'inscrit notre projet, mettant en lumière les défis de l'alphabétisation et de l'accès à l'information au Sénégal. À travers l'entreprise sociale Jokalante et le projet Kallaama, nous avons exploré les efforts déployés pour combler ces lacunes, avec un focus particulier sur l'utilisation des langues locales pour améliorer l'accessibilité numérique. Dans le prochain chapitre, nous aborderons la modélisation de la parole, élément central pour le développement des systèmes de reconnaissance vocale.

Chapitre 2 : MODELISATION DE LA PAROLE

Introduction

Les systèmes de reconnaissance automatique de la parole (SRAP) ont pour objectif d'automatiser la transcription d'un message oral en texte[10]. Les systèmes captent les sons produits par la voix humaine à l'aide d'un microphone, puis analysent ces sons vocaux pour les transformer en mots écrits. Cette technologie peut notamment être utilisée pour des applications de dictée vocale, pour la commande vocale, pour la transcription automatisée ou encore pour l'interaction verbale avec des assistants virtuels. La reconnaissance vocale représente le principal verrou technologique à lever pour développer des services vocaux (voicebot, callbot etc.). Les SRAP peuvent être classés en deux (2) catégories : les systèmes traditionnels[4] et les systèmes de bout en bout (end to end - E2E).

Dans ce chapitre, nous passons en revue les SRAP traditionnels et les SRAP de bout-en-bout, puis discutons de l'évaluation des performances et des outils existants pour la reconnaissance vocale. Nous abordons également les défis et travaux récents sur les SRAP pour les langues peu-dotées et l'importance des données dans ce domaine.

2.1 Principe générale des SRAP traditionnels

Dans un SRAP de type HMM-GMM/DNN, l'objectif consiste à déterminer la séquence de mots la plus probable à partir d'une séquence de paramètres acoustiques. Ce type de système repose sur un modèle acoustique, un lexique phonétisé (modèle de prononciation) et un modèle de langue qui évalue la probabilité d'une séquence de mots. Dans cette architecture, la parole est supposée être générée par un modèle de langage qui fournit des estimations des probabilités pour une séquence de mots, le modèle de prononciations génère des séquences de phonèmes associées aux mots, et le modèle acoustique qui code le message dans le signal sonore.

Hidden Markov Model (HMM), Gaussian Mixture Model (GMM) et Deep Neural Network (DNN).

[4] On entend par « traditionnel » un système construit à partir d'une architecture hybride GMM/HMM ou DNN/HMM, en comparaison aux architectures de bout-en-bout (*end-to-end*) qui sont entièrement neuronales.

Le processus de reconnaissance de la parole traditionnelle peut être divisé en deux étapes : la paramétrisation et le décodage, illustrés sur la figure 3.

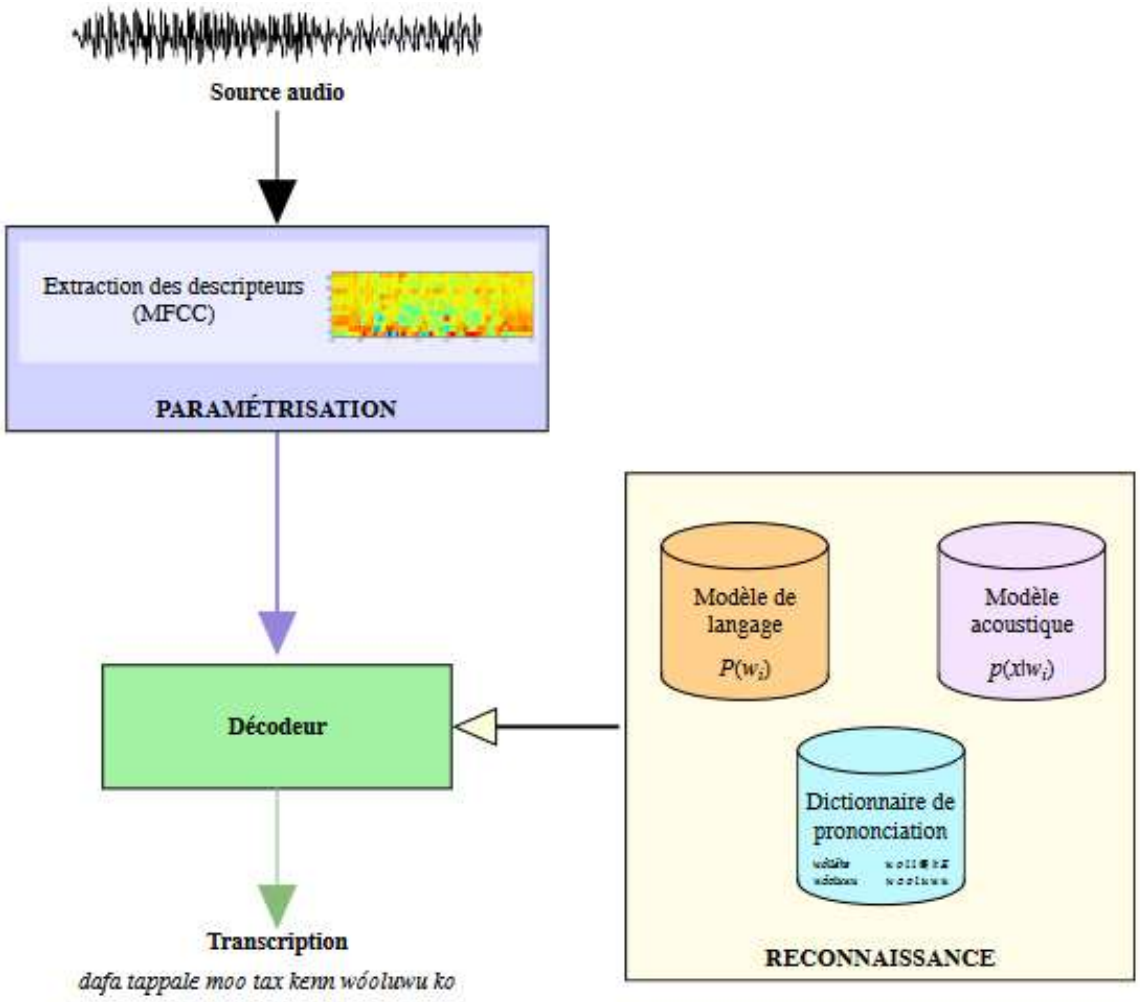

Figure 4 : Architecture d'un système de reconnaissance de la parole [11]

La paramétrisation du signal de parole consiste à transformer le signale acoustique du domaine temporel vers le domaine spectral (fréquentiel) afin d'en extraire les paramètres pertinents [3]. Le décodeur est le moteur du système de reconnaissance vocale qui découvre les séquences de mots possibles à partir des informations pertinentes en utilisant les connaissances des modèles acoustiques, de prononciation et de langue [11].

D'un point de vue linguistique, dans un SRAP, ces modèles ont des représentations spécifiques dans une langue. Le modèle acoustique s'attache à la phonologie de la langue,

c'est-à-dire à la manière dont les sons sont organisés et perçus. Le modèle de prononciation, quant à lui, se concentre sur le vocabulaire et les différentes façons de prononcer les mots. Enfin, le modèle linguistique s'intéresse à la grammaire de la langue, englobant les règles et structures qui régissent la formation de phrases.

Ces trois modèles travaillent en concert pour permettre une reconnaissance précise de la parole, couvrant les aspects acoustiques, lexicaux et grammaticaux de la langue considérée. Dans les sections suivantes, nous présentons de manière approfondie ces trois modèles.

2.1.1 Le décodeur

Dans la reconnaissance vocale, le décodeur est le composant qui découvre les séquences de mots à partir du signal vocal ou plus précisément des vecteurs de caractéristiques. La recherche de la séquence de mots la plus probable peut être réalisée en maximisant la probabilité postérieure pour les vecteurs de caractéristiques donnés. Il est difficile de calculer efficacement et solidement la probabilité postérieure. Par conséquent, au lieu de calculer directement la probabilité postérieure, on peut l'exprimer sous une autre forme en utilisant le théorème de Bayes (équation 1) :

$$W^* = argmax_w P(W/O)$$

$$= argmax_w \frac{P(W)P(O/W)}{P(O)} \qquad (1)$$

$$W^* = argmax_w P(W)P(O/W)$$

où W est la séquence de mots $w_1, w_2, \dots, w_m$ qui donne la probabilité postérieure maximale $P(W/O)$ étant donné O, une série d'observations $O_1, O_2, \dots$ sur lesquelles est produite la séquence de mots. Cela signifie que la meilleure séquence de mots peut être trouvée en combinant la probabilité linguistique de la séquence de mots $P(W)$ (probabilité a priori) avec la probabilité acoustique de la séquence de mots $P(W/O)$ (probabilité conditionnelle) à partir d'un modèle acoustique qui donne la valeur la plus élevée.

2.1.2 Le modèle acoustique

Le rôle principal du modèle acoustique est de générer, à partir de paramètres acoustiques, des hypothèses phonétiques associées à une probabilité pour chaque unité de segment de parole[12]. La construction d'un modèle acoustique robuste est l'un des principaux défis dans le domaine de la reconnaissance vocale. La difficulté de modéliser les caractéristiques acoustiques de manière robuste est due à la variabilité qui existe dans la parole. La variabilité du contexte peut se produire au niveau de la phrase, du mot et de la phonétique[13].

Il existe de nombreuses approches possibles pour modéliser les unités acoustiques, par exemple le modèle de Markov caché (HMM), le réseau neuronal artificiel, etc... [14]. Les modèles hybrides qui combinent des réseaux de neurones avec des modèles de Markov cachés sont largement adoptés pour modéliser les sons d'une langue et générer la séquence de mots la plus probable. Pour les réseaux de neurones, nous pouvons citer par exemple, des modèles entraînés à partir de réseaux de neurones convolutifs (communément abrégé « CNN » pour Convolution Neural Networks)[15], de réseaux de neurones récurrents (« RNN » pour Recurrent Neural Networks, « LSTM » pour Long Short-Term Memory)[16] etc. Avant l'incorporation des réseaux de neurones, la modélisation acoustique reposait sur le calcul d'un mélange de gaussiennes, connu en anglais sous le nom de Gaussian Mixture Model (GMM)[17].

Toutefois, dans le cadre du stage, l'accent était mis spécifiquement sur la création de ressources textuelles pour l'apprentissage de modèles de langue et l'apprentissage de modèles phonétiques, et donc, je ne détaillerai pas d’avantage la partie relative aux modèles acoustiques.

2.1.3 Le modèle de prononciation

La modélisation de la prononciation consiste à créer des modèles de mots ou de syllabes plus grands en utilisant les unités acoustiques définies dans le modèle acoustique. Un modèle de prononciation peut être créé à la main, c'est-à-dire en rédigeant des transcriptions de prononciation à partir d'une liste de mots. Étant donné que les phonèmes sont utilisés comme unités de modélisation acoustique dans le SRAP, la transcription de la prononciation peut

être obtenue directement à partir d'un dictionnaire standard qui contient des descriptions sur la manière dont les mots doivent être prononcés [4].

Cependant, tous les dictionnaires ne disposent pas de ces informations, en particulier les dictionnaires de langues peu documentées. Par conséquent, dans ce cas, un linguiste est nécessaire pour aider à convertir les graphèmes en phonèmes correspondants en utilisant les Alphabets Phonétiques Internationaux (API). Cependant, tous les symboles API ne sont pas lisibles par un ordinateur. Les phonéticiens ont donc créé SAMPA (Speech Assessment Methods Phonetic Alphabet), un alphabet phonétique standard lisible par une machine[18], conçu pour traiter uniquement les langues de l'Union européenne ; il a donc été étendu à X-SAMPA (Extended-SAMPA), qui peut être appliqué à toutes les langues[19]. Un autre alphabet phonétique lisible par ordinateur est appelé Arpabet pour décrire les phonèmes de l'anglais américain. Par exemple, l'ensemble de phonèmes utilisé dans le dictionnaire de prononciation de la CMU[5] est basé sur les symboles Arpabet. Il existe actuellement plusieurs techniques pour développer des systèmes de conversion graphème-phonème (G2P) afin de générer automatiquement des transcriptions de la prononciation.

Dans les cas où il n'existe pas de règles pour convertir les graphèmes en phonèmes et où la compréhension de la langue est limitée, des études ont montré que l'utilisation des graphèmes (dépendants du contexte) comme unités acoustiques pour modéliser la prononciation peut produire des performances de reconnaissance vocale acceptables, légèrement inférieures à celles des mots modélisés en utilisant des phonèmes[20]. À noter que cela signifie également que les unités de graphèmes doivent être entraînées dans le modèle acoustique.

En utilisant un modèle acoustique basé sur les phonèmes, les mots peuvent être modélisés dans le dictionnaire de prononciation comme suit :

comme [k ɔ m]

regardé [ʁ ə g a ʁ d e]

Pour augmenter la flexibilité de la prononciation et tenir compte des variations d'élocution, des variantes phonétiques peuvent être intégrées, comme illustré ci-dessous :

finalement (1) [f i n a l m ɑ̃]

finalement (2) [f i n a l ø m ɑ̃]

[5] Source : http://www.speech.cs.cmu.edu/cgi-bin/cmudict

De manière similaire, les liaisons entre les mots peuvent être ajoutées, permettant de modéliser à la fois « ils ont » et « ils iront », comme démontré dans l'exemple suivant :

ils_ont [i l z ɔ̃]

ils_iront [i l z i ʁ ɔ̃]

Par ailleurs, il peut être pertinent, dans le dictionnaire de prononciation, de modéliser des enchaînements de mots courts ou longs qui sont reconnus comme des mots composés dans le vocabulaire et dans le dictionnaire de prononciation[21] :

il y a [i l i j a]

pomme de terre [p ɔ m d ə t ɛ ʁ]

au fur et à mesure [o f y ʁ e a m ə z y ʁ]

2.1.3.1 La boîte à outils Grapheme-to-Phoneme (G2P) : Phonetisaurus

Une boîte à outils G2P open source appelée Phonetisaurus[6] a été développée dans le travail [22] qui a utilisé le cadre WFST[7] dans l'architecture du système. Pour entraîner un modèle G2P, la boîte à outils a besoin d'un dictionnaire de prononciation de base. La première étape consiste à aligner les séquences de graphèmes et de phonèmes dans le dictionnaire. Ensuite, la séquence de paires graphème-phonème est obtenue en concaténant les séquences de graphèmes et de phonèmes. Voici un exemple de séquence de paires graphème-phonème alignées pour le mot wolof " alxames " produit par Phonetisaurus :

Alxames → a l x a m ɛ s

Le modèle de prononciation est ensuite construit en entraînant un modèle N-gram à l'aide de l'ensemble des séquences conjointes acquises. Dans ce cas, des outils de modélisation du langage tels que SRILM [23] ou MITLM [24][8] peuvent être utilisés. Le modèle de sortie est ensuite converti en accepteurs d'états finis pondérés, qui sont ensuite convertis en WFT

[6] https://github.com/AdolfVonKleist/Phonetisaurus

[7] WFST : Weighted Finite State Transducer, désigne un transducteur fini pondéré, un modèle utilisé en TALN pour représenter des séquences de symboles avec des pondérations

[8] SRILM (Stanford Research Institute Language Modeling Toolkit) et MITLM (Massachusetts Institute of Technology Language Modeling Toolkit), sont des outils utilisés pour la création et l'application de modèles de langage statistiques

(Weighted Finite Transducer) pour obtenir les étiquettes d'entrée (graphème) et de sortie (phonème).

Il a été démontré que Phonetisaurus ainsi que Sequitur G2P4, un autre convertisseur G2P basé sur un modèle N-gram conjoint [25], ont surpassé les autres méthodes en termes de précision des phonèmes. En outre, la performance de cette boîte à outils pour une tâche G2P a été mesurée et comparée à d'autres techniques de modélisation G2P[26].

Nous utiliserons cette boîte à outils pour créer des systèmes G2P en raison de sa capacité à former des modèles en un temps raisonnable.

2.1.4 Le modèle de langue

Un SRAP dépend fortement de la connaissance linguistique de la parole. Les meilleurs systèmes de décodage acoustico-phonétique qui n'utilisent aucun modèle de langage n'atteignent qu'un taux d'exactitude en phonèmes de l'ordre de 50% environ[27]. C'est pourquoi l'inclusion de modèles de langage dans le système est essentielle pour déterminer la forme lexicale correspondante, c'est-à-dire la séquence de mots la plus probable du point de vue linguistique. Par exemple, la séquence "je suis ici" est plus probable du point de vue linguistique que "jeu suis ici" ou encore "jeux suit y si", même si l'aspect acoustique est presque similaire. Pour une séquence de phonèmes donnée, il peut y avoir plusieurs centaines de phrases possibles, et le rôle principal du modèle de langage est de les ordonner en fonction de leur plausibilité linguistique.

Généralement, un modèle de langage définit les phrases ou suites de mots, que le SRAP peut reconnaitre. Un modèle de langage est développé en se basant sur un corpus textuel, comprenant notamment les transcriptions du corpus d'apprentissage qui ont été utilisées pour créer le modèle acoustique. Il existe diverses techniques de modélisation, dont les modèles probabilistes traditionnels (les modèles n-grammes) qui ont été largement utilisés, et les modèles basés sur des réseaux de neurones, de plus en plus préférés en raison de leurs performances supérieures par rapport aux modèles classiques[28].

2.1.4.1 Les modèles de langue probabiliste

Les modèles de langue couramment employés dans un SRAP sont des modèles statistiques. Ils sont construits en utilisant des lois de probabilités conditionnelles, en estimant la probabilité maximale qu'une séquence de mots apparait effectivement dans la langue.

En effet, il revient au modèle de langage d'estimer la probabilité d'apparition d'une séquence complète de mots $(w_1, w_2, \dots, w_m)$ présente dans un vocabulaire (lexique). Plus précisément, le modèle de langage évalue la probabilité d'un mot w_m en tenant compte de tous les mots qui le précèdent $(w_1, w_2, \dots, w_{m-1})$. La probabilité d'apparition de la séquence de mots $(w_1, w_2, \dots, w_m)$ est estimée selon l'équation(2).

$$P(w_1, w_2, \dots, w_m) = \prod_{i..m} P(w_i | w_1, w_2, \dots, w_{i-1}) \qquad (2)$$

$$P(w) = \prod_{i..m} P(w_i | h_i)$$

Ou $w = w_1, w_2, \dots, w_m$ est les séquences de mot, $h_i = w_1, w_2, \dots, w_{i-1}$ l'historique du mot w_i et $P(w_i|h_i)$ est la probabilité du mot w_i, sachant tous les mots précédents.

En pratique, au fur et à mesure que la séquence de mots h_i s'enrichit, une estimation des valeurs des probabilités conditionnelles $P(w_i|h_i)$ devient de plus en plus difficile complexe.

Afin de réduire la complexité du modèle de langage, et par conséquent de son apprentissage, l'approche n-grammes peut être utilisée. Dans ce contexte, le terme "gramme" représente une unité lexicale, qui, dans notre cas, est un mot (bien que cela puisse également être un caractère ou une syllabe). L'indice n reflète la longueur de la séquence de mots à modéliser. Par exemple, un modèle 5-grammes signifie que le modèle peut estimer la probabilité d'apparition d'un mot en se basant sur la séquence de 1 à 4 mots qui le précède[17]. Le principe est donc le même, seul l'historique est limité aux n-1 mots précédents. La probabilité est donc approximée selon l'équation (3) :

$$P(w) = \prod_{i..m} P\left(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-n}\right) \qquad (3)$$

L'estimation de probabilité du mot wi sachant son histoire réduite consiste à compter le nombre d'occurrences C(wi) des n-grammes dans un corpus d'apprentissage. Cette estimation est évaluée selon un vocabulaire de référence V. Selon l'équation (4), nous avons :

$$P\left(w_n | w_1, \dots, w_{n-1}\right) = \frac{C\left(w_1, w_2, \dots, w_n\right)}{C\left(w_1, \dots, w_{n-1}\right)} \qquad (4)$$

Cette estimation devient complexe lorsque la taille du vocabulaire V augmente, car même un modèle n-gramme (n>3) nécessite une quantité considérable de données textuelles d'apprentissage pour estimer efficacement le modèle. Ce défi revêt une importance particulière dans notre domaine de travail axé sur les langues peu dotées particulièrement les locales, où l'obtention d'un grand corpus de textes s'avère souvent difficile.

Pour pallier le manque de données d'apprentissage, plusieurs techniques peuvent être appliquées, notamment les modèles n-classes, les modèles cache, et la méthode de Good-Turing pour le lissage avec le repli de Katz [29].

2.1.4.2 Les modèles de langue basés sur les réseaux neuronaux

Malgré leur simplicité et leur efficacité, les modèles n-grammes sont de moins en moins privilégiés au profit des modèles basés sur les réseaux de neurones. Tout comme les modèles n-grammes, l'objectif des modèles de réseaux de neurones est d'estimer la probabilité d'une séquence de mots. Dans les publications récentes, divers types de réseaux sont appliqués à l'apprentissage linguistique. On peut citer entre autres, les Réseaux de Neurones Profonds (DNN)[30], les Réseaux de Neurones Récurrents (RNN)[31], les Réseaux de Neurones à Mémoire à Long Terme (LSTM)[32] ainsi que les Réseaux de Neurones Convolutifs (CNN)[33]. Nous assistons également l'émergence récente de l'utilisation des modèles basées sur une nouvelle architecture de réseau simple : le Transformer. Sorti en décembre 2017, le Transformer [34] représente la dernière grande révolution technologique dans le domaine du

traitement automatique du langage. Cette architecture novatrice se compose d'un encodeur et d'un décodeur, chacun jouant un rôle dans le processus. L'encodeur est chargé de traiter l'entrée de manière à capturer pleinement le contexte, permettant ainsi une compréhension maximale. En parallèle, le décodeur opère de manière auto régressive, générant des prédictions pas à pas. Chaque prédiction devient alors l'entrée pour la prédiction suivante, créant ainsi un processus itératif qui permet au modèle d'appréhender et de générer des séquences de manière dynamique et contextuelle. Parmi les modèles de langues basés sur l'architecture Transformer, nous pouvons citer BERT (Bidirectional Encoder Representations from Transformers)[35], les modèles GPT (Generative Pre-trained Transformer) d' OpenAI[9], etc..

2.1.4.3 Évaluation du modèle de langue

L'évaluation de la qualité d'un modèle de langue repose sur la mesure de perplexité, qui représente l'inverse de la probabilité d'un corpus d'évaluation, ajusté en fonction du nombre de mots. La formule associée à cette mesure est la suivante (équation 5):

$$PP(W) = P\left(w_i | w_1, w_2, \dots, w_{i-1}\right)^{\frac{-1}{N}} \qquad (5)$$

où N est le nombre total de mots, W une séquence de mots et w un mot. Le test de perplexité offre une évaluation de la capacité d'un modèle de langue à anticiper un mot inédit en fonction du contexte. En théorie, une perplexité réduite reflète une meilleure capacité de prédiction, signalant ainsi une performance améliorée du modèle. Par conséquent, le meilleur modèle est celui qui prédit le mieux les mots d'un dataset inconnu.

2.2 Les SRAP de bout-en-bout (end to end – E2E)

Les systèmes traditionnels reposent sur un pipeline comprenant plusieurs composants tels que le modèle acoustique, le dictionnaire de prononciation, et le modèle linguistique. Ces systèmes assurent la reconnaissance vocale en transmettant séquentiellement les

[9] OpenAI est une entreprise fondée en décembre 2015 par des géants de la technologie, Elon Musk et Sam Altman, avec le support de Tesla, Amazon Web Services, Ycombinator, Peter Thiel, LinkedIn, avec un budget d'un milliard de dollars.

informations. Cependant, une approche innovante introduite au cours des dernières années suggère de remplacer les trois composants du pipeline traditionnel par un unique modèle basé sur les réseaux de neurones, appelé modèle de bout en bout.

Les systèmes de bout en bout, basés sur des modèles neuronaux, apprennent directement la relation entre le signal audio et la transcription textuelle, éliminant ainsi le besoin de connaissances phonétiques ou lexicales préalables. Le modèle de langage, autrefois considéré comme indispensable, devient facultatif dans ces configurations. Typiquement, ces systèmes sont construits à partir de réseaux de neurones profonds exploitant plusieurs unités de traitement graphique (GPU). Cette architecture permet un entraînement direct à partir de vastes corpus de données, pouvant atteindre plusieurs milliers d'heures de parole. Il est important de noter que ces systèmes sont très exigeants en termes de données.

Les principaux modèles de bout en bout incluent le modèle de classification temporelle connexionniste (CTC) [61-63], le modèle de codeur-décodeur d'attention (AED) [36] [37], le modèle de transducteur à réseau neuronal récurrent (RNN-T) [38], et les modèles de transformer [39] [40]. Ces architectures révolutionnaires ont propulsé des avancées significatives dans la reconnaissance automatique de la parole, améliorant les performances tout en simplifiant considérablement le processus de modélisation.

2.3 Évaluation de la performance d'un SRAP

La performance d'un SRAP est évaluée à partir du taux d'erreur de mots, WER (Word Error Rate) qui permet d'évaluer le taux de reconnaissance du système après le décodage. En effet, le WER compare la transcription du signal audio à reconnaître, appelée "référence", avec la transcription effectivement produite par le système, appelée "hypothèse". Le WER correspond à la somme du nombre d'opérations (substitution S, suppression D et insertion I) qui distinguent les deux transcriptions, rapportée au nombre total N de mots dans la référence, soit[28] (équation 6) :

$$WER = \frac{S + D + I}{N} \times 100 \qquad (6)$$

Il existe d'autres mesures similaires du WER, telles que le taux d'erreurs de caractères (« CER » pour Character Error Rate) ou le taux d'erreurs de syllabes (« SER » pour syllable

error rate), qui sont principalement utilisées pour les langues présentant des problèmes de segmentation.

2.4 Outils existants pour développer des SRAP

Divers outils open source dédiés à la reconnaissance vocale sont accessibles pour appuyer la recherche et l'innovation dans le domaine des applications. Parmi ces solutions, on distingue RASR[10] (abréviation de RWTH ASR) écrite en C++ par le groupe de technologie du langage humain et de reconnaissance des formes de l'université RWTH d'Aix-la-Chapelle. Le système est développé depuis 2001 et les détails de la boîte à outils ont été publiés plus tard en 2009[41]. De nos jours, deux outils majeurs ont été élaborés, à savoir Kaldi et Hugging Face Transformers, outils utilisés dans la conception d'un SRAP en fournissant des fonctionnalités et des approches diverses. Les sections suivantes examineront en détail l'utilisation de ces deux outils largement reconnus.

2.4.1 Kaldi

Kaldi[11], une boîte à outils dédiée à la reconnaissance vocale, a été développée par Daniel Povey et son équipe pour la recherche dans ce domaine[42]. Elle présente un système fondé sur des transducteurs d'état finis pondérés (WFST), exploitant OpenFST et codé en C++. En outre, Kaldi intègre les dernières avancées en modélisation acoustique, incluant l'usage de modèles de mélange gaussien sous-espace et de réseaux neuronaux profonds. Pour évaluer ses fonctionnalités, Kaldi propose un ensemble substantiel de recettes, constamment actualisé par ses contributeurs. Un élément majeur réside dans la description détaillée des fonctionnalités de Kaldi sur son site web, ainsi que dans la présence d'un forum actif dédié aux discussions et résolutions de problèmes liés à l'implémentation de cette boîte à outils.

[10] RWTH ASR fait référence à la reconnaissance automatique de la parole développée par l'Université RWTH Aachen en Allemagne, site officiel : https://www-i6.informatik.rwth-aachen.de/rwth-asr/
[11] https://github.com/kaldi-asr/kaldi

2.4.2 Hugging Face Transformers

La bibliothèque Hugging Face Transformers[12] est largement reconnue dans le domaine de l'apprentissage automatique et du traitement du langage naturel, proposant une variété de modèles pré-entraînés pour diverses tâches telles que la classification de texte, la génération de texte, la traduction automatique, et la transcription automatique de la parole.

Parmi ces modèles, le wav2vec 2.0[13] [41] se distingue comme une amélioration de l'approche wav2vec de Facebook AI, offrant une efficacité particulière dans l'apprentissage de représentations de haute qualité à partir de données vocales non annotées. En parallèle, Whisper[14], modèle pré-entraîné pour la reconnaissance automatique de la parole d'OpenAI en septembre 2022[43]. Contrairement à ses prédécesseurs, tel que Wav2Vec 2.0, Whisper se distingue par son pré-entraînement sur une vaste quantité de données de transcription audio étiquetées, totalisant 680 000 heures, dont 117 000 heures sont dédiées à l'ASR multilingue. Ces modèles, pré-entraînés sur des ensembles de données vocales étendues, captent des motifs complexes et apprennent des caractéristiques pertinentes de la parole. L'exploitation de modèles pré-entraînés présente l'avantage d'utiliser des connaissances préalables provenant de vastes ensembles de données, réduisant ainsi le temps et les ressources nécessaires pour entraîner des modèles spécifiques. Ces modèles peuvent ensuite être adaptés (fine-tuning)[15] sur des données spécifiques à la tâche, améliorant ainsi leurs performances pour des scénarios particuliers, notamment la reconnaissance vocale dans des domaines d'application spécifiques.

2.5 Défis et travaux récents sur les SRAP pour les langues peu-dotées

Les applications vocales ont facilité l'interaction homme-machine pour de nombreuses tâches, par exemple la commande vocale, l'identification du locuteur et les systèmes de traduction vocale. Aujourd'hui, ces applications sont à portée de main et peuvent être intégrées dans des ordinateurs de bureau ou des appareils mobiles.

[12] https://huggingface.co/
[13] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec/unsupervised
[14] https://github.com/openai/whisper
[15]https://huggingface.co/blog/fine-tune-whisper, https://huggingface.co/blog/fine-tune-wav2vec2-english

Les systèmes de commande vocale sont utilisés pour aider l'homme à effectuer des activités physiques, notamment pour aider les personnes handicapées, les systèmes d'identification du locuteur sont utilisés pour déterminer l'identité des personnes, et les systèmes de traduction de la parole peuvent aider les personnes à communiquer dans plusieurs langues. De nos jours, de nombreux systèmes de reconnaissance vocale peuvent fonctionner dans les langues dominantes telles que l'anglais, le français, le mandarin, le japonais, etc. Étant donné qu'un grand nombre de ressources sont disponibles dans ces langues, des modèles statistiques robustes peuvent être formés pour des SRAP. De nombreuses langues ne sont toujours pas disponibles pour les applications vocales, en particulier les langues peu-dotées. Dans le contexte de la technologie du langage humain, une « langue peu-dotée » se réfère à une langue qui présente les problèmes suivants : ressources insuffisantes en termes de données vocales transcrites ou de dictionnaires de prononciation, système d'écriture instable, faible quantité de vocabulaire ou absence (ou inexistence) de ressources électroniques pour le traitement de la parole et du langage, etc.[44] [45].

2.5.1 Défis

Comme souligné par Besacier[46], le défi majeur dans la mise en place d'un SRAP pour les langues peu-dotées réside dans le déficit de ressources. Les systèmes de Traitement Automatique du Langage Naturel (TALN) exigent une quantité importante de données dans la langue cible pour former des modèles statistiques robustes dédiés à la reconnaissance vocale, ce qui pose des difficultés particulières pour les langues rares ou en voie de disparition. La collecte de données, surtout dans des régions rurales, nécessite des méthodologies innovantes pour garantir des corpus durables. Impliquer la communauté, selon Walsh[47], est important, mais cela implique de surmonter le fossé entre les compétences techniques des locuteurs natifs et des experts en technologie. L'éthique intervient également, avec des règles nécessaires pour traiter des langues spécifiques, notamment sur le choix et la publication des données. Enfin, les modèles à faibles ressources sont un autre défi, car l'entraînement sur des données limitées peut conduire à des performances dégradées des systèmes de reconnaissance vocale. Les chercheurs doivent ainsi élaborer des stratégies innovantes pour améliorer ces modèles limités.

2.5.2 Bref historique des initiatives de recherche pour les langues peu dotées

Au fil des dernières décennies, diverses recherches ont été entreprises pour faciliter l'identification, l'exploration et la légitimation des langues peu dotées. Besacier, Le et al. [21] ont été les précurseurs en présentant des travaux de reconnaissance automatique de la parole pour les langues vietnamiennes et khmères à faibles ressources. Ultérieurement, des études ont été publiées concernant des langues africaines telles que le somali[48], l'amharique[49] et les langues sud-africaines[50]. L'intérêt croissant pour les langues peu dotées a conduit à la rédaction d'un article introductif sur la reconnaissance automatique de la parole pour ces langues[46]. Par ailleurs, Elodie Gauthier a consacré ses efforts aux langues sub-sahariennes, notamment le wolof, où elle a travaillé sur la collecte de ressources pour la reconnaissance de la parole[28], spécifiquement pour le wolof, financée par le projet ALFFA. Le projet ANR Blanc ALFFA (African Languages in the Field : Speech Fundamentals and Automation) vise à mettre en place des méthodes novatrices pour décrire les langues africaines peu dotées en utilisant des outils de traitement automatique de la parole tels que le SRAP et la synthèse vocale (TTS). Récemment, Elodie a également contribué à la création d'un bot vocal dialoguant en wolof[51].

Cette section rend compte des travaux réalisés jusqu'à présent dans le domaine du traitement automatique de la parole pour les langues peu dotées, ainsi que des initiatives visant à promouvoir les technologies vocales pour et dans ces langues. Toutefois, des défis subsistent, notamment en ce qui concerne l'accès et le partage des ressources.

2.6 Les données

2.6.1 Les ressources nécessaires pour l'apprentissage des modèles

Pour établir un SRAP classique, l'apprentissage des modèles requiert plusieurs ressources essentielles. En premier lieu, un corpus audio substantiel, généralement de plusieurs dizaines à plusieurs centaines d’heures, est impératif pour développer les modèles acoustiques du système. Ce corpus audio transcrit forme le socle sur lequel le moteur de reconnaissance s'appuie afin de comprendre et d’interpréter les variations acoustiques du langage parlé.

Parallèlement, un vaste corpus de textes, idéalement de plusieurs dizaines à plusieurs centaines de millions de mots, est indispensable pour construire un modèle de langue capable de couvrir un large vocabulaire et d'embrasser une diversité de thèmes [42]. En complément, un lexique de prononciations, comprenant au moins quinze milles (15 000) entrées, est nécessaire pour assurer la correspondance entre les modèles acoustiques et syntaxiques, établissant ainsi une liaison dans le processus de reconnaissance [5]. Ces ressources, avec une emphase particulière sur le corpus audio transcrit, sont fondamentales pour garantir un apprentissage robuste et une performance optimale du système de reconnaissance automatique de la parole dans un cadre classique.

Pour mettre en place un SRAP de bout en bout (E2E), une quantité substantielle de données est nécessaire, atteignant parfois plusieurs milliers d'heures de parole. Ces systèmes, basés sur des modèles neuronaux, sont extrêmement gourmands en données. L'apprentissage d'un système E2E exige une quantité importante de données vocales transcrites, une ressource souvent limitée pour les langues à faibles ressources. Cette approche novatrice permet d'apprendre de manière holistique la relation entre le signal audio et la transcription textuelle, réduisant ainsi la dépendance envers des ressources spécifiques telles que des lexiques de prononciations étendus. En résumé, la mise en place d'un système E2E exige une masse considérable de données vocales pour obtenir des performances optimales, soulignant l'importance de l'accès à des corpus de grande envergure dans le développement de ces systèmes avancés.

2.6.2 L’acquisition des données (corpus textuel – dictionnaire de prononciation)

Pour construire un modèle de langue et un modèle de prononciation, il nous faut respectivement un corpus textuel et un dictionnaire de mots accompagnés de leur transcription phonétique et les modèles statistiques. Ces données doivent être de quantité significative pour une modélisation efficace de la parole. En effet, on peut obtenir ces données de différentes manières, que ce soit par l’achat de données ou en recourant à la constitution de ses propres datasets (en collectant les données par ses propres moyens ou en les acquérant par le biais d’accords partenariaux).

2.6.2.1 L’achat des données

Pour l'achat de données, deux possibilités se présentent :

- l'achat de jeux de données disponibles sur catalogue : qui consiste à acquérir des ensembles de données préexistants disponibles dans un catalogue. Ces catalogues regroupent souvent une variété de données issues de sources diverses, offrant ainsi une sélection préétablie couvrant différentes thématiques. Cependant, cette solution ne convient que lorsque la langue recherchée est disponible dans ces catalogues, ce qui n'est pas toujours le cas, en particulier pour les langues subsahariennes, où la disponibilité de données prédéfinies peut être limitée ;
- l'achat de jeux de données sur demande : dans ce cas nous avons la possibilité de passer une commande personnalisée, qui permet aux clients d’avoir possibilité de formuler des commandes sur mesure, spécifiant les critères précis dont ils ont besoin. Cela peut inclure des spécifications telles que le type de données, la période temporelle, ou d'autres caractéristiques spécifiques en fonction de leurs besoins particuliers. Cette approche sur mesure est particulièrement avantageuse lorsque les besoins en données sont spécifiques et nécessitent une adaptation précise aux exigences du projet en cours.

2.6.2.2 Constitution de son propre dataset

Obtenir des ressources sans passer par une commande auprès d'un prestataire offre deux possibilités :

- moissonner le Web pour la collecte de données en ligne : étant donné que les fournisseurs de banques de données linguistiques sont rares, voire inexistants, pour les langues d'Afrique subsaharienne, les ressources peuvent être directement récoltées sur le Web. Les sites diffusant des données libres de droits, tels que les dépôts Github[16], les archives documentaires[17], ainsi que les ressources linguistiques dédiées au traitement automatique des langues[18][19], peuvent constituer des sources

[16] https://github.com/
[17] https://archive.org/
[18] https://openslr.org/resources.php
[19] https://linguistic-lod.org/

intéressantes pour initier la collecte de données textuelles langagières. L'avantage principal de cette méthode réside dans la maîtrise complète de la chaîne de collecte de jeux de données. La collecte peut être automatisée via des méthodes de crawling et de scraping de sites Web, permettant la collecte de volumes importants de données textuelles hétérogènes pour la création de modèles linguistiques. Le processus d'acquisition de données par ses propres moyens présente divers avantages et inconvénients. L'approche semi-automatique offre plusieurs bénéfices, notamment une acquisition efficace et rapide des données. La maîtrise totale de la chaîne de collecte constitue un avantage essentiel, permettant un contrôle optimal sur le processus. Cependant, des défis subsistent, notamment la recherche de données qui se révèle souvent difficile pour les langues d'Afrique subsaharienne, limitant ainsi la disponibilité de ressources ;

- contractualiser avec un partenaire : établir un partenariat avec une entreprise ou une ONG détenant des jeux de données correspondant aux besoins peut être une solution rapide pour obtenir des ressources. Cependant**,** il est pertinent de rechercher des partenaires œuvrant pour la promotion et la sauvegarde des langues locales. Cette approche favorise une collaboration efficace pour l'obtention rapide de ressources nécessaires.

Conclusion

En conclusion, ce chapitre sur la modélisation de la parole a exploré les différents aspects des SRAP. Nous avons examiné les systèmes traditionnels et ceux de bout en bout, ainsi que les méthodes d'évaluation de leur performance. Nous avons également passé en revue les outils existants, les défis pour les langues peu dotées et l'importance des données. Dans le chapitre suivant, nous aborderons la réalisation technique du projet, mettant en pratique les concepts discutés ici. Nous détaillerons les étapes de développement du système de reconnaissance de la parole pour le projet Kallaama, en mettant l'accent sur les choix techniques et les défis rencontrés.

Chapitre 3 : REALISATION TECHNIQUE

Introduction

Dans ce chapitre, nous allons voir en détail la mise en œuvre pratique des objectifs énoncés précédemment. À travers une méthodologie rigoureuse, nous explorons les différentes étapes nécessaires à la construction des datasets pour les trois (3) principaux langues vernaculaires du Sénégal, notamment le wolof, le sérère et le pulaar. De la récupération des données au dépôt final, chaque étape est cruciale pour garantir la qualité et l'efficacité des modèles que nous cherchons à développer par la suite.

3.1 Méthodologie

La réalisation technique de ce projet s'articule autour d'une méthodologie visant à construire des datasets pour le développement de modèles de langage. La méthodologie comprend plusieurs étapes clés, chacune contribuant de manière significative à la constitution d'un ensemble de données de qualité.

3.1.1 Le workflow

Le processus de création de datasets pour les modèles linguistiques commence par la collecte de données authentiques et diversifiées provenant de diverses sources telles que des livres, articles de presse, enregistrements audio, dialogues quotidiens, chansons et contes traditionnels. La collaboration avec les communautés locales est essentielle pour garantir l'authenticité et la pertinence culturelle des données. Une fois collectées, les données subissent un prétraitement rigoureux pour éliminer les éléments superflus, normaliser les textes, corriger les erreurs et segmenter les phrases et les mots, tout en respectant la confidentialité et l'éthique. Ensuite, les données sont organisées en corpus distincts pour chaque langue, enrichis de métadonnées et validés pour assurer leur cohérence et représentativité. La création d'un lexique de prononciation précis est également précieuse, nécessitant une collaboration avec des linguistes pour valider la phonétique des mots. Enfin, les données sont déposées de manière structurée et documentée, avec un système de versionnement pour assurer la traçabilité, rendant ces ressources accessibles et utilisables

pour la recherche et le développement de modèles linguistiques pour les langues locales du Sénégal.

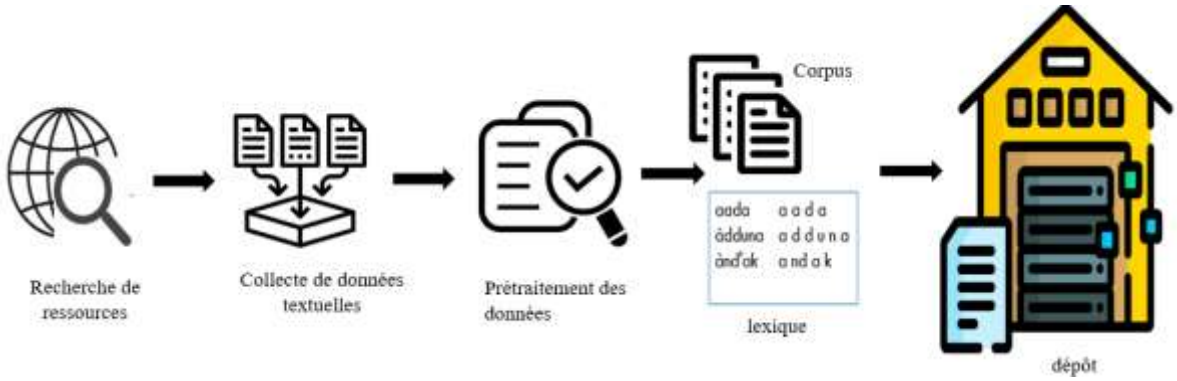

Figure 5 : le workflow

3.1.2 Méthodologie de gestion du projet

Le choix du modèle de développement dépend de la nature et de l'envergure du projet. Pour les projets où les données sont incomplètes ou les besoins vagues, une approche itérative ou axée sur les prototypes est recommandée. Les méthodologies AGILE, particulièrement populaires aujourd'hui, se distinguent par leur approche collaborative et adaptable, centrée sur les besoins du client et opposée aux méthodes traditionnelles telles que le modèle en cascade. Conçue initialement pour le développement web et informatique, l'Agile est désormais appliquée à divers secteurs grâce à sa flexibilité. Nous avons choisi la méthodologie Scrum, une approche Agile, pour sa capacité à répondre aux besoins évolutifs du projet et son succès éprouvé dans différents contextes.

3.1.2.1 La Méthodologie Scrum

La méthodologie de gestion de projets la plus renommée issue de l'approche Agile est incontestablement le "Scrum", baptisé ainsi en référence à la "mêlée" dans le langage du rugby. Dans ce contexte, le chef de projet est désigné sous le nom de "Scrum Master".

Cette approche est structurée autour de cycles courts, communément appelés des itérations, et dans le langage Scrum, chaque itération est appelée un "sprint". À chaque nouveau sprint, l'équipe projet se réunit pour établir la liste des tâches à accomplir, désignée comme le "sprint backlog".

Cette méthodologie est intrinsèquement liée à la logique de développement de produit, d'où l'existence d'acteurs spécifiques tels que le Product Owner. Des réunions Scrum, couramment organisées quotidiennement, jouent un rôle essentiel. Ces rencontres sont de courtes sessions d'échange au cours desquelles les membres de l'équipe projet partagent leurs progrès ainsi que leurs défis rencontrés.

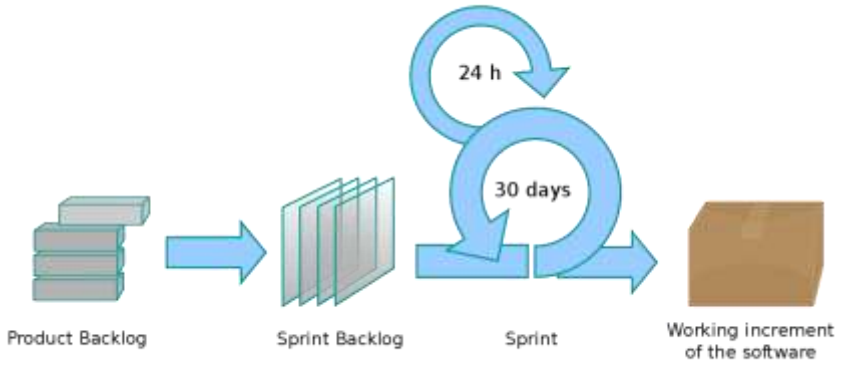

Figure 6: Cycle de vie de Scrum

Source : https://www.istex.fr/agile-scrum/

3.2 Récupération des données

3.2.1 Méthode de collecte

Pour la collecte de données, nous avons opté la méthode du moissonnage Web (communément appelé « Web Scraping »), un processus qui consiste concrètement à extraire des informations à partir d'un site web. Cette méthode permet d'acquérir divers types de données précieuses, tels que des articles, des adresses e-mail, des numéros de téléphone etc., et de les regrouper dans une base de données.

Le Web Scraping peut être effectué de deux manières distinctes : manuellement ou automatiquement.

Le Scraping manuel implique le copier-coller d'informations/données pour constituer une base de données. Cette approche est souvent utilisée pour des quantités limitées de données en raison de sa nature chronophage.

En revanche, le Scraping automatique repose sur l'utilisation d'outils dédiés qui explorent et extraient des informations depuis des sites web. Nous privilégierons cette méthode en raison du volume important de données que nous souhaitons recueillir pour des fins d'apprentissage automatique.

3.2.1.1 Fonctionnement

Le Web Scraping repose sur deux étapes clés : la navigation automatique des sites web et l'extraction des données. Les **crawlers** ou **spiders** sont des programmes qui parcourent le web pour rechercher et indexer le contenu, et sont souvent utilisés par les moteurs de recherche comme Google pour mettre à jour les index et les classements des sites web. Ils sont disponibles sous forme d'outils préconstruits permettant de spécifier un site ou un terme de recherche. Une fois sur le site cible, les **scrapers** interviennent pour extraire les informations pertinentes. Utilisant des structures HTML, les scrapeurs appliquent des techniques telles que les expressions régulières (regex), XPath et les sélecteurs CSS pour repérer et extraire les données spécifiques, comme un nom de marque ou un mot-clé donné. En résumé, les crawlers s'occupent de la navigation et de l'indexation, tandis que les scrapeurs se chargent de l'extraction des données ciblées.

Le processus de base du web scraping se décompose en quelques étapes simples :

- spécifiez les URL des sites web et des pages que nous souhaitons scraper ;
- effectuez une requête HTML vers ces URL (c'est-à-dire, "visitez" les pages) ;
- utilisez des localisateurs tels que des expressions régulières pour extraire les informations souhaitées du HTML ;
- enregistrez les données dans un format structuré, tel que CSV ou JSON.

Ainsi, le Web Scraping permet de collecter rapidement des données spécifiques sur le web, en automatisant le processus d'exploration et d'extraction.

Figure 7 : fonctionnement du Web Scraping

3.2.1.2 Environnement de travail : Google Colab

Google Colab[20], également connu sous le nom de Colaboratory, est un service cloud gratuit de Google basé sur Jupyter Notebook. Le choix de Google Colab comme environnement de travail découle de ses fonctionnalités avancées, offrant un accès en ligne à des ressources de calcul telles que les unités de traitement graphique (GPU). Cette caractéristique permet d'assurer une collecte de données textuelles rapide et efficace.

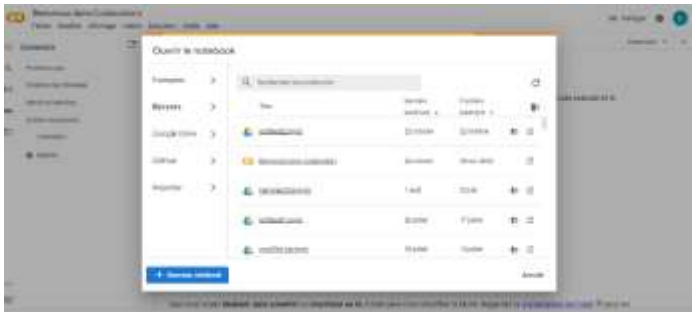

Figure 8 : interface d'accueil de Google Colab

3.2.1.3 Langage de programmation : Python

Python[21] est un langage de programmation de haut niveau, polyvalent et extrêmement populaire. Il a été créé en 1991 par Guido Van Rossum et est devenu l'un des langages de programmation les plus largement utilisés dans le monde. Reconnu pour sa simplicité et sa facilité d'utilisation, Python est idéal pour le calcul scientifique, l'analyse de données et l'intelligence artificielle. En effet, le choix de Python comme langage de programmation est motivé par sa polyvalence, sa puissance et son écosystème riche en bibliothèques

[20] https://colab.research.google.com/
[21] https://www.python.org/

spécialisées. Python est largement utilisé dans le domaine de l'analyse de données et du NLP, offrant ainsi une multitude d'outils pour simplifier et accélérer le processus de collecte de données textuelles. Cette approche permet également une personnalisation approfondie du processus d'extraction, assurant une adaptation précise aux exigences spécifiques du projet.

3.2.1.4 Outils de collecte : BeautifulSoup et Requests

Les outils de collecte, BeautifulSoup et Requests, ont été rigoureusement sélectionnés pour leur efficacité dans l'extraction de données à partir de pages web. BeautifulSoup, une bibliothèque Python spécialisée dans le web scraping, excelle dans l'analyse de documents HTML et XML, offrant une navigation intuitive au sein de leur structure. Elle se distingue par sa capacité à traiter des codes mal formatés, et ses classes telles que BeautifulSoup, Tag, Navigable String et Comment facilitent l'analyse approfondie des données. D'autre part, le module Requests est dédié à l'envoi de requêtes HTTP, simplifiant l'interaction avec les serveurs web et retournant un objet Response contenant toutes les données de la réponse. Le choix de ces deux outils repose sur la puissance et la flexibilité de BeautifulSoup, largement adoptée pour des tâches de web scraping, et sur la simplicité d'utilisation.

3.2.1.5 Outils d'implémentation de lexique : Phonetisaurus et Docker

Docker est une plateforme open-source qui permet de créer, déployer et exécuter des applications dans des conteneurs légers et isolés. Ces conteneurs encapsulent l'application avec toutes ses dépendances, assurant une exécution cohérente sur différents environnements. Pour implémenter un dictionnaire de prononciation, Docker peut être utilisé pour configurer un environnement stable et reproductible, garantissant que les outils nécessaires fonctionnent correctement sur n'importe quel système.

En combinant l'outil Phonetisaurus avec Docker, il est possible de créer un environnement de développement uniforme pour travailler sur des dictionnaires de prononciation, simplifiant l'intégration et le déploiement des modèles phonétiques.

3.2.2 Le Web Scraping avec BeautifulSoup

Dans cette section, nous allons présenter les différentes étapes pour la récupération des données avec BeautifulSoup et request https://www.wolof-online.com sur Google Colab.

Pour utiliser Google colab il suffit d'aller sur google drive ensuite cliquer sur nouveau ensuite sur « plus » et choisissez « Colaboratory » ou bien accéder directement à l'interface de google colab en aller sur ce lien https://colab.research.google.com/.

La fenêtre se présentera comme suit :

Figure 9 : interface de développement

Pour installer BeautifulSoup et requests, vous pouvez taper les commander suivantes

```
!pip install bs4
!pip install requests
```

Une fois que ces deux bibliothèques installées, nous pouvons procéder étape par étape pour récupérer les données du site wolof-online.

Étape 1 : Importation des bibliothèques

Dans un premier temps, nous procédons à l'importation des bibliothèques requises. Dans un second temps, Nous utiliserons la bibliothèque requests pour transmettre une requête GET au site web, et de la bibliothèque beautifulsoup4 pour analyser le contenu HTML du site.

```
import requests
from bs4 import BeautifulSoup
```

Étape 2 : Configurer le Scraper

Cette étape consiste à envoyer une requête GET au site web afin de récupérer le contenu HTML. Nous utiliserons la fonction requests.get() à cet effet.

```
# Effectuer une requête vers le site web
```

```
url = 'https://www.defuwaxu.com/'
response = requests.get(url)
```

Maintenant que nous avons récupéré le contenu HTML, procédons à son analyse avec BeautifulSoup. Nous ferons usage de la fonction BeautifulSoup () à cette fin.

```
# Analyser le contenu HTML du site web
soup = BeautifulSoup(response.content, 'html.parser')
```

La fonction BeautifulSoup () prend deux arguments : le premier est le contenu HTML que nous voulons analyser, et le deuxième est le parseur que nous souhaitons utiliser. Dans ce cas, nous utilisons le parseur html.parser.

Maintenant que l'analyse du contenu HTML est effectuée, entamons le processus de web scraping. Supposons que notre objectif soit d'extraire des informations telles que les titres et les descriptions du site « defuwaxu ». Afin d'atteindre cet objectif, nous devons identifier l'élément HTML qui renferme les données que nous souhaitons extraire. Pour ce faire, nous devons naviguer sur le site, effectuer un clic droit sur la page de l'article spécifique, puis sélectionner l'option « Inspecter ». À partir de là, nous serons en mesure de localiser le schéma HTML qui englobe les informations recherchées.

Figure 10 : Page d'acceuil du site defuwaxu

Figure 11 : inspection d'un article

On observe que, pour chaque publication, le titre est contenu dans une balise h1 et la description dans des balises h2.

Ensuite, nous procéderons à l'inventaire des sous-liens ainsi que du nombre de pages pour chaque catégorie. La figure ci-dessous illustre la catégorie "li-fes", où l'on peut constater une pagination étendue sur 61.

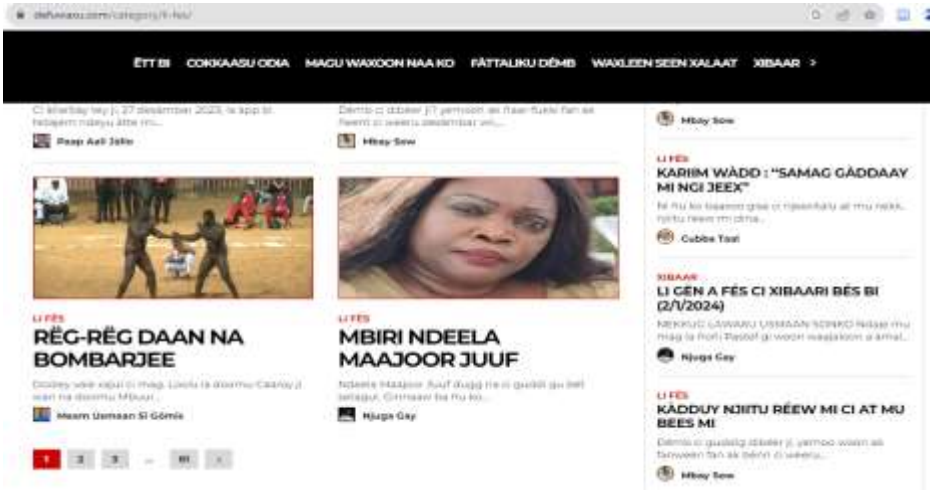

Figure 12 : repérage des sous liens et numéro de page

Le script ci-dessous a pour but d'afficher les différents liens de catégories :

```
# Définition de la fonction pour extraire les liens des catégories
def extract_category_links(url):
    # Création d'un ensemble vide pour stocker les liens uniques
    links = set()
```

```
    # Itération sur toutes les balises <a> dans le contenu HTML
    for link in soup.find_all('a'):
        # Récupération de la valeur de l'attribut 'href' de la balise
<a>
        href = link.get('href')

        # Vérification des conditions pour inclure le lien dans les
résultats
        if href and
href.startswith('https://www.defuwaxu.com/category/') and 'cokkaasu-
odia/' not in href:
            # Ajout du lien à l'ensemble s'il satisfait les conditions
            links.add(href)

    # Retourne l'ensemble de liens extraits
    return links

# Extraction des liens de catégorie en utilisant la fonction définie
category_links = extract_category_links(url)

# Affichage des liens extraits
for link in category_links:
    print(link)
```

Les résultats devraient ressembler à ce qui suit :

```
https://www.defuwaxu.com/category/xibaar/politig/
https://www.defuwaxu.com/category/waxleen-seen-xalaat/
https://www.defuwaxu.com/category/fattaliku-demb/
https://www.defuwaxu.com/category/xibaar/bindkatu-ayubes-bi/
https://www.defuwaxu.com/category/xibaar/
https://www.defuwaxu.com/category/xibaar/bayyi-ci-xel/
https://www.defuwaxu.com/category/xibaar/gis-gis/
https://www.defuwaxu.com/category/xibaar/kii-kumu/
https://www.defuwaxu.com/category/magu-waxoon-naa-ko/
https://www.defuwaxu.com/category/xibaar/koom-koom/
https://www.defuwaxu.com/category/li-fes/
https://www.defuwaxu.com/category/xibaar/diine/
https://www.defuwaxu.com/category/xibaar/taggat-yaram/
```

Ensuite, nous passerons à l'extraction des liens à partir d'une URL, ainsi qu'au contenu des pages web, en tenant compte de la possibilité de présence de pagination.

```
# Fonction pour extraire les liens à partir d'une URL
def extract_links(url):
    # Effectuer une requête HTTP pour obtenir le contenu de la page
web
```

```
    response = requests.get(url)

    # Utiliser BeautifulSoup pour analyser le contenu HTML de la page
    soup = BeautifulSoup(response.text, 'html.parser')

    # Initialiser un ensemble pour stocker les liens uniques
    links = set()

    # Itérer sur toutes les balises <a> de la page
    for link in soup.find_all('a'):
        # Récupérer l'attribut 'href' de la balise <a>
        href = link.get('href')

        # Vérifier les conditions pour inclure le lien dans les résultats
        if href and href.startswith('https://www.defuwaxu.com/') and href
!= 'https://www.defuwaxu.com/' and '/category/' not in href and
'/puukare' not in href:
            # Ajouter le lien à l'ensemble s'il satisfait les conditions
            links.add(href)

    # Retourner l'ensemble de liens extraits
    return links

# Fonction pour extraire le contenu d'une page web
def extract_content(url):
    # Effectuer une requête HTTP pour obtenir le contenu de la page web
    response = requests.get(url)

    # Utiliser BeautifulSoup pour analyser le contenu HTML de la page
    soup = BeautifulSoup(response.text, 'html.parser')

    # Extraction des titres (balise h1)
    titles = [h1.get_text() for h1 in soup.find_all('h1')]

    # Extraction des descriptions (balises h2 et p)
    descriptions = [tag.get_text() for tag in soup.find_all(['h2', 'p'])]

    # Retourner les titres et les descriptions extraits
    return titles, descriptions
```

Étape 3 : Préparation pour l'exécution

Une fois que nous avons identifié l'élément HTML contenant les informations à extraire, nous pouvons donner un lien de catégorie et procéder à l'extraction. Par conséquent, le script final devrait ressembler à ce qui suit :

```
# URL de base pour la catégorie spécifique
base_url = 'https://www.defuwaxu.com/category/magu-waxoon-naa-ko/'
```

```
# Extraction des liens pour chaque page et écriture des résultats dans un fichier
with open('resultats.txt', 'w', encoding='utf-8') as file:
    # Itération sur les pages de 1 à 6
    for page in range(1, 7):
        # Construction de l'URL de la page actuelle
        url = f"{base_url}page/{page}/"

        # Appel de la fonction extract_links pour obtenir les liens de la page
        page_links = extract_links(url)

        # Écriture des liens de la page dans le fichier
        file.write(f"Liens pour la page {page}:\n")
        for link in page_links:
            # Appel de la fonction extract_content pour obtenir les titres et les descriptions de chaque lien
            titles, descriptions = extract_content(link)

            # Écriture des titres dans le fichier
            file.write("Titres:\n")
            for title in titles:
                file.write(f"{title}\n")

            # Écriture des descriptions dans le fichier
            file.write("Descriptions:\n")
            for description in descriptions:
                file.write(f"{description}\n")

            # Ajout d'une ligne vide pour séparer les informations de chaque lien
            file.write('\n')
```

Les résultats finaux devraient ressembler à ce qui suit pour le sous liens https://www.defuwaxu.com/category/magu-waxoon-naa-ko/.

Figure 13 : résultat obtenu après moissonnage

Pour les autres sous liens, nous devons suivre les mêmes étapes pour récupérer les données.

En appliquant ces étapes à d'autres sites web, cette approche méthodique offre une flexibilité pour s'adapter aux structures variées des pages HTML. Chaque site peut nécessiter une adaptation des critères de filtrage et des balises spécifiques, mais le cadre général reste applicable. En suivant ces étapes, il est possible de collecter efficacement des données spécifiques à partir de différentes sources en ligne.

3.2.3 Données récupérées

Dans notre démarche visant à enrichir les ressources linguistiques, nous avons entrepris la collecte de données à partir de diverses sources, allant des projets de recherche open source aux sites web et fichiers PDF éducatifs. Cette diversité d'origines a permis d'obtenir un corpus textuel représentatif et varié, favorisant ainsi une meilleure compréhension des nuances linguistiques propres à chaque langue.

3.2.3.1 Le wolof

Pour constituer le corpus textuel en wolof, nous avons commencé par la collecte de données open source provenant du projet ALFFA, dirigé par Elodie Gauthier[22], dans le but de construire un modèle de langue[52]. Ils ont rassemblé et prétraité des données textuelles en référence à (Nouguier Voisin, 2002)[53], totalisant ainsi 106 206 mots. De plus, des données textuelles extraites du Web, telles que la Déclaration universelle des droits de l'homme, le

[22] https://github.com/getalp/ALFFA_PUBLIC/tree/master/ASR/WOLOF/LM

Message de Silo, la Bible et la base de données de Wikipédia, ont été incluses, atteignant un total de 641 483 mots. Nous avons également récupéré des données de Masakhane[23] à partir du dépôt GitHub, comprenant 44 812 mots brutes extraits des sites d'actualités Defuwaxu et Saaba. De plus, nous avons intégré des données open source issues du challenge challenge Programme Algorithme et Solution (PAS)[24] organisé par l'Institution des Algorithmes du Sénégal (IAS), totalisant 156 712 mots.

Afin d'enrichir davantage notre corpus textuel, nous avons opté pour la collecte de données textuelles supplémentaires en wolof sur le Web. Nous avons recherché des données bien structurées, conformes aux règles syntaxiques, et avons trouvé des fichiers PDF provenant de sites web éducatifs, religieux et d'actualités, collectant ainsi un total de 17 160 mots. Cependant, étant donné la rareté des documents écrits en wolof, nous avons extrait des contenus de Wikipédia depuis le dépôt de la base de données de Wikipédia wolof, récupérant ainsi 497 766 mots. Enfin, nous avons procédé à la récupération automatique des données textuelles sur des sites pertinents tels que des sites d'actualités, socio-politiques, culturels, historiques, linguistiques et religieux, aboutissant à l'extraction de 578 807 mots au total.

Les tableaux ci-dessous montrent les différents liens trouvés, leurs descriptions, les dates où ils ont été explorés, ainsi que le nombre de mots récupérés.

Tableau 1: sites explorés et le nombre de mots récupérés

Liens (Wolof)	Description	Date d'exploration	Nombres de mots
defuwaxu.com	site d'actualité et d'info socio politique	03/07/2023	475 000
wolof-online.com	site d'actualité et d'info, sur les cultures, l'histoire, la langue et la politique	03/07/2023	35 000
seneplus.com	site d'actualité et d'info socio politique	20/05/2023	25 418
jotnanews.com	site d'actualité et d'info socio politique	20/05/2023	4 217
etabetapi.com/read/wolnt/Matt/1	nouveau testament en wolof par Matthew	19/05/2023	23 000

[23] https://www.masakhane.io/
[24] https://www.ias.sn/pas

historique-et-biographique-de-cheikh	site dédié à l'histoire et à la biographie de Serigne Touba	22/05/2023	8 983
projectfichte.org/xellu-jukki-iii/	article sur l'histoire d' Hubert Fichte	22/05/2023	7 189

Pour résumé, nous avons obtenu plus de 915k mots issus de projet de recherche, 515k mots extrait de Wikipédia et d'ouvrages trouvés dans le net. Avec la technique du Web Scraping automatique, nous avons collecté plus de 578k mots issus de différents sites.

Le tableau 2 résume les données finalement récupérées.

Tableau 2: données wolof récupérés

Sources	Nombre de mots
ALFFA	747 689
Masakhane	44 812
Challenge PAS	156 712
Ouvrages	17 160
Wikipédia	497 766
Sites web	578 807
Total	**2 042 946**

3.2.3.2 Le pulaar

Dans notre quête de données exhaustive sur le pulaar, nous avons recueilli des données bien structurées et conformes aux règles syntaxiques et à la codification de la langue pulaar qui est le « fuc ». Notre recherche a abouti à la découverte de fichiers PDF éducatifs, de sensibilisation et de dictionnaires pulaar-français, contenant un ensemble de 23 888 mots. Cependant, conscients de la rareté des documents rédigés en pulaar, nous avons élargi notre démarche en procédant à la récupération automatique de données textuelles à partir de sites pertinents tels que des plateformes d'actualités, socio-politiques, culturelles et historiques. Cette approche a abouti à l'extraction remarquable de 1 066 265 mots au total, enrichissant ainsi notre corpus de manière significative.

Les liens des sites visités, ainsi que le nombre de mots récupérés pour chaque site, sont représentés dans le tableau 3.

Tableau 3 : liens explorés et nombre de mots pulaar récupéré

Liens (Pulaar)	Description	Date d'exploration	Nombres de mots
pulaar.org (Mauritanie)	Un site multimédia d'actualités et de plaidoyer en Pulaar	27/06/2023	392 600
binndipulaar.com (Sénégal)	site d'actualité et d'info, sur les cultures, l'histoire, la langue et la politique	30/05/2023	351 529
golal.info (Mauritanie)		22/06/2023	7 986
		20/05/2023	
rendofulbe.blogspot.com/ (Mauritanie)	blog	25/05/2023	10 987
lowre-pulaagu.com (Mauritanie)	Site d'actualité et d'info	21/06/2023	230 667
hammadi-jah (Mauritanie)	Site de blog scientifique	23/05/2023	42 496
dingiralfulbe.com (Sénégal)	Site d'actualité	22/06/2023	30 000

3.2.3.3 Le sérère

Malgré sa position en tant que la troisième langue la plus parlée au Sénégal, la langue sérère présente un défi particulier en matière d'acquisition de données écrites. Actuellement, aucun document écrit en sérère n'a été identifié dans web, et le contenu textuel en sérère sur le Web est pratiquement inexistant. Dans notre quête de données, nous avons entrepris des recherches à diverses bibliothèques, y compris l'Institut Fondamental d'Afrique Noir (IFAN) à Dakar, dans l'espoir de trouver des ouvrages rédigés en sérère. Malheureusement, même après avoir trouvé des livres, le contenu sérère disponible dans ces ouvrages s'est avéré limité.

L'acquisition de données en sérère demeure donc un défi majeur. Cependant, afin de pallier cette lacune, nous explorons la possibilité d'obtenir des transcriptions d'audios en sérère.

3.3 Prétraitement des données

Cette section dédiée au prétraitement des données revêt une importante capitale dans le cadre de notre démarche de NLP, plus particulièrement dans le contexte de la création de modèles de langues. Les données textuelles, extraites initialement de diverses sources, sont fréquemment entachées de bruit, de balises HTML indésirables, et de caractères spéciaux. Afin de les rendre adaptées à l'utilisation dans la création de modèles de langue, un processus de prétraitement systématique s'avère indispensable.

3.3.1 Clean_v1 : élimination des caractères indésirables

La première phase, Clean_v1, est dédiée à l'élimination de certains caractères spéciaux, des balises HTML et d'autres éléments indésirables. Cette étape comprend également les actions suivantes :

- suppression des espaces vides supplémentaires,
- remplacement des guillemets par des guillemets informatiques,
- suppression des numérotations et des symboles en début de lignes,
- remplacement de toutes les occurrences de "…." et ".." par un espace suivi de trois points,
- remplacement des trois points collés à un mot par un espace suivi de trois points,
- etc.

3.3.2 Clean_v2 : filtrage des phrases et mots non pertinents

La phase Clean_v2 vise à filtrer les phrases et les mots non pertinents dans le but de garantir la qualité sémantique des données. Les actions entreprises au cours de cette étape comprennent :

- Suppression des lignes qui ont moins de 3 mots.
- Suppression des lignes en doublon tout en conservant l'ordre d'apparition.

Cette étape intermédiaire s'inscrit dans notre démarche de prétraitement, visant à optimiser la qualité et la cohérence sémantique du corpus textuel.

Nous avons écrits des scripts Python pour nettoyer et traiter efficacement les données textuelles extraites du site defuwaxu. Ci-dessous, les scripts :

```
import re

def supp_doublons(input_file, output_file):
    with open(input_file, 'r', encoding='utf-8') as file:
        lines = file.readlines()

    first_line = lines[0].strip()

    # Supprimer les doublons en conservant l'ordre d'apparition
    unique_lines = list(dict.fromkeys(lines))

    with open(output_file, 'w', encoding='utf-8') as file:
        file.write(first_line + '\n')

        # Écrire les lignes uniques dans le fichier de sortie
        for line in unique_lines:
            if line.strip() != first_line:
                file.write(line)

def cleanning(chemin_fichier):
    input_file = chemin_fichier
    output_file = 'temp.txt'
    supp_doublons(input_file, output_file)

    with open(output_file, 'r', encoding='utf8') as f:
        lignes = f.readlines()

    clean = []

    for ligne in lignes:

        # Suppression des espaces en début et fin de ligne
        ligne = ligne.strip()

        # Suppression des lignes qui les mots suivants
        if 'Defu Waxu mooy' in ligne or 'Bindul soo' in ligne or '©
Newspaper' in ligne or 'Liens pour' in ligne or 'Aali Jàllo' in ligne
or 'AALI JÀLLO' in ligne:
            continue

        # Suppression des lignes qui commencent les mots suivants
        if ligne.startswith('Titres') or ligne.startswith('Save my') :
            continue

         # Suppression des liens HTTP et HTTPS
```

```
        ligne = re.sub(r'https?://\S+', '', ligne)

        ligne = ligne.replace('S. K.', "").replace('B. B. J.',
"").replace("NSS.", "").replace("N.S.S", "").replace("M. J.",
"").replace("N.S.S :", "").replace("MS & FB", "")
        ligne = ligne.replace("AG", "").replace("[", "").replace("]",
"").replace("PAJ", "").replace("CAS", "").replace("SJ",
"").replace("#", "")
        ligne = ligne.replace("LDW", "").replace("US",
"").replace("NKF", "").replace("SMA", "").replace("SMB", "")

        # Remplacement les guillemets par des guillemet informatique
        ligne = ligne.replace("»", "\"").replace("«",
"\"").replace("”", "\"").replace("`", "\"").replace("„", "\"")
        ligne = ligne.replace('‘’','"')
        ligne = ligne.replace('’’','"')

        # Suppression des chiffres + symboles en début de lignes
        ligne = re.sub(r'^\d+\)|^\(\d+\)|^\d+\.|^\d+-|^\d+ ', '',
ligne)

        # espacement
        ligne = re.sub(r'\((\w)', r'( \1', ligne)
        ligne = re.sub(r'(\w)\)', r'\1 )', ligne)
        ligne = re.sub(r'(\w)\=', r'\1 =', ligne)
        ligne = re.sub(r'(\w)\:', r'\1 :', ligne)
        ligne = re.sub(r'(\w)\!', r'\1 !', ligne)
        ligne = re.sub(r'(\w)\?', r'\1 ?', ligne)
        ligne = re.sub(r'\"(\w)', r'" \1', ligne)
        ligne = re.sub(r'(\w)\"', r'\1 "', ligne)

 # Remplacer toutes les occurrences de "…" et ".." par un espace suivi
de trois points
        ligne = ligne.replace("…", " ...")
        ligne = ligne.replace("..", " ...")

# Remplacer les trois points collés à un mot par un espace suivi de
trois points
        ligne = re.sub(r"\.\.\.(\w)", r" ... \1", ligne)

        ligne = ligne.replace("....", " ...")

        ligne = ligne.replace(". ...", "")

 # Supprimer les trois points précédés immédiatement par un guillemet
```

```
        ligne = re.sub(r'"\s*\.\.\.', '"', ligne)

# Supprimer les trois points précédés immédiatement par une parenthèse
        ligne = re.sub(r'(?<=\( )\.\.\.', '', ligne)

# Suppression de tous les caractères spéciaux (sauf les parenthèses et
les guillemets) en début de ligne
        ligne = re.sub(r'^[^\w\d\s()"]', '', ligne)

        # Supprimer les ":" en début de ligne
        ligne = re.sub(r'^\s*:', '', ligne)

        # Supprimer les ... en début de ligne
        ligne = re.sub(r'^\s*\.\.\.', '', ligne)

      # Suppression des espaces vides supplémentaires entre deux mots
        ligne = re.sub(r'\s+', ' ', ligne)

        #Suppression des ligne contenant moins de 3 mots
        if len(ligne.split()) < 3:
            continue
        # Suppression les espaces en début de ligne
        ligne = ligne.lstrip()

        clean.append(ligne)

    cleandata = 'defuwaxu_clean.txt'
    with open(cleandata, 'w', encoding='utf8') as f:
        f.write('\n'.join(clean))

    return cleandata

def nbMots(file_path):
    with open(file_path, 'r', encoding='utf8') as file:
        content = file.read()
        mots = len(content.split())
        return mots

src = 'defuwaxu.txt'
dest = cleanning(src)

nb_mots = nbMots(dest)
print(f"Nombre de mots : {nb_mots}")
```

3.3.3 Clean_final : normalisation et préparation pour l'analyse

La dernière étape du processus de nettoyage, Clean_final, se concentre sur la normalisation des encodages, prépare les données pour l'analyse ultérieure, tout en préservant les entités nommées essentielles. Cette phase inclut la mise en minuscule des textes, à l'exception des noms propres tels que les noms de personnes, de villes, de pays, etc. Par ailleurs, une vérification rigoureuse effectuée par des linguistes est nécessaire pour garantir la qualité et la cohérence des données traitées.

3.3.4 Données obtenues après prétraitement

Pour évaluer l'impact positif du processus de prétraitement sur nos données linguistiques, nous examinons les résultats obtenus après l'application des différentes phases de nettoyage. Cette évaluation se concentre sur deux langues locales du Sénégal, à savoir le wolof et le pulaar.

Wolof :

- Nombre de mots après le prétraitement : 947 113
- Nombre de phrases après le prétraitement : 46 468

Pulaar :

- Nombre de mots après le prétraitement : 652 414
- Nombre de phrases après le prétraitement : 32 213

Il est important de noter qu'aucune source de données n'a été identifiée pour la langue sérère. Malgré nos recherches approfondies sur le web et les ressources disponibles dans les bibliothèques, très peu d'ouvrages ont été trouvés. Cette situation démontre les défis persistants liés à la collecte de données pour le sérère, soulignant le besoin de consacrer des efforts supplémentaires à la documentation et à la préservation de cette langue, qui semble souffrir d'un manque significatif de données disponibles.

3.4 Création de dictionnaire de prononciation

En utilisant des ressources existantes, nous avons élaboré un modèle de prononciation visant à perfectionner la représentation phonétique du vocabulaire dans notre corpus. En fusionnant les transcriptions phonétiques issues des ouvrages de référence [54] et [55], nous avons

constitué une graine de 8 724 entrées qui servira de base pour enrichir le modèle de prononciation du wolof. L'emploi de Phonetisaurus (Novak, 2011), un système de conversion graphème-phonème (G2P), a permis la création du modèle en générant automatiquement des transcriptions phonétiques pour le vocabulaire non encore phonétisé dans nos modèles linguistiques.

Pour mettre en place un dictionnaire de prononciation avec Phonetisaurus, voici le processus détaillé.

3.4.1 Préparation

Pour commencer, téléchargez et installez Docker Desktop depuis son site officiel[25]. Une fois Docker installé, explorez la documentation de Phonetisaurus en visitant le dépôt GitHub de Phonetisaurus[26]. Ensuite, utilisez les entrées disponibles en Wolof, comprenant les mots et leurs phonétisations, pour entraîner un modèle G2P et sauvegardez ce modèle avec l'extention « .fst » (exemple model.fst). Après l'entraînement du modèle, créez un dossier local où vous stockerez à la fois le modèle et la liste de mots que vous souhaitez phonétiser qui prend une extention « .wlist » (exemple listMotWolof.wlist). Placez ces fichiers dans le dossier nouvellement créé pour une utilisation ultérieure.

3.4.2 Commandes Docker

Pour lancer le conteneur Docker avec le montage du répertoire et l'exécution d'une commande bash, utilisez la commande suivante :

```
docker run --rm -it -v ${PWD}:/mnt phonetisaurus/phonetisaurus bash -c
"lexique"
```

- `docker run --rm -it` : Lance un conteneur Docker de manière interactive et le supprime après utilisation.
- `-v ${PWD}:/mnt` : Monte le répertoire actuel (`${PWD}`) dans le conteneur sous `/mnt`.

[25] https://www.docker.com/products/docker-desktop/

[26] https://github.com/AdolfVonKleist/Phonetisaurus

- `phonetisaurus/phonetisaurus` : Utilise l'image Docker de Phonetisaurus.
- `bash -c "lexique"` : Exécute la commande bash avec l'argument "lexique". Ici, "lexique" est un placeholder pour les instructions spécifiques.

Ensuite, pour exécuter Phonetisaurus et générer des prononciations à partir d'un fichier de mots, utilisez la commande suivante :

```
phonetisaurus-apply -m /mnt/model.fst -wl /mnt/wo.wlist
```

- `phonetisaurus-apply` : Commande pour appliquer un modèle G2P afin de générer des prononciations.
- `-m /mnt/model.fst` : Spécifie le chemin du modèle G2P (`model.fst`) dans le répertoire monté.
- `-wl /mnt/wo.wlist` : Spécifie le chemin de la liste de mots (`wo.wlist`) dans le répertoire monté.

3.4.3 Résultats : lexique de prononciation

Après avoir exécuté les commandes ci-dessus, vous obtiendrez un lexique de prononciation pour les mots en Wolof. Voici un exemple de résultat typique :

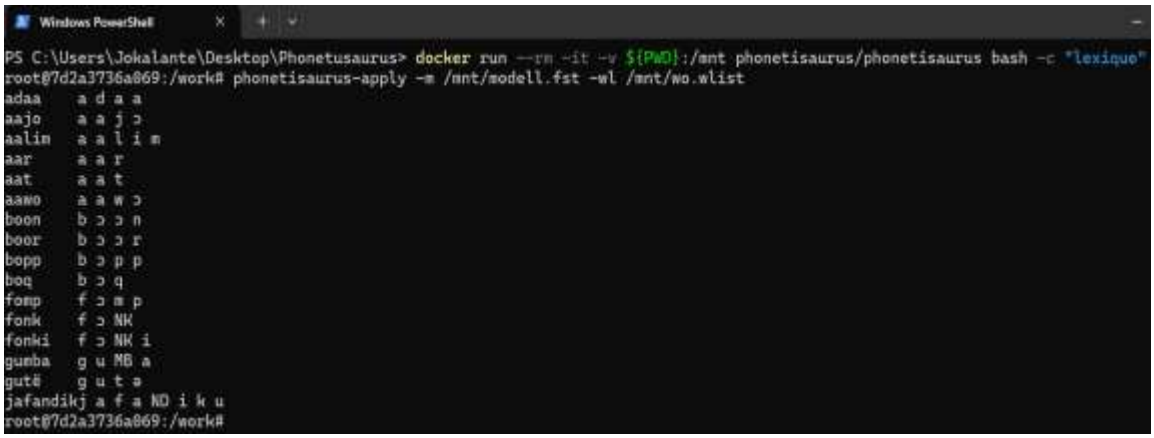

Figure 14 : résultat du lexique de prononciation

Ce processus a abouti à l'enrichissement de notre corpus wolof d'une représentation phonétique plus précise, facilitant l'analyse phonologique et renforçant la qualité des modèles de langues que nous construisons pour le wolof.

Malgré nos efforts assidus, nous n'avons pu trouver de mots accompagnés de leur prononciation pour le pulaar et le sérère. Cette lacune s'explique par la limitation des ressources disponibles en ligne et dans les bibliothèques, mettant en évidence les défis persistants liés à la collecte de données pour ces langues spécifiques. L'indisponibilité de ces données a considérablement entravé notre capacité à élaborer un dictionnaire de prononciation pour le pulaar et le sérère. Cette réalité souligne l'urgence de redoubler d'efforts dans la recherche et la documentation pour ces langues sous-représentées, afin de contribuer à la préservation et à la compréhension approfondie de leurs caractéristiques phonétiques et linguistiques.

3.5 Dépôt des données

Les données collectées au cours de notre projet sont soigneusement déposées dans des dépôts distants tels que GitHub[27], OpenSLR[28] et Zenedo[29].

GitHub est une plateforme collaborative pour le code source, offrant un suivi des versions et une large visibilité. OpenSLR est spécialisée dans les ressources vocales et linguistiques pour la reconnaissance vocale. Zenedo facilite le partage de données académiques et scientifiques, soutenant l'accès ouvert et la transparence.

En optant pour ces dépôts, notre objectif est de rendre les données accessibles en open source, favorisant ainsi la disponibilité générale et la transparence. Cette approche permet aux chercheurs et à la communauté scientifique d'accéder librement aux données, les utilisant comme ressource précieuse pour leurs propres travaux et contribuant à l'avancement global de la recherche. En choisissant des dépôts renommés et dédiés, nous nous engageons à créer un accès ouvert et facilité aux données, encourageant la collaboration et l'innovation au sein de la communauté scientifique.

[27] https://github.com/gauthelo/kallaama-speech-dataset

[28] https://www.openslr.org/151/

[29] https://zenodo.org/records/10892569

Conclusion

Ce chapitre a mis en lumière les efforts et les processus nécessaires pour concrétiser notre vision de construire ressources qui seront utilisé à des fin d'apprentissage automatique pour les langues local du pays. À travers une méthodologie bien définie, nous avons exploré la récupération des données, le prétraitement nécessaire pour assurer leur qualité, la création de dictionnaires de prononciation, et enfin, le dépôt des données pour assurer leur accessibilité à tous les intéressés.

CONCLUSION

La construction de datasets, incluant des corpus textuels, des lexiques de prononciation et des audios transcrites, constitue le cœur du projet Kallaama, affirmant ainsi notre engagement envers la préservation et l'enrichissement des langues vernaculaires du Sénégal. La réalisation de ce travail s'appuie sur une approche méthodique, débutant par l'exploration approfondie des ressources disponibles, suivie de la collecte rigoureuse des données et du prétraitement nécessaire pour garantir leur qualité. La création de lexiques de prononciation spécifiques pour les trois principales langues vernaculaires du Sénégal, le wolof, le pulaar et le sérère, est essentielle pour assurer une représentation précise et fonctionnelle de ces langues. Ces données sont ensuite déposées dans des hubs open source, favorisant ainsi leur accessibilité et leur utilisation par la communauté. Cette démarche permet non seulement de construire des ressources linguistiques fondamentales, mais aussi de soutenir le développement de technologies linguistiques innovantes et inclusives. En somme, notre initiative représente une avancée majeure dans la valorisation des langues locales, en contribuant à un avenir numérique où chaque langue est reconnue et chaque voix est entendue, tout en soutenant le progrès technologique et culturel au Sénégal.

Les perspectives de la construction de datasets, se déploient dans un panorama riche de cas d'utilisation et d'avantages anticipés, dessinant ainsi une trajectoire prometteuse pour le développement socio-économique et technologique au Sénégal. Le développement agricole et les services vocaux représentent une application clé des données générées, facilitant la communication avec les petits producteurs en leur fournissant des informations et conseils dans leur langue maternelle, ce qui est crucial pour renforcer la sécurité alimentaire et offrir des solutions adaptées aux personnes peu ou pas lettrées. Kallaama ouvre également des horizons pour des entreprises comme Jokalante, qui peuvent développer des offres de services à forte valeur ajoutée, en éliminant les barrières d'accès à l'information pour les populations illettrées. Orange, en tant qu'opérateur multiservice dominant, contribue à réduire la fracture numérique en introduisant la transcription de données audio dans les langues principales du pays, et l'École Polytechnique de Thiès (EPT), soutenue par les partenaires du projet, créera un centre d'acquisition de données langagières pour le traitement

automatique. Ce centre sera un hub stratégique pour la collecte et l'analyse des données linguistiques spécifiques, contribuant ainsi à l'évolution du paysage technologique et académique au Sénégal. Les données produites seront également une ressource précieuse pour la recherche académique sur les langues vernaculaires, offrant aux linguistes et chercheurs des données authentiques pour une meilleure compréhension des caractéristiques linguistiques, renforçant ainsi la place du Sénégal dans la recherche scientifique.

BIBLIOGRAPHIE

[1] H. Bahi, « Hybrid ASR system for teaching pronunciation », sept. 2008.

[2] E. Barnard, M. Davel, et C. van Heerden, « ASR corpus design for resource-scarce languages », présenté à Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, sept. 2009, p. 2847-2850. doi: 10.13140/RG.2.1.1824.2000.

[3] E. Barnard, J. Schalkwyk, C. van Heerden, et P. Moreno, « Voice search for development », sept. 2010, p. 282-285. doi: 10.21437/Interspeech.2010-111.

[4] A. Kumar, A. Tewari, S. Horrigan, M. Kam, F. Metze, et J. Canny, « Rethinking Speech Recognition on Mobile Devices », janv. 2011.

[5] S. Voisin, « Possession adnominale dans différentes variétés de wolof », *Afr. Online*, 2021, Consulté le: 27 décembre 2023. [En ligne]. Disponible sur: https://hal.science/hal-03351465

[6] M. Guérin, « Les constructions verbales en wolof : vers une typologie de la prédication, de l'auxiliation et des périphrases », phdthesis, Université Sorbonne Paris Cité, 2016. Consulté le: 27 décembre 2023. [En ligne]. Disponible sur: https://theses.hal.science/tel-01557412

[7] M. T. Cisse, « PROBLEMES DE PHONETIQUE ET DE PHONOLOGIE EN WOLOF ».

[8] « Diouf et al. - 2018 - Dynamique et transmission linguistique au Sénégal .pdf ». Consulté le: 15 novembre 2023. [En ligne]. Disponible sur: https://www.erudit.org/fr/revues/cqd/2017-v46-n2-cqd04128/1054052ar.pdf

[9] I. Medhi, S. Patnaik, E. Brunskill, S. N. N. Gautama, W. Thies, et K. Toyama, « Designing mobile interfaces for novice and low-literacy users », *ACM Trans. Comput.-Hum. Interact.*, vol. 18, n° 1, p. 2:1-2:28, mai 2011, doi: 10.1145/1959022.1959024.

[10] B. Lecouteux, « Reconnaissance automatique de la parole guidée par des transcriptions a priori », phdthesis, Université d'Avignon et des Pays de Vaucluse, 2008. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://hal.science/tel-01381704

[11] T. T. Ping, « Automatic Speech Recognition for Non-Native Speakers ».

[12] F. A. A. Laleye, « Contributions à l'étude et à la reconnaissance automatique de la parole en Fongbe », phdthesis, Université du Littoral Côte d'Opale ; Université d'Abomey-Calavi (Bénin), 2016. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-01628455

[13] X. Huang, A. Acero, et H.-W. Hon, « Spoken Language Processing: A Guide to Theory, Algorithm, and System Development », janv. 2001.

[14] L. R. Rabiner et B. Juang, « Fundamentals of speech recognition », présenté à Prentice Hall signal processing series, 1993. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.semanticscholar.org/paper/Fundamentals-of-speech-recognition-Rabiner-Juang/df50c6e1903b1e2d657f78c28ab041756baca86a

[15] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, et D. Yu, « Convolutional Neural Networks for Speech Recognition », *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 22, nº 10, p. 1533-1545, oct. 2014, doi: 10.1109/TASLP.2014.2339736.

[16] H. Sak, A. Senior, et F. Beaufays, « Long short-term memory recurrent neural network architectures for large scale acoustic modeling », in *Interspeech 2014*, ISCA, sept. 2014, p. 338-342. doi: 10.21437/Interspeech.2014-80.

[17] « Elodie - Méthodologie outillée de localisation d'agents con.pdf ».

[18] « Full Text PDF ». Consulté le: 20 novembre 2023. [En ligne]. Disponible sur: https://www.researchgate.net/profile/Briony-Williams/publication/2750136_The_Segmentation_and_Labelling_of_Speech_Databases/links/5597199708ae21086d220f6a/The-Segmentation-and-Labelling-of-Speech-Databases.pdf

[19] « Wells - Computer-coding the IPA a proposed extension of S.pdf ». Consulté le: 20 novembre 2023. [En ligne]. Disponible sur: https://www.phon.ucl.ac.uk/home/sampa/ipasam-x.pdf

[20] S. Hahn, P. Vozila, et M. Bisani, « Comparison of Grapheme-to-Phoneme methods on large pronunciation dictionaries and LVCSR tasks », vol. 3, p. 2537-2540, janv. 2012.

[21] L. Besacier, « Reconnaissance Automatique de la Parole pour des Langues peu Dotées: Application au Vietnamien et au Khmer », p. 6-10, janv. 2005.

[22] J. Novak, D. Yang, N. Minematsu, et K. Hirose, « Initial and Evaluations of an Open Source WFST-based Phoneticizer ».

[23] A. Stolcke, « SRILM - an extensible language modeling toolkit », in *7th International Conference on Spoken Language Processing (ICSLP 2002)*, ISCA, sept. 2002, p. 901-904. doi: 10.21437/ICSLP.2002-303.

[24] B.-J. Hsu et J. Glass, « Iterative language model estimation: efficient data structure & algorithms », in *Interspeech 2008*, ISCA, sept. 2008, p. 841-844. doi: 10.21437/Interspeech.2008-255.

[25] M. Bisani et H. Ney, « Joint-Sequence Models for Grapheme-to-Phoneme Conversion », *Speech Commun.*, vol. 50, nº 5, p. 434, mars 2008, doi: 10.1016/j.specom.2008.01.002.

[26] « Hahn et al. - Comparison of Grapheme-to-Phoneme Methods on Large.pdf ». Consulté le: 20 novembre 2023. [En ligne]. Disponible sur: http://www-i6.informatik.rwth-aachen.de/publications/downloader.php?id=811&row=pdf

[27] V. B. Le, « Reconnaissance automatique de la parole pour des langues peu dotées ».

[28] E. Gauthier, « Collecter, Transcrire, Analyser: quand la machine assiste le linguiste dans son travail de terrain ».

[29] S. Katz, « Estimation of probabilities from sparse data for the language model component of a speech recognizer », *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, nº 3, p. 400-401, mars 1987, doi: 10.1109/TASSP.1987.1165125.

[30] E. Arisoy, T. N. Sainath, B. Kingsbury, et B. Ramabhadran, « Deep Neural Network Language Models ».

[31] « Mikolov et al. - Recurrent Neural Network Based Language Model.pdf ». Consulté le: 23 novembre 2023. [En ligne]. Disponible sur: https://people.cs.pitt.edu/~huynv/research/deep-nets/Recurrent%20neural%20network%20based%20language%20model.pdf

[32] « Sundermeyer et al. - 2012 - LSTM neural networks for language modeling.pdf ». Consulté le: 23 novembre 2023. [En ligne]. Disponible sur: http://www-i6.informatik.rwth-aachen.de/publications/downloader.php?id=820&row=pdf

[33] N.-Q. Pham, G. Kruszewski, et G. Boleda, « Convolutional Neural Network Language Models », in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, 2016, p. 1153-1162. doi: 10.18653/v1/D16-1123.

[34] A. Vaswani *et al.*, « Attention is All you Need », in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[35] J. Devlin, M.-W. Chang, K. Lee, et K. Toutanova, « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », 24 mai 2019, *arXiv*: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[36] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, et Y. Bengio, « End-to-End Attention-based Large Vocabulary Speech Recognition », 14 mars 2016, *arXiv*: arXiv:1508.04395. Consulté le: 25 décembre 2023. [En ligne]. Disponible sur: http://arxiv.org/abs/1508.04395

[37] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, et Y. Bengio, « Attention-Based Models for Speech Recognition », 24 juin 2015, *arXiv*: arXiv:1506.07503. doi: 10.48550/arXiv.1506.07503.

[38] J. Li, R. Zhao, H. Hu, et Y. Gong, « Improving RNN Transducer Modeling for End-to-End Speech Recognition », 26 septembre 2019, *arXiv*: arXiv:1909.12415. doi: 10.48550/arXiv.1909.12415.

[39] L. Dong, S. Xu, et B. Xu, « Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition », in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, avr. 2018, p. 5884-5888. doi: 10.1109/ICASSP.2018.8462506.

[40] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, et S. Watanabe, « End-to-End Speech Recognition: A Survey », 2 mars 2023, *arXiv*: arXiv:2303.03329. doi: 10.48550/arXiv.2303.03329.

[41] D. Rybach *et al.*, « RASR - The RWTH Aachen University Open Source Speech Recognition Toolkit », déc. 2011.

[42] D. Povey *et al.*, « The Kaldi Speech Recognition Toolkit ».

[43] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, et I. Sutskever, « Robust Speech Recognition via Large-Scale Weak Supervision », 6 décembre 2022, *arXiv*: arXiv:2212.04356. Consulté le: 6 août 2024. [En ligne]. Disponible sur: http://arxiv.org/abs/2212.04356

[44] V. Berment, « Méthodes pour informatiser les langues et les groupes de langues `` peu dotées ' ' ».

[45] K. P. Scannell, « The Crúbadán Project: Corpus building for under-resourced languages ».

[46] L. Besacier, E. Barnard, A. Karpov, et T. Schultz, « Automatic speech recognition for under-resourced languages: A survey », *Speech Commun.*, vol. 56, p. 85-100, janv. 2014, doi: 10.1016/j.specom.2013.07.008.

[47] M. Walsh, « Will indigenous languages survive? », *Annu. Rev. Anthropol.*, vol. 34, p. 293-315, sept. 2005, doi: 10.1146/annurev.anthro.34.081804.120629.

[48] A. Nimaan, P. Nocera, et J.-M. Torres-Moreno, « Boîte à outils TAL pour des langues peu informatisées : le cas du somali », 2006.

[49] T. Pellegrini, « Transcription automatique de langues peu dotées ».

[50] E. Barnard, M. Davel, et G. Van Huyssteen, « Speech Technology for Information Access: a South African Case Study. », présenté à AAAI Spring Symposium - Technical Report, mars 2010. doi: 10.13140/RG.2.1.1988.0407.

[51] E. Gauthier *et al.*, « Preuve de concept d'un bot vocal dialoguant en wolof », in *Traitement Automatique des Langues Naturelles (TALN 2022)*, Y. Estève, T. Jiménez, T. Parcollet, et M. Zanon Boito, Éd., Avignon, France: ATALA, juin 2022, p. 403-412.

Consulté le: 22 juillet 2023. [En ligne]. Disponible sur: https://hal.science/hal-03701495

[52] E. Gauthier, L. Besacier, S. Voisin, M. Melese, et U. P. Elingui, « Collecting Resources in Sub-Saharan African Languages for Automatic Speech Recognition: a Case Study of Wolof », in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, et S. Piperidis, Éd., Portorož, Slovenia: European Language Resources Association (ELRA), mai 2016, p. 3863-3867. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://aclanthology.org/L16-1611

[53] S. Voisin, « Relations entre fonctions syntaxiques et fonctions sémantiques en wolof », *Dr. Sci. Lang. Univ. Lumière …*, déc. 2002, Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.academia.edu/1916793/Relations_entre_fonctions_syntaxiques_et_fonctions_s%C3%A9mantiques_en_wolof

[54] J. L. Diouf, *Dictionnaire wolof-français et français-wolof*. KARTHALA Editions, 2003.

[55] A. Fal, R. Santos, et J. L. Doneux, *Dictionnaire wolof-français: suivi d'un index français-wolof*. Karthala, 1990.